Design, simulation and analysis of collaborative strategies in multiagent systems: The case of supply chain management

Thierry Moyaux

To cite this version:

Thierry Moyaux. Design, simulation and analysis of collaborative strategies in multiagent systems: The case of supply chain management. Networking and Internet Architecture [cs.NI]. Université Laval, 2004. English. NNT : . tel-00091295

HAL Id: tel-00091295
https://theses.hal.science/tel-00091295
Submitted on 5 Sep 2006

THIERRY MOYAUX

DESIGN, SIMULATION AND ANALYSIS OF COLLABORATIVE STRATEGIES IN MULTI-AGENT SYSTEMS: THE CASE OF SUPPLY CHAIN MANAGEMENT

Thèse présentée
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de doctorat en informatique
pour l'obtention du grade de Philosophiæ Doctor (Ph.D.).

FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

NOVEMBRE 2004

Résumés

Résumé court

Une chaîne logistique est composée d'entreprises fabriquant et distribuant des produits aux consommateurs. En modélisant chacune de ses entreprises comme un agent intelligent, nous étudions l'effet « coup de fouet » qui s'y propage. Cet effet consiste en l'amplification de la variabilité des commandes passées par les entreprises lorsque l'on s'éloigne du client final.

Dans un premier temps, nous modélisons chaque entreprise d'une chaîne logistique forestière québécoise comme un agent intelligent, afin de proposer deux mécanismes de coordination décentralisés réduisant ce phénomène. Des simulations de ce système multiagent montrent que ce mécanisme est efficace pour une chaîne logistique dans son ensemble.

Dans un second temps, d'autres simulations sont utilisées pour construire un jeu, que nous analysons avec la Théorie de Jeux. Nous vérifions ainsi que les entreprises n'ont pas intérêt d'arrêter unilatéralement d'utiliser nos mécanismes de coordination (équilibre de Nash).

Résumé long

Une chaîne logistique est composée d'entreprises fabriquant et distribuant des produits aux consommateurs. En modélisant chacune de ses entreprises comme un agent intelligent, nous étudions l'effet « coup de fouet » (*Bullwhip effect*) qui s'y propage. Cet effet est une amplification de la variabilité de la demande lorsque l'on s'éloigne du client final. On peut aussi voir ce phénomène comme un cas particulier de fluctuations des flux dans un système distribué. Ces fluctuations réduisent l'efficacité de la chaîne logistique, principalement du fait de l'élévation des niveaux d'inventaire et de la réduction de l'agilité... On estime que ce phénomène coûterait de 40 à 60 millions USD pour une papetière de 300 kilotonnes.

L'effet coup de fouet étant provoqué par un manque de coordination entre les agents, nous proposons deux principes qui doivent inspirer tout mécanisme de coordination, à savoir : (i) commander ce que l'on nous commande élimine l'effet coup de fouet mais ne gèrent pas les inventaires, et (ii) les entreprises ne devraient réagir qu'une seule fois à chaque changement dans la consommation du marché. Afin de valider ces deux principes, nous simulons une chaîne logistique forestière appelée le Jeu du Bois Québécois. Ce jeu permet d'enseigner ce qu'est l'effet coup de fouet. Chaque joueur-entreprise y est modélisé par un agent intelligent appliquant une stratégie donnée pour passer ses commandes. À cet effet, nous avons conçu deux stratégies suivant nos deux principes.

Dans un premier temps, nous comparons expérimentalement l'efficacité de ces deux stratégies avec cinq autres stratégies. Nous supposons ici que la chaîne logistique est homogène, c'est-à-dire que toutes ses entreprises utilisent la même stratégie de commande. Nous vérifions ainsi que nos deux mécanismes de coordination, implémentés sous la forme de stratégies, sont efficaces pour la chaîne logistique dans son ensemble.

Dans un second temps, nous cessons de supposer la chaîne homogène pour faire davantage de simulations nous permettant de construire un jeu. En analysant ce jeu avec la Théorie de Jeux, nous vérifions que les entreprises n'ont pas intérêt d'arrêter unilatéralement d'utiliser nos deux mécanismes de coordination (équilibre de Nash).

Abstracts

Short abstract

A supply chain is a set of companies that manufacture and distribute products to consumers. We study the “bullwhip effect” that is propagated therein by modelling each company as an intelligent agent. This effect is the amplification of the variability of orders placed by companies, as one moves away from end-customers.

Firstly, we model each company in a Québec wood supply chain as an intelligent agent, in order to propose two decentralized coordination mechanisms reducing this phenomenon. Simulations of this multi-agent system show that our mechanism is efficient for a supply chain as a whole.

Secondly, additional simulations are used to build a game, which we analyze with Game Theory. We verify here that companies have no incentive to cease unilaterally from using our two coordination mechanisms (Nash equilibrium).

Long abstract

A supply chain is a set of companies that manufacture and distribute products to end-customers. We model each company as an intelligent agent to study the "bullwhip effect", that is, the amplifcation of the variability in orders placed by companies. This phenomenon can also be seen as a particular case of stream fluctuations in a distributed system. Such fluctuations incur costs due to higher inventory levels and supply chain agility reduction. This cost has been estimated at 40-60 million USD for a 300 kton paper mill.

Since the bullwhip effect is due to a lack of coordination, we propose two principles that any coordination mechanism should respect: (i) ordering what is ordered from us eliminates the bullwhip effect but do not manage inventory, and (ii) companies should react only once to each market consumption change. In order to validate these two principles, we simulate a forest supply chain called the Québec Wood Supply Game. This game is used to teach the bullwhip effect. Each company-player is modelled as an intelligent agent applying a given ordering rule. For this simulation, we have designed two rules that respect our two principles.

Firstly, we compare experimentally the efficiency of these two rules with five other rules. We assume here that the supply chain is homogeneous, that is, every company uses the same ordering rule. In this way we check that our two coordination mechanisms, i.e., our two ordering rules, are efficient for the supply chain as a whole.

Secondly, we stop assuming the supply chain is homogeneous to carry out additional simulations in order to build a game. The analysis of this game according to Game Theory shows that companies do not have incentive to cease unilaterally from using our two coordination mechanisms (Nash equilibrium).

Avant-propos

Je tiens tout d'abord à exprimer ma plus grande reconnaissance à mon directeur Brahim Chaib-draa et à ma co-directrice Sophie D'Amours pour m'avoir soutenu moralement et financièrement dans le cheminement de quatre ans[1] que représente la présente thèse. Étant donné qu'ils œuvrent dans deux domaines différents, je les remercie d'avoir guidé ma recherche dans leur domaine respectif, tout en faisant l'effort de comprendre les problèmes apparaissant dans le domaine de leur collègue. Je tiens également à exprimer ma profonde gratitude à Bernard Espinasse (Professeur à l'Université Paul Cézanne de Marseille, France) qui a co-dirigé « à distance » cette thèse et qui fut auparavent mon directeur de DEA. C'est en particulier lui qui m'a motivé à poursuivre des études de doctorat.

Que Jean-Paul Barthès (Professeur à l'Université de Technologie de Compiègne, France) et Peter Kropf (Professeur à l'Université de Neuchâtel, Suisse) reçoivent ma reconnaissance pour m'avoir fait l'honneur d'évaluer ma thèse et d'assister à sa soutenance. J'exprime aussi ma gratitude à Jean-Marc Frayret (Professeur à l'Université Laval et directeur adjoint de la recherche à FOR@C) pour avoir non seulement évalué ma thèse et assisté à sa défense, mais aussi adapté le jeu du bois finlandais au contexte québécois afin que je m'en serve dans mes travaux.

Ensuite, il m'est très important de mentionner la collaboration de plusieurs personnes sans qui ce travail n'aurait pas été possible. Je commencerai par les membres du laboratoire DAMAS, à savoir Amar, Arsène, Charles, Dominic, Étienne, Emmanuel, Fabiola, François, Frédérick, Houssein, Imed, Jean-François, Julien, Ludovic, Marc-André, Martin, Maxime, Mathieu, Mourad, Nicolas (deux fois), Olivier, Patrick, Philippe (deux fois), Pierrick, Roberto, Sébastien, Simon, Stéphane et Vincent. Il ne manque à cette liste qu'Ève que je remercie plus particulièrement pour la belle interface graphique qu'elle m'a programmée (voir la figure 6.10). Je remercie aussi les membres de FOR@C, et particulièrement Martin Cloutier qui m'a gentiment donné les paramètres nécessaires à l'application de mon travail au contexte de l'industrie forestière québé-

---

[1] ... dont presque deux sous la « maudite m... blanche »...

coise. Je tiens également à exprimer ma reconnaissance à Connie et Phyllis qui ont corrigé tout ce que j'ai rédigé en anglais.

Enfin, j'adresse toute ma sympathie à mon entourage personnel, et en premier lieu à mes colocataires : notre « propriétaire-colocataire » Jean-François bien sûr, mais aussi Cecilia, Christian, Éric, France, Jocelyn et Richard. Ma gratitude va aussi aux membres de ma famille pour leur patience durant ces quatre années passées chez « nos cousins » québécois. En particulier, mes derniers remerciements, mais non les moindres, s'adressent à mes parents qui m'ont donné le goût de la découverte.

à mes parents et grands-parents,

Là où y'a d'la chaîne,
y'a du plaisir !

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Stream fluctuations may occur in many types of distributed systems, or in other words, the streams in these systems are not stable. Since such an instability decreases the efficiency of the considered distributed system, we would like to reduce it. In particular, Multi-Agent Systems are distributed systems that are made of several autonomous entities, called agents, that interact by requesting to other to carry out some tasks (request stream) and by carrying out these tasks (task stream). Since such multi-agent systems are distributed, their efficiency may also be reduced by fluctuations in their internal streams. In this dissertation, we focus on supply chains dynamics using multi-agent systems to come up with fundamental learning about the value of collaboration within a supply chain. A supply chain is a network of companies producing and distributing products to end-customers. In this context, fluctuations affect order stream, and are called the "bullwhip effect". This phenomenon decreases supply chain efficiency, because it induces an inventory level increase and a supply chain agility decrease. The problem of stream fluctuations, and in particular the bullwhip effect, are presented in Section 1.1.

This dissertation pursues two goals:

- the understanding of the dynamics of a whole supply chain in order to propose some coordination mechanisms reducing the bullwhip effect while taking into account operational constraints and maintaining service level;
- the verifying of agents' incentive to use such coordination mechanisms.

To achieve these goals, we model each company as an intelligent agent, i.e., as an autonomous, software entity, which allows us to consider a whole supply chain, instead of one or two companies, as is usually done in other studies of the bullwhip effect. With this agent model, our first goal is addressed by designing each company-agent's behaviour in order to reduce the bullwhip effect in the supply chain. More precisely, agent behaves according to the distributed coordination mechanism that we propose as a solution to the bullwhip effect. However, the emphasis of this work is more on methodology than on optimization of the coordination mechanism, because simulations show that coodination decreases logistics costs by several orders of magnitude.

The problem is that agents may disagree to use our coordination mechanism. For this reason, our second goal is to check whether every agent has an individual incentive to behave according to our coordination mechanism. Precisely, game theory is used to analyze inter-agent interactions in our simulation of a supply chain. These research questions are motivated in Section 1.2. Those questions are interesting for both supply chain management and multi-agent systems. We detail our contributions for these two fields in Section 1.4. Next, we compare our work with existing research contribution in Section 1.5. Finally, Section 1.6 outlines the content of this thesis.

1.1 Stream Fluctuations in Distributed Systems: The Bullwhip Effect

The performance of many distributed systems is reduced by the unstability of their internal streams. As a distributed system, a multi-agent system may face this problem too. In this dissertation, we focus on supply chains modelled using multi-agent systems. This dissertation focusses on this particular type of stream fluctuation.

Supply chains are distributed systems composed of different companies producing and distributing products to customers. In such a supply chain, fluctuations affect order streams, as illustrated in Figure 1.1:

- the Retailer sells to the customer and buys from the Wholesaler;
- the Wholesaler sells to the Retailer and buys from the PaperMill;
- the PaperMill sells to the Wholesaler and buys from an unknown supplier.

As orders flow within the supply chain, their variability increases. This phenomenon of fluctuation of the order stream is known as the "bullwhip effect". Lee et al. [1997a]

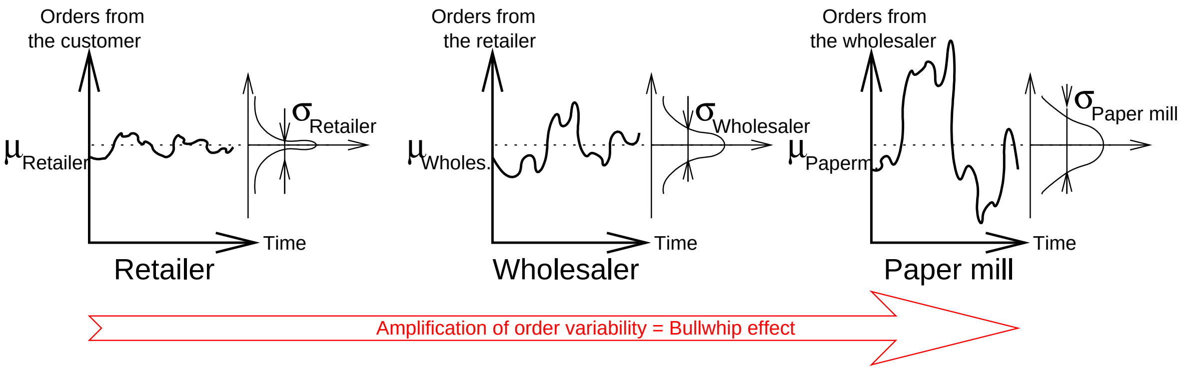


Figure 1.1: The bullwhip effect [Lee et al., 1997a,b].

said about this phenomonon: "The ordering patterns of the three companies share a common, recurring theme: the variabilities of an upstream site are always greater than those of the downstream site". As a variability, the bullwhip effect is measured by the standard deviation σ of orders (notice that the means μ of orders are all equal in our example in Figure 1.1).

Forrester [1958] was the first author to describe this phenomenon. Next, Lee et al. [1997a,b] have proposed four causes of this effect: (i) demand forecast updating, (ii) order batching, (iii) price fluctuation and (iv) rationing and shortage gaming. Other authors have extended Lee and his colleagues' causes and solutions to the bullwhip effect: (v) misperception of feedback in the supply chain [Sterman, 1989], (vi) local optimization without global vision [Kahn, 1987; Naish, 1994; Shen, 2001], and (vii) variabilities due to company processes [Taylor, 1999].

There are several consequences of the bullwhip effect. In a few words, this effect incurs costs due to higher inventory levels, supply chain agility reduction, decrease of customer service levels, ineffective transportation, missed production schedules... In fact, such fluctuations of the demand lead every participant in the supply chain to stockpile because of a high degree of demand uncertainties and variabilities [Lee et al., 1997b]. An insight into the importance of this problem is given by Carlsson and Fullér [2001] who estimate that the costs incurred by the bullwhip effect are 200-300 MFIM (40-60 millions USD) annually for a 300 kton North-European paper mill.

According to Cooper et al. [1997], the bullwhip effect is a coordination problem between *autonomous* companies, which can thus be considered as agents. Therefore, the proposed solution to the bullwhip effect is a decentralized coordination technique. On account of this, the first point addressed by this dissertation is the design of *coordi-*

nation techniques reducing information flow fluctuations[1] in distributed systems. This point is studied for the special case of a supply chain modelled as a multi-agent system, in which the demand amplification has to be reduced, while keeping low inventories and good customer service levels. It is worth noting here the difference between the words "distributed" and "decentralize". In fact, the systems considered in this dissertation are distributed in a geographic and/or in a functional way. It follows that the proposed coordination mechanism has to take into account this distribution. The simplest solution would be to centralized the coordination, i.e., to transmit the whole requested information to a node in the system, then to compute the optimal optimization, and finally to make this coordination run in the system. This approach is not the actual philosophy of multi-agent systems. In fact, our coordination mechanism has no center, or more precisely, every node in the system takes part to the coordination process.

Another issue may appear with such coordination techniques: some agents may disagree to use it, because they are better off with fluctuations. More generally, agents may refuse to coordinate in a multi-agent system, because they do not profit, or even suffer, from the increase of the system efficiency incurred by the coordination mechanism. Similarly, coordination may be expensive, and each agent would prefer that the rest of the multi-agent system coordinates, but without itself, because this agent would profit from the effort of coordination made by the others without participating in this effort. In most supply chain, companies have different conflicting objectives. In particular, on the one hand, the bullwhip effect disturbs the PaperMill in Figure 1.1 while it cannot reduce it, and on the other hand, the Retailer in Figure 1.1 is the first company to generate this phenomenon while it does not have incentives to reduce it, because it does not directly suffer from it. This problem is thus interesting, because if the Retailer spends money to reduce the bullwhip effect, it does not directly profit from this reduction while it pays for it. In fact, the Wholesaler and the PaperMill directly profit from this reduction, and the Retailer has only an indirect benefit, due to the fact that it receives product with lower delays, because the Wholesaler and the PaperMill have less stockouts and are thus able to ship product to the Retailer in time. In other words, the problem is not only to understand the dynamics generating fluctuations in order to propose some solutions, but also to check if companies will individually profit from this solution, and if this is not the case, to propose some incentive alignment mechanism. This problem of individual incentive that may be in contradiction with the group welfare is the second point addressed in this dissertation, in the special case of the adoption of a mechanism reducing the bullwhip effect in the supply chain.

To sum up, the two *goals* of this thesis are (i) to study how stream fluctuation propagates in a distributed system by considering the case of a supply chain modelled

---

[1] Both product and order streams in supply chains may fluctuate and be disturbed.

as a multi-agent system reducing these fluctuations, and to propose a coordination mechanism as solution, and (ii) to check if company-agents have individual incentives to use such a coordination mechanism. We now motivate our approach to address these two points.

1.2 Motivations

We have previously presented the bullwhip effect as a particular case of stream fluctuations in a distributed system. In this thesis, we use simulations to address the two questions related to stream fluctuations, as stated in the previous section. First, we study how such fluctuations are generated and how to reduce them. To achieve that, we see each company in a supply chain as an intelligent agent, and we look for agents' individual decisions reducing order fluctuations, i.e., reducing the bullwhip effect. In this dissertation, an intelligent agent is a software conceived as an autonomous entity interacting with other agents. Inter-agent interactions are formalised with game theory in the second question addressed. This question concerns company-agents' incentive to reduce stream fluctuations by adopting the behaviour proposed in our coordination mechanism. We now develop our motivations to address these two questions.

1.2.1 Motivation to Propose a Coordination Mechanism

First, we take the agent model of a company in the QWSG (Québec Wood Supply Game), which is a board-game based on the Beer Game to teach supply chain dynamics in the Québec forest industry. That is, we replace human players in the QWSG by intelligent agents. The supply chain model in the QWSG is interesting, because it is simple while the emerging dynamics are very complex. In this context, our goal is to understand how these agents have to place orders in order to minimize the overall supply chain cost, according to the QWSG rules. From the viewpoint of supply chain management, we look for an ordering scheme that *stabilizes placed orders*, while minimizing inventory levels, avoiding stockouts, and fulfilling market demand. From the viewpoint of multi-agent systems, the market demand can be seen as the output of the multi-agent system, and our goal is to design agent's behaviour so that the system reponses the most efficiently to the demand. To do that, we design a decentralized coordination mechanism making each agent behave according to the set point requested as system output, where the set point corresponds to the market demand when the considered system is a supply chain.

To obtain this stabilization, the basic idea is to make all agents' orders conform to the market consumption. This solution to the bullwhip effect is intuitive, but as we will see later, it does not take operational constraints into account, i.e., inventories in the QWSG are not well managed. Such a solution can be summarized in two principles:

Principle 1: *Lot-for-lot orders eliminate the bullwhip effect, but do not manage inventories.*

Lot-for-lot ordering means that each company orders what has been demanded from it, e.g., if its client wants 10 products, the company places an order for 10 products. With such a strategy, the bullwhip effect is eliminated, but inventory levels are not managed. Therefore, we keep lot-for-lot orders, but we add another piece of information to manage inventory levels. Now, orders are vectors (O, Θ) instead of a unique number X. O in (O, Θ) are set according to the lot-for-lot scheme. This second piece of information is also an order, but its objective is only to stabilize inventory at its *initial level*. Therefore, the Retailer transmits the market consumption to the Wholesaler in O, next the Wholesaler transmits this information to the PaperMill, etc. As a result, the first principle rules O, but we need another principle to rule Θ. The way to emit this information is also studied in order to avoid that this second type of order be emitted at any time, i.e., to avoid the bullwhip effect being moved in this second type of order. The needs for this point are summarized in our second principle:

Principle 2: *Companies should react only once (by under- or overordering) to each market consumption change.*

In this context, Θ are equal to zero all the time, except when market consumption changes, in which case companies react to this change by sending non-zero Θ in order to stabilize their inventory to the initial level. In other words, Θ are proportional to the variation of the market consumption. As we can see, these two principles imply collaboration in the supply chain, where the *collaboration is based on information sharing*. In this thesis, we consider that collaboration is demand information sharing, that is, companies share their demand with their suppliers. We also study information centralization, where retailers multi-cast the market consumption in real-time to the whole supply chain, accelerating real demand transmission.

As an illustrative example of our two principles, we propose two ordering schemes reducing the bullwhip effect. As stated above, agents place (O, Θ) vectors as orders, where O are ruled according to our first principle and Θ according to our second principle. As a result, there is *information sharing*, because each company transmits in O

the market consumption, the difference between the two proposed ordering schemes lies in the manner of performing this information sharing. In our first scheme, O represent the market consumption transmitted from each company to its suppliers, and Θ are used to order more or less than O when required. In this case, O are used to perform the information sharing. Our second scheme works in a similar way, except that we assume there is information centralization, i.e., retailers multi-cast in *real-time* the market consumption in the whole supply chain. As a consequence, company-agents base their placed orders directly on the actual market consumption, instead of on O. In this case, O also represent the market consumption transmitted by the client, but a 'fresher' information is provided by information centralization. In fact, the market consumption transmitted in O in the first scheme is as slow as orders, while information centralization in the second scheme is assumed to be instantaneous and in real-time.

The efficiency for reducing the bullwhip effect of these two schemes is validated by simulations of the QWSG. Precisely, our two schemes are compared with five other ordering rules that are not ruled by our two principles. As we focus on the bullwhip effect, our metric is the standard deviation of orders placed by companies. We also use three other metrics to measure the consequences of the bullwhip effect. We note in our simulations that our two ordering schemes *change the supply chain dynamics* in two ways. The first way is the reduction of information stream fluctuations, i.e., our goal of reducing the bullwhip effect is reached. The second way is more surprising: there is a distribution change in backorder durations among companies. In fact, retailers' backorders become longer within a collaborative scheme, while the retailers are the least disturbed companies by backorders within a traditional approach. The consequence of this change in backorder duration distribution is a change in the cost distribution in the supply chain.

These solutions to the bullwhip effect are also interesting for multi-agent systems, because some causes of this effect may also exist for other types of distributed systems. When this is the case, our solution only needs to be translated for these other types of distributed systems; we will see the example of vehicles on roads in Subsection 5.6.2. Moreover, conclusions drawn from experiments on our special case of distributed system, i.e., the supply chain in the QWSG, may also apply to these other distributed systems, such as multi-agent systems. For instance, the effect noted on the change in backorder duration distribution may also occur in other distributed systems, but in a different way affecting system dynamics.

1.2.2 Motivations for Studying Incentives to Use our Coordination Mechanism

As previously stated, there is a change in backorder duration distribution, that arises the second question addressed in this dissertation. In fact, companies may refuse to use our schemes because of this distribution change, which may be an issue, because our ordering schemes are based on the market consumption measured by retailers and transmitted to other companies. In this context, if a company disagrees to communicate the market consumption information to the rest of the supply chain, our two ordering schemes using (O, Θ) orders cannot work properly, and companies become unable to apply our second principle. Moreover, each company may prefer that all other companies collaborate but not itself. This is the reason why we study individual incentives to use either the (s, S) policy (a classic ordering policy in the field of Inventory Management), or one of the two proposed schemes.

Each company decides which ordering schemes to use. This choice is made to minimize its costs, but this choice also impacts on the rest of the supply chain. Many simulations are carried out to check this effect, that is, what happens for each company when a particular company chooses a particular scheme. Game theory is used to analyze this huge quantity of simulation outcomes. During this analysis, overall supply chain cost is again taken into account, as well as each company's cost, to find when each company has the lowest cost depending on others' choice.

The result of our analysis is that all companies prefer to use our ordering schemes, because if Company i does not choose one of our two schemes while the rest of the supply chain does so, then Company i has a higher cost. In other words, using one of our two schemes is a Nash equilibrium, and using the (s, S) policy is not a Nash equilibrium. Thus, we got significant results with game theory. Notice that game theory is more and more frequent in both supply chain management [Cachon and Netessine, 2003] and multi-agent system fields [Boutilier et al., 1994]. Furthermore, Geoffrion and Krishnan [2003] cite program objectives for National Science Foundation [2004]'s Advanced Computational Research Program showing that simulations are very useful in research, i.e., our multi-agent simulations are useful for supply chain management, even if they are not theoretical:

> As pointed out in many documents and reports, computer simulation has now joined theory and experimentation as a third path to scientific knowledge. Simulation plays an increasingly critical role in all areas of science and engineering.

Geoffrion and Krishnan [2003] also support this idea:

> Our second main observation concerning management science work [...] is that there is a trend toward making greater use of computation and communication.

Next, they explain this observation:

> Both empirical and methodological work are gaining at the expense of purely theoretical work. It is easy to understand such a trend in terms of the changing relative productivities of these three kinds of research work, because computation, data, and communication are much more important as factors of production in empirical and methodological work than in theoretical work. The cost and availability of these factors have changed enormously while technology has made thinking, the main ingredient of theoretical work, only moderately more productive per labour hour or per resource dollar.

We now present how we use agent-based simulations in supply chain management.

1.3 Methodology

In this research, we propose two decentralized coordination mechanisms to reduce order fluctuations in a supply chain, next we check if company-agents will agree to use this mechanism. In the simulations used in these two parts[2], each company is programmed as an agent that is travelled by order and product streams, and we combine these agents to form our simulation of a supply chain. Each agent's behaviour is implemented by its ordering scheme. In the first part, we simulate the seven following ordering schemes:

- an ordering scheme called (s, S) policy, which is a classic ordering policy in the field of inventory management;
- two ordering schemes which are the decentralized coordination mechanisms using (O, Θ) orders that we propose;

---

[2] Two simulators were developpped, but only the first one was used in our experiments.

- four ordering schemes which are used to understand which base principles are key factors to reduce the bullwhip effect.

Since the four latter schemes were simulated only to understand that we need our two principles to reduce the bullwhip effect, we drop them in the second part of our work, in order to focus on the confrontation of the well-known (s, S) with our two schemes. The methodology in these two parts is now detailed.

1.3.1 Methodology for Validating our Coordination Mechanism

In this first part, we check if the two schemes based on our two principles minimize the bullwhip effect, while inventories are well managed, that is, inventory levels are as low as possible, but sufficient to fulfill incoming demand (in practice, the two schemes are designed so that inventory levels stabilize at the initial inventory level when the demand is constant for a sufficient time, but we only do so in our simulation to check the efficiency of our concepts). Here, all companies behave the same, which is referred to a “homogeneous supply chain”.

Since our goal is to reduce the bullwhip effect, the considered metric is the standard deviation of orders placed by companies, and in particular, orders placed by the most upstream supplier. We also consider three indirect metrics measuring the harmful consequences of the bullwhip effect. The first of these three indirect metrics is the logistics cost incurred by the entire supply chain. We consider in this first part the supply chain as a whole, without considering each company’s interest for the moment. The two other indirect metrics are (i) the standard deviation of the inventory level, to measure the average inventory level (the greater the standard deviation of inventory level is, the more inventory level fluctuates, and thus, the higher the average inventory level is, because companies increase their safety stock to avoid stockouts), and (ii) the number of backordered items, to measure the customer service levels.

1.3.2 Methodology for Checking Incentives for Collaboration

In the second part, we assume that the supply chain is heterogeneous by letting companies use any of the three schemes: the classic (s, S) or one of our two schemes. The question is now to find which scheme will be chosen by each company. In fact, our two schemes may incur negative consequences for some agents, although these schemes are good for the overall supply chain. The methodology consists in using the same simulation as in the homogeneous case, but each company can use different ordering schemes from other companies. Precisely, $3^6 = 729$ simulations are carried out, in which all companies use the first ordering scheme in the first simulation, next, all companies use the first ordering scheme except one company which uses the second scheme in the second simulation, ... and all companies use the third scheme in the 729^{th} simulation. As a comparison, we have carried out only one simulation per each of the 7 ordering schemes in the first part of our experiment program.

For each simulation and for each company, we record the backorder and inventory holding costs. We use these 6×3^6 costs to build a game in the normal form in which we look for dominated strategies, Nash equilibria and Pareto dominations of these equilibria. As this game is very large, a software called "Gambit" was used for this analysis. Results from Gambit show that using one of our ordering schemes is a Nash equilibrium, that is, no company would like to stop using this scheme if the rest of the supply chain also continues using it, while using (s, S) is never a Nash equilibrium.

We can note that we have not only used multi-agent systems to model the supply chain, but we also contribute to this field by designing a decentralized coordination mechanism aiming at reducing the bullwhip effect, and by proposing a game theory-based methodology to study incentives for using a coordination mechanism. We should note that this methodology appears as the main contribution of this work because it allows evaluating if collaborative approaches are collectively and individually efficient. We now develop how our work contributes to the two fields of supply chain management and multi-agent systems.

1.4 Contributions

As previously stated, our main contribution is the methodology that applies game theory to analyze multi-agent simulations. This methodology is interesting for the two fields adressed in this disseratation; our contributions for supply chain management are presented first, followed by our contributions for multi-agent systems.

1.4.1 Contributions to Supply Chain Management

Contributions to supply chain management are obvious, because the studied problem belongs to this field. In fact, the three main contributions of this thesis for supply chain management are:

1. the explanation of *why delays are themselves a cause of the bullwhip effect*, while such delays are only seen in the literature as an aggravating factor of another cause of the bullwhip effect;
2. the proposition of *two principles for minimizing this cause* of the bullwhip effect, and two schemes illustrating these two principles;
3. the study of *companies' incentive* for collaboration, i.e., the use of our methodology that applies *game theory* to verify the incentive for applying our two principles;
4. a broad, critical review of literature about the bullwhip effect, that presents this effect as a special case of *stream fluctuations in distributed systems*;
5. the study of the dynamics in an *entire supply chain*. In particular, we verify that collaboration is not only the best strategy for the whole supply chain, but also, that no companies have incentives to stop collaborating.

The two first contributions are closely related. On the one hand, our explanation of why delays are themselves a cause of the bullwhip effect permits us to propose our two principles, and on the other hand, these two principles aim at reducing the bullwhip effect generated by delays. Two ordering schemes are then proposed based on these two principles. In the first scheme, companies only have O to know the market consumption. Therefore, Θ are proportional to the variation of O to respect the second principle. This first version is called “Ordering scheme B” in this document. In the second ordering scheme, information centralization is used, that is, retailers multi-cast the market consumption to the whole supply chain. Like (O, Θ) orders, information centralization also transmits the market consumption information, except that information centralization is much quicker because it is instantaneous and in real-time. To profit from this acceleration of information sharing, our second ordering scheme is made more efficient by setting Θ proportional to the variation of the market consumption: as soon as the market consumption changes, non-zero Θ are sent by all companies. Moreover, companies set O equal to the market consumption transmitted by retailers instead of on incoming O, again in order to react quicker to the market consumption change.

These two ordering schemes are then compared with simulations under different scenarios. Precisely, we compare our two ordering schemes with five other schemes under nineteen different market demand patterns to study the efficiency of our schemes at the supply chain and individual levels, that is, we check if our schemes are efficient for the entire supply chain in the QWSG, and if every company agrees to use them.

In order to address the third main contribution listed above, i.e., companies' incentive for collaboration, we carry out additional simulations. In these simulations, we use a cost function adapted from the Québec forest industry, and we apply ordering parameters that are optimized so that the overall supply chain cost is minimum. In fact, our two ordering schemes assume companies collaborate, because they share demand information, but we have to check if they agree to share this information. Here, we consider not only the efficiency of collaboration by information sharing for the whole supply chain, but also the efficiency for each company. That is, we study selfish companies' behaviours versus the minimization of the overall cost with game theory. The analysis of simulation outcomes with game theory shows that collaborating is a Nash equilibrium, while non-collaborating is not a Nash equilibrium. Therefore, every company prefers collaboration, because its logistics cost increases when it ceases unilaterally to collaborate.

Finally, our last contribution to supply chain management is a broad, critical review of literature about the bullwhip effect. In particular, we introduce the bullwhip effect as a particular case of stream fluctuations in distributed systems, and we present some studies of the bullwhip effect in several fields, such as inventory management, economics, traffic flow theory, control theory... We now present contributions of this dissertation for the multi-agent field.

1.4.2 Contributions to Multi-Agent Systems

Multi-Agent Systems allow us to design, simulate and analyze our collaboration strategies, but we also contribute to this field by doing our research. In particular:

1. We have proposed a *decentralized coordination mechanism* based on communication to stabilize linked streams in a distributed system;
2. We have taken into account *both the global efficiency of the system and individual agents' incentives* to evaluate our coordination mechanism by applying concepts from game theory. This point was addressed with a methodology involving game theory to analyze simulation outcomes;

3. We have used game theory and multi-agent systems in a complex realistic study of a problem from supply chain management, and to our knowledge, this is the first time that these three fields have been applied in a common approach.

4. We have written a broad, critical review of literature about the use of multi-agent systems in supply chain management.

5. We have compared coordination in supply chain management and coordination in multi-agent systems.

Before we see these three main contributions for multi-agent systems, we recall that the bullwhip effect is a distortion of demand information when this information is transmitted as orders along the supply chain up to the most upstream suppliers. Therefore, we can note that this deformation of information not only interests supply chain management, but also Computer Science, because Computer Science studies information processing, and thus, information distortion.

From a less general point of view, this thesis studies the *dynamics in a complex distributed system* that can be viewed as a multi-agent system. In particular, the performance of many systems is reduced by fluctuations in their internal streams, e.g., traffic density during computerized network and road congestions. As a distributed system, a multi-agent system may face this problem too. For this reason, reducing the bullwhip effect in supply chains may give some hints for fluctuation reduction in other kinds of distributed systems, in particular when these systems are modelled as multi-agent systems. Therefore, the two above principles may be adapted by replacing some words, e.g., the bullwhip effect can be replaced by system instability, e.g., traffic congestions, market consumption by system input, company by agent, ordering scheme by agents' behaviour... With these replacements, the first principle 'Lot-for-lot orders eliminate the bullwhip effect' can be translated into 'If agents do exactly what they are asked to, according to the system output, the system fluctuations will be eliminated' and the second principle 'Companies should react only once to each market consumption change' into 'Agents should react only once to each change in the system input'.

Of course, like in supply chain management, these two principles only hold for fluctuations incurred by delays in the system, and these principles have to be adapted if other causes of fluctuations also occur. But these principles are general, because delays may also cause fluctuations in many other types of distributed systems than supply chains or multi-agent systems. For example, if all cars on a highway were decelerating and accelerating at the same rate and as soon as the first car, the road stream would be smoother, which would reduce the number of traffic jams.

Next, the agents' behaviour based on these two principles can be seen as a decentralized coordination mechanism based on communication. The problem lies in the fact that agents are selfish, and have therefore the choice of obeying or not to this coordination mechanism. This is the reason why the second contribution of this thesis is to take into account both individual and common interests. This is achieved through our third contribution consisting in the use of game theory to analyze inter-agent interactions. This methodology can easily be applied to analyze other kinds of multi-agent systems. The advantage of game theory is to give a high point of view on local interactions in a multi-agent system. Precisely, this high point of view allows researchers to immediately identify bad and good agents' behaviours before they implement a multi-agent system. In our case, we do not know how companies choose their ordering scheme, but we know which one they may eventually choose, because they correspond to a Nash equilibrium.

Moreover, game theory allows understanding why local optimizations do not lead to a global optimum, and how to add social rules constraining agents such as to reach this optimum. In this thesis, game theory is used to analyze a huge quantity of simulation outputs. Each simulation computes company's costs in the supply chain when each company decides to use a specific ordering scheme. Next, the application of some game-theoretic concepts such as dominations, Nash equilibria or Pareto-dominations causes relevant simulation outputs to appear, and therefore, the inter-agent dynamics in the multi-agent system is made clearer.

Finally, our approach is supported by a broad review of literature about the use of multi-agent systems in supply chain management. In particular, this review allows us to compare our work with some others, and also, to compare coordination in supply chain management with coordination in multi-agent systems. Some of the possible comparisons of our work with others are described in the next section.

1.5 Related Work

We first compare our work with other studies of the bullwhip effect with multi-agent systems. In fact, the nearest approach is from Kimbrough et al. [2002] who use a multi-agent system and a genetic algorithm to find a good ordering scheme for Sterman [1989]'s Beer Game. It is close to our approach because it used multi-agent concepts with a model of company from the Beer Game. From this point of view, the only difference is in the structure of the supply chain, straight in the Beer Game versus divergent in the QWSG. However, there are many other differences. The main one is that they fix an ordering scheme and the genetic algorithm looks for the optimal value of the parameter in this scheme. This optimal value is calculated so that the overall cost of the supply chain is minimized for the duration of the simulation. The optimum may therefore change depending on the simulation duration. On the other hand, our work focuses on the design of the ordering scheme itself, and we do fewer efforts to find optimal parameters, because we rely on the Solver in Excel from Microsoft Corp. [2004a], while they implement an optimization algorithm (genetic algorithm) by themselves. Conversely to their work, our ordering schemes are not designed to be efficient, but to stabilize ordering flows and inventories. We focus on efficiency only after this, when we optimize parameters in our two ordering schemes, but this is the essence of our approach. Though, these parameters are determined like Kimbrough and his colleagues, that is, they are set up such as the overall supply chain cost is a minimum for a specific duration of the simulation (but not for any duration).

The second approach using multi-agent systems is from Yung and Yang [1999a,b] who, like us, seek to reduce the bullwhip effect by improving the visibility and taking constraints into account. Instead of basing their model on the Beer Game, they use a model very close to the reality that can be operated by managers in an actual supply chain. Conversely to our higher level approach, they model processes in companies, but the supply chain considered in their experiments has only two levels (manufacturers and warehouses, where customers buy from these warehouses), while we consider four levels (retailers, wholesalers, paper and sawmill, and forest). Conversely to our agents applying an ordering scheme, their agents are more complex because they are able to:

- process routine jobs in place of humans on real installations;
- communicate information about operations, and use this information to optimize decisions with genetic algorithms;
- give managers a view of the state of the whole supply chain state, and in particular, a detection of the bullwhip effect, and an extraction of useful information received from other companies by a technique of constraint propagation.

The last approach with multi-agent systems is Yan [2001]'s study of the impact of delay distribution on the bullwhip effect. As it focuses on impact of delays in the supply chain, this work is very similar to ours. Basically, this agent-based supply chain checks experimentally that Chen et al. [2000]'s formal quantification of the bullwhip effect is correct. In particular, they verify like us that information centralization improves the efficiency of the overall supply chain.

We now compare our work with research studying the impact of delays on the bullwhip effect. All the following work focuses on the difficulty of forecasting future demand when delays are longer, because the forecast has to be made on a longer horizon, conversely to our view, in which the bullwhip effect is also increased when there are no forecasts, because delays are a cause of the bullwhip effect by themselves. In this context, the approach from Chen et al. [2000], mentioned above, proposes a formal model in which the bullwhip effect is both induced by a forecasting technique, a moving average, and aggravated by delays. This paper was extended for another forecasting technique, an exponential smoothing, by Ryan [1997] and Simchi-Levi et al. [2000]. Lee et al. [1997a,b] also explain how forecasting techniques induce the bullwhip effect, and how this is aggravated when delays are longer.

In general, such models are based on the Beer Game. As we use a simulation model derived from this game, we do so, but in a more empirical way. The Beer Game was studied by Sterman [1989] with human players, while we replace people by software agents. Therefore, we do not focus on players' psychology that cannot understand the whole dynamics of the supply chain. Nevertheless, we also have to take such dynamics into account when proposing the ordering schemes used by our agents. As presented in this thesis, the design of our ordering schemes addresses the issue of delays in supply chains, because it is the only chain dynamics explaining the apparition of the bullwhip effect in the QWSG and the Beer Game played by software agents. From this point of view, we give a better understanding of Sterman's "supply chain dynamics", so that players should know how to play the Beer Game more efficiently. On the other hand, issues related to delays are very well known in supply chain management. In particular, Sjöström [2001] studied their impact in the North European forest industry. This work is similar to ours because it addresses a forest industry, but ours is less "empirical" as it uses simulation rather than real-life.

The last point of comparison that we consider is information sharing in supply chains, which has been well studied [Anderson and Morrice, 2000; Cachon and Lariviere, 1999; Chatfield, 2001; D'Amours et al., 1999; Lee and Whang, 1998; Yu et al., 2001]. In fact, we propose to share information to reduce the bullwhip effect induced by delays, but information may be shared for other reasons. In our proposition of a mechanism for sharing information, we were inspired by Porteus [2000]'s Responsability Tokens to design our solution, even if our ordering scheme is eventually very different from this mechanism. In fact, Responsability Tokens deal with transfer payments in the supply chain to give incentives to upstream suppliers to ship products required downstream by retailers. On the contrary, our ordering scheme does not rely on money, because its only objective is to give enough information to companies, so that they can stabilize orders without neglecting operational constraints, and in particular inventory management. In fact, information sharing is the most often proposed solution to the bullwhip effect [Chen et al., 2000; Lee et al., 1997a; Simchi-Levi et al., 2000]. In this context, our two principles improve the understanding of why and how to share information to reduce the bullwhip effect.

Another type of information sharing, supplier capacity information, is studied by Swaminathan et al. [1997]. But from a more general point of view, two subfields of Economics can be applied to study information sharing. The first subfield is Information Economics which studies the "consequences for the character and the efficiency of the interaction between individuals or organizations when one party has more or better information on some aspect of the relationship". This is the condition of asymmetric information, under which the information gap will be exploited if, by doing so, the better-informed party can achieve some advantage" [MacHo-Stadler et al., 2001]. In particular, Cachon and Lariviere [1999] studied the value of information in supply chains. To illustrate information economics, let us assume that using our schemes was not Nash equilibria for the supply chain. In such an event, we could have given a price to the market consumption information. The question would have been, what minimal price would be demanded by retailers to accept to broadcast the market consumption information, and what maximal price other companies would agree to pay for this information. Depending on these minimal and maximal prices, we would have been (or not) able to fix a price to the market consumption information, such as our schemes become Nash equilibria.

The second economics subfield that can be applied to supply chain management is game theory used to study incentives [Umbhauer, 2002]. In particular, Cachon and Netessine [2003] gave an overview of such studies and notice the "recent explosion of game-theoretic papers in supply chain management". Our work belongs to this type of work, except that we replace the analytical model by a multi-agent model that we simulate, as suggested by papers in the multi-agent system field [Boutilier et al., 1994; Rosenschein and Zlotkin, 1994; Sandholm, 1999]. This approach gives us the ability to solve more complex sets of relations.

1.6 Chapter Layout

This dissertation is organized in nine chapters. Chapters 1, 2, 3, 4 introduce the background of this dissertation, and in particular our presentation of the bullwhip effect as an issue of stream fluctuations and our synthesis of literature about multi-agent systems in supply chain management, while Chapters 5, 6, 7, 8 and 9 present our contributions.

Chapter 1, i.e., the current chapter, introduces the context of this research, the bullwhip effect and some incentive-related issues of possible solutions to this phenomenon. The research is then motivated through the presentation of the bullwhip effect as a particular case of stream fluctuations in a distributed system. These fluctuations are difficult to reduce because of the complex dynamics of the considered system, i.e., the supply chain in our case, and because of the selfish behaviour of its nodes, i.e., the companies in the supply chain. This research has two successive parts: first, the proposition of two principles for reducing the bullwhip effect and the validation of their effectiveness, next the study of individual incentives for their use by companies. After this motivation, contributions for supply chain management and for multi-agent systems are presented for these two parts. Finally, this dissertation is compared with some other approaches

Chapter 2 reviews the literature on the background of this work. The field of supply chain management is first presented, next the area of multi-agent systems. Finally, we synthetise some work applying agents to supply chains.

Chapter 3 focuses the literature review on the bullwhip effect. Precisely, we propose to see this problem as a special case of stream fluctuations occuring in supply chains. Then, this problem, its consequences and its known causes and solutions are presented. Finally, some studies from different fields conclude this chapter.

Chapter 4 concludes the literature review by introducing interactions modelled with game theory, for their use in Chapter 8, and illustrates them with some classic games from game theory.

Chapter 5 proposes seeing ordering and shipping delays as a cause of the bullwhip effect. Two basic principles are next proposed to solve this cause, and these principles are instanciated in two ordering schemes. As the bullwhip effect is a coordination problem, these two ordering schemes are also some decentralized coordination mechanisms. Next, the behaviour of the supply chain under one of these two mechanisms is described. Finally, an example illustrates the use of these two ordering rules with the case of a company buying and producing different types of products.

Chapter 6 introduces the simulation model used for the validation of the two principles introduced in Chapter 5. This simulation model is based on the QWSG (Québec Wood Supply Game) which is first described. A strict implementation of this model is next outlined, and detailed in Appendix A. Finally, a more realistic implementation of this model is detailed. The first simulation is used in Chapters 7 and 8, while the second is still *under development*.

Chapter 7 presents the first series of experiments carried out with our first simulator to validate the efficiency of the two proposed principles for reducing the bullwhip effect generated by ordering and shipping delays. Technical details about this chapter are in Appendix B. The main metrics is the order variability, but three other metrics are also considered. We note here the change in backorder duration distribution between the classic situation with the bullwhip effect, and the situation incurred by our ordering schemes.

Chapter 8 presents the second series of experiments carried out with our first simulator, and technical details about this chapter are in Appendix C. Before this presentation, we introduce the method of cost evaluation that includes parameters from the Québec forest industry. Parameters of ordering schemes are optimized according to this cost. Next, game theory concepts are adapted to our simulation. Finally, the second series of experiments checks if all companies have incentives for following these two principles. Specifically, simulations are carried out and costs calculated with parameters from the Québec forest industry are analyzed with the adapted game-theoretic concepts.

Chapter 9 summarizes the two parts of this thesis, that is, the proposition and the validation of two decentralized coordination mechanisms for reducing the bullwhip effect and the study of individual incentives for using such mechanisms. We discuss next both our methodology to reduce the bullwhip effect that is summarized by our two principles, and our methodology to verify that agents have invididual incentives to apply our two principles. Finally, some future work is outlined.

Chapter 2

Supply Chain Management and Multi-Agent Systems: Background

The previous chapter has outlined the context of our research, that is, the problem of stream fluctuations in distributed systems. Since the focus in this dissertation is on supply chains and on multi-agent systems as two special cases of distributed systems, this chapter now presents these two fields.

More precisely, we study ways to reduce the bullwhip effect through multi-agent simulations. Consequently, we first present supply chains. To this end, supply chain management is first introduced as a business practice to plan and synchronize operations within a network of firms, by providing the concept of inter-company collaboration. We next present a classic model used in supply chain management, which is called Economic Order Quantity (EOQ), to quantify orders. EOQ is used in our model to set the parameters of the (s, S) ordering policy used in our simulation. These ordering policies and parameters are introduced in Section 2.1.

As previously stated, we use intelligent agents to model companies in supply chains. The concept of "agent" is first defined, and next compared with another concept from Computer Science, the concept of "object". After that, the general agent architectures outline the different levels of agent sophistication. Then, we motivate the use of multi-agent systems, and we compare these systems with some other "scientific" approaches. Finally, we illustrate this section with some examples involving multi-agent systems in different fields or applications. This presentation of multi-agent systems is in Section 2.2.

A synthesis of supply chain management and multi-agent systems extend the previous illustrations of agents applied in different fields. For that purpose, agents are first introduced as a new information technology for supply chain management. The arguments pro agents outlined in Section 2.2 are next extended for the special case of agents in supply chains. Finally, some projects applying agents to supply chains illustrate this section. This synthesis of supply chain management and multi-agent systems is detailed in Section 2.3.

2.1 Supply Chain Management

From a general point of view, the bullwhip effect is a particular case of the problem of stream fluctuations in a distributed system. From the particular point of view of supply chain management, the bullwhip effect is a problem among other industrial problems. After the presentation of some of these other problems, we introduce the concepts of supply chains and collaboration in such chains as solutions to these issues. As the bullwhip effect is the amplification of order variability in supply chains, this effect is related to order placement. The traditional Economic Order Quantity (EOQ) model is often used to define the quantity to be ordered to a supplier. Hence, we then introduce this model.

2.1.1 Background

Industrial Problems

First, companies face a huge number of problems, such as how to make decisions concerning production planning, inventory management and vehicule routing. These three decisions are managed separately in most organizations because taking each individual decision is very difficult, since many constraints have to be satisfied (production, shipping and inventory capacities, precedence order on activities, legal obligations, etc.) [Mallya, 1999]. For instance, the multistage, multicommodity inventory management problem and the vehicule routing problem are both known to be NP-hard problems ([Savelsbergh, 1985] cited by Mallya [1999]), i.e., very difficult.

Second, the problem is yet harder in reality because the decisions concerning production planning, inventory management and vehicule routing are *interdependent*. Hence, these three decisions should be taken together, which makes the planning problem harder.

Third, companies are not isolated, but impact on and are impacted by their partners. As a result, when a company maximizes its profits, it may disturb other companies, which may result in globally underoptimal decisions, because organizations may have different conflicting objectives [Simchi-Levi et al., 2000, pp. 3]. The best solution would be to make the decisions together concerning production planning, inventory management and vehicule routing for several companies. As this planning problem is hard for a single company, considering all companies decisions together is very hard. The concept of supply chains was proposed to address this problem of minimization of total supply chain cost, while meeting fixed and given demand by points-of-sale, e.g., by retailers [Shapiro, 2000, pp. 8].

The Concept of Supply Chains as a Solution

Let us now focus on “supply chains”, which is a quite old term; we do not have found the first definition of this term, but we have found, for example, that Burns and Sivazlian [1978] referred it in 70’s. According to Muckstadt et al. [2001], there are many definitions and interpretations of the term supply chain management. These authors defined a supply chain as “the set of firms acting to design, engineer, market, manufacture, and distribute products and services to end-consumers”. In general, this set of firms is structured as a network, as illustrated in Figure 2.1 [Swaminathan et al., 1998; Davidsson and Werstedt, 2002]. In the same context, Shapiro [2000] noted that “supply chain management is a relatively new term that crystallizes concepts about integrated business planning that have been espoused by logistics experts, strategists, and operations research practitioners as far back as the 1950s”. Similarly, Simchi-Levi et al. [2000] defined this term as “a set of approaches utilized to efficiently integrate suppliers, manufacturers, warehouses, and stores, so that merchandise is produced and distributed in the right quantities, to the right locations, and at the right time, in order to minimize systemwide costs, while satisfying service level requirements”. Poirier and Reiter [2001] noted that the concept of supply chains improves the competitive position of collaborating companies, because it supports the creation of synergies among these companies. In particular, such synergies are due to the fact that a supply chain is a system, and as a consequence, this system is superior to the sum of the constituting companies. As previously explained, the concept of inter-company *collaboration* is a way to create such synergies in a supply chain.

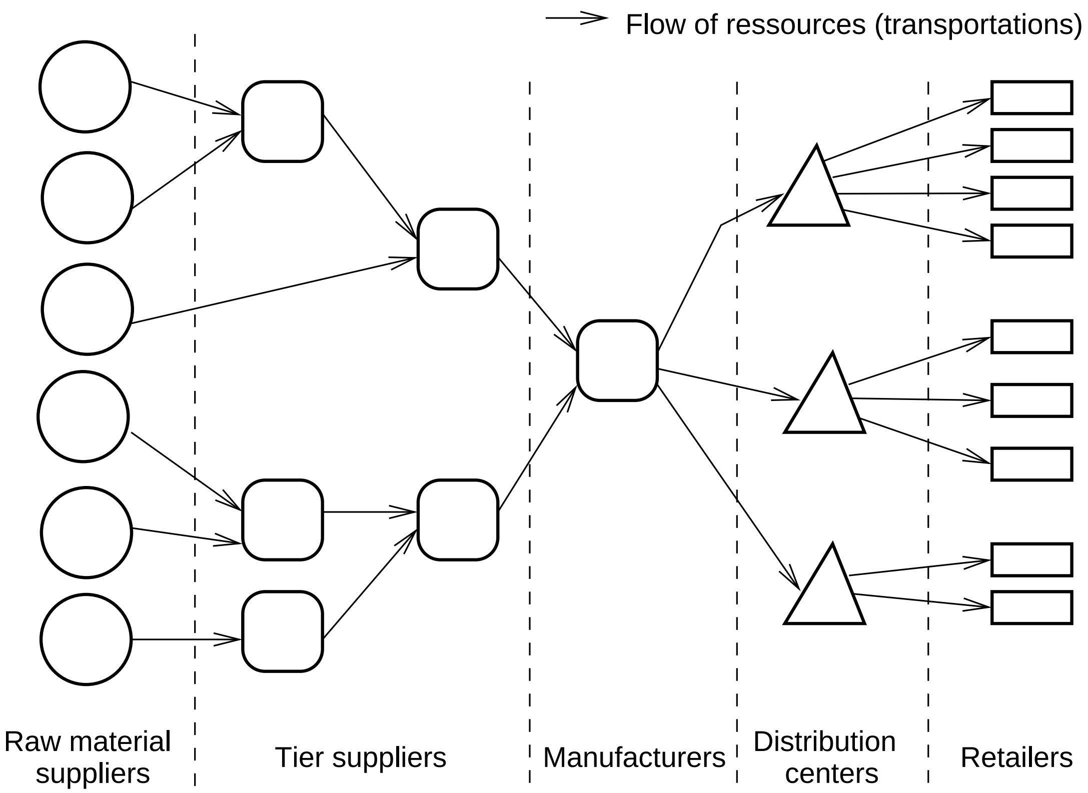


Figure 2.1: A supply chain [Swaminathan et al., 1998; Davidsson and Werstedt, 2002].

Collaboration in Supply Chains

Muckstadt et al. [2001] noticed that the term *collaboration* is confusing because it has taken on several interpretations when used in the context of supply chain management. Various levels of collaboration techniques based on *information sharing* were set up in real supply chains. It is important to note that we always refer to collaboration as information sharing in this dissertation, even if collaboration is in general wider than only information sharing. We represent in Figure 2.2 how some of these information sharing techniques overlaps. These techniques are essentially information centralization, Vendor Managed Inventory/Continuous Replenishement Program, and Collaborative Planning Forecasting and Replenishment. They are now reviewed in detail.

Information centralization: This is the most basic technique of information sharing in which retailers broadcast the market consumption (approximated as their sales) to the rest of the supply chain. As we also refer in this dissertation to information centralization, it is necessary to distinguish information sharing from information centralization: the latter is a particular case of the former, because

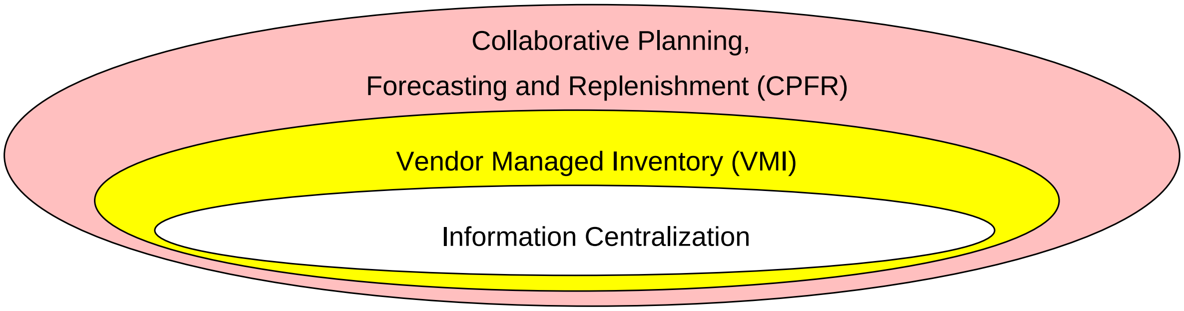


Figure 2.2: Overlap of some collaboration techniques.

information centralization is the multi-casting in *real-time* and *instantaneously* of the market consumption information, while information sharing is only the sharing of the demand information between any companies[1], and includes thus information centralization. Chen et al. [1994] formally showed, for two forecasting methods, that information centralization reduces the bullwhip effect. In fact, information centralization reduces the bullwhip effect, because each level of the chain can base its forecasts on the actual market consumption, instead of basing them on incoming orders, which can be much more variable than the actual market consumption [Lee et al., 1997a,b]. Among the two ordering schemes that we propose in this thesis, we see in Chapter 7, that the one using information centralization has the best results, which confirms Chen et al. [1994]'s results.

It is also necessary to distinguish information centralization from centralized coordination: the latter assumes there is a central coordinator that knows and controls everything in the supply chain, while information centralization only assumes retailers announce all their sales in real-time.

Vendor Managed Inventory (VMI) [John Taras CPIM, 2003] and **Continuous Replenishment Program (CRP):** These two collaboration techniques are very similar, but are used in different industries. The idea is that retailers do not need to place orders because wholesalers use information centralization to decide when to replenish them. Although these techniques could be extended to a whole supply chain, current implementations only work between two business partners. In fact, many customers are attracted to these techniques, because they mitigate uncertainty of demand, a consequence of the bullwhip effect. Moreover, the frequency of replenishment is usually increased from monthly to weekly (or even daily), which benefits both partners. These techniques were popularized in the late 1980's by Wal-Mart Stores, Inc. [2004] and the Procter & Gamble Company

---

[1] We only consider in this dissertation the demand information sharing, but in general, companies can also share other kinds of information, e.g., their available production capacity, their inventory level. . .

[2004]. In particular, VMI has became one of the key element of the quick response program in the grocery industry [Waller et al., 1999].

Collaborative Planning, Forecasting and Replenishment (CPFR): This technique developed by the VICS Association [2003] (Voluntary Interindustry Commerce Standards) is a standard that enhances VMI and CRP by incorporating joint forecasting. Like VMI and CRP, current implementations of CPFR only include two levels of a supply chain, i.e., retailers and their wholesalers. With CPFR, companies electronically exchange a series of written comments and supporting data, which includes past sales trends, scheduled promotions, and forecasts. This allows the participants to coordinate joint forecasts by focussing on differences in forecasts. Companies try to find the cause of such differences and agree on joint, improved forecasts. They also jointly define plans to follow when specific contingencies occur [Simchi-Levi et al., 2000, pp.239].

It is worth mentioning that we propose in this disseration a new coordination mechanism that could benefit to these collaboration techniques. More precisely, the ideas to reduce the bullwhip effect (that we call our two principles) could enhance VMI/CRP and CPFR by providing them with insights on this effect.

Supporting technologies

Because there exists other models, these three techniques of information sharing, i.e., information centralization, VMI/CRP and CPFR, can be supported by information technologies such as e-Hubs [Lee, 2001]. The basis of these information technologies is currently the Internet, but other technologies are also used, e.g., the protocol for Electronic Data Interchange (EDI). The first advantage of the Internet on every other technology is to provide a low-cost communication infrastructure available almost anywhere in the world. This first advantage allows companies to increase information streams, and more precisely in our context, to share more information. The second advantage of the Internet is to provide some standardized file formats, which reduce the cost of information technologies. We develop the goals of information technologies and the way to achieve these goals during the presentation of the application of multi-agent systems in supply chain management in Section 2.3.

2.1.2 Order Placement in Supply Chains: the EOQ Model

One of the most basic means of communication between companies is the placement of orders when a company needs to procure some products. This ordering decision may look very easy at a first glance, but it is the key problem in the bullwhip effect. In fact, the bullwhip effect is generated by the ordering policy used by companies in the supply chain. Nowadays, such ordering policy is based on mathematical models from inventory management. The goal of inventory management is to find the ordering quantity that minimizes the cost of an inventory system. Several approaches have been proposed in inventory management to address the range of possible contexts. For example, the Wagner-Whitin algorithm periodically reviews the optimal ordering quantity when demand is not constant. Such approaches are described in many books, e.g., Hax and Candea [1984], Johnson and Montgomery [1974] and Nahmias [1997]. Since we adapt the model of the Economic Order Quantity (EOQ) to our simulation in Section B.2, we now present this model. We recall here Johnson and Montgomery [1974]'s presentation of the EOQ model, which is in fact its extension into a *single-item model with static demand and permitted backorder*[2]. The following notations are used only to describe this model in the current subsection and in our adaptation to the QWSG in Subsection B.2 of Appendix B. These notations do not apply outside of these particular two subsections.

D = demand rate (units per year).

P = production rate (units per year).

A = fixed cost of a replenishment order ($ per order).

C = unit variable cost of a production (or purchase).

h = inventory carrying cost ($ per unit per year).

π = shortage cost per unit short, independent of the duration of the shortage ($ per unit short).

$\hat{\pi}$ = shortage cost ($ per unit short per year).

τ = replenishment lead time, the time between the placement and receipt of an order (years).

---

[2] It seems that, in comparison with the presented model, the basic EOQ model be the same, but without production ($C = 0$ and $P \to \infty$) and backorder ($\pi = \hat{\pi} = 0$, and thus, the decision variable $b = 0$).

Q = order quantity (units per order).

s = reorder point, i.e., inventory level at which an order is placed (units in inventory).

I_{max} = maximum on-hand inventory level (units in inventory).

b = maximum backorder level permitted (units missing in inventory).

T = cycle length, i.e., the length of time between placement (or receipt) of replenishment orders (years).

K = average annual cost which is a function of the inventory policy ($).

Based on these notations, we can now recall the EOQ model. The advantage of the formulation presented below is its generalization, because it only requires simplifications to be adapted to many situations. Figure 2.3 represents the most general behaviour of the considered inventory system, as described now. When a shipping arrives, inventory position increases at a rate $P - D$ during T_1 and T_2. When production has finished[3], inventory position stops increasing and starts decreasing at a rate D during T_3 and T_4. Inventory carrying costs are incurred in T_3 and T_4, while backorder costs are incurred in T_1 and T_4. The decision variables are the order quantity Q and the maximum backorder level b. The average cost per cycle is the sum of the procurement, inventory, and shortage costs during the cycle.

There are D/Q identical cycles in a year, because there are lesser placed orders, and thus, lesser cycles, when the quantity ordered Q increases. The average annual cost K of the considered inventory system is the multiplication of the average cost per cycle by the number of cycles D/Q (see Equation 2.1 and its details in Equations 2.2, 2.3, 2.4, 2.5 and 2.6).

$$K(Q,b) = \frac{AD}{Q} + CD + \frac{h[Q(1-D/P)-b]^2}{2Q(1-D/P)} + \frac{\hat{\pi}b^2}{2Q(1-D/P)} + \frac{\pi bD}{Q} \tag{2.1}$$

where:

$$\frac{AD}{Q} = \text{annual ordering cost} \tag{2.2}$$

$$CD = \text{annual production (or purchase) cost} \tag{2.3}$$

---

[3] When the company produces nothing, e.g., a retailer, we have $P = 0$, and thus, $T_1 = T_2 = 0$. In other words, the inventory level spring from $-b$ to I_{max} instantaneously, because the inventory considered in the model EOQ contains products available to shipping, and not raw material.

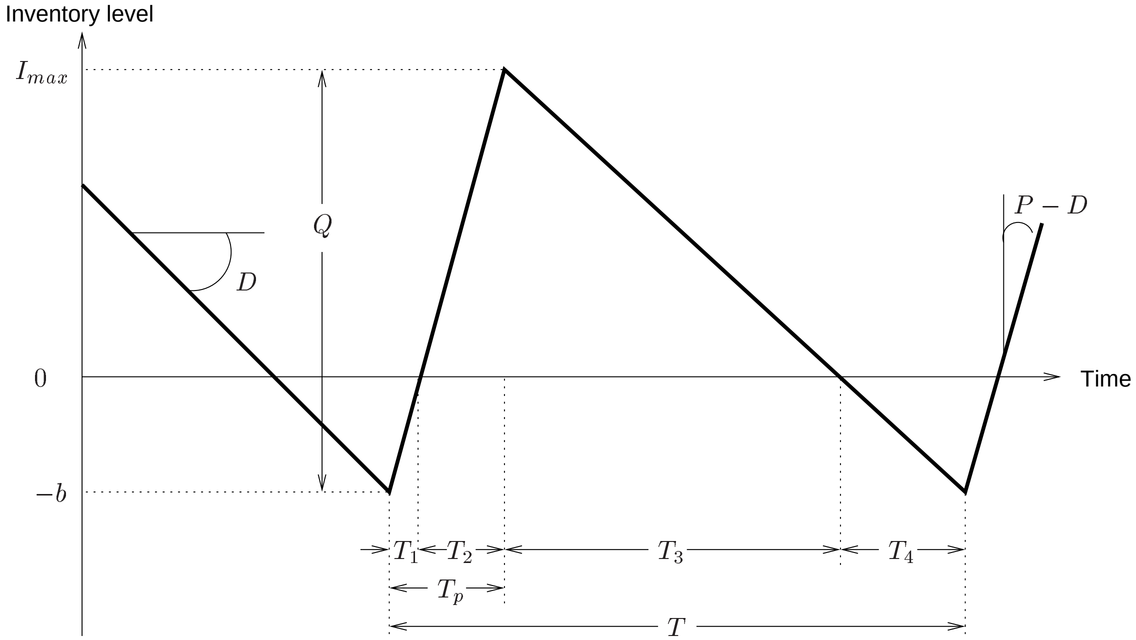


Figure 2.3: A cycle for the inventory system [Johnson and Montgomery, 1974].

$$\frac{h[Q(1-D/P)-b]^2}{2Q(1-D/P)} = \text{annual inventory carrying cost} \tag{2.4}$$

$$\frac{\hat{\pi} b^2}{2Q(1-D/P)} = \text{annual shortage cost} \tag{2.5}$$

$$\frac{\pi b D}{Q} = \text{cost of having some backorders during the year} \tag{2.6}$$

The decision variables Q and b are set up by solving the two simultaneous equations given by Equation 2.7.

$$\frac{\partial K}{\partial Q} = \frac{\partial K}{\partial b} = 0 \tag{2.7}$$

The general solutions to this problem are given in Equations 2.8 and 2.9:

$$Q^* = \sqrt{\frac{2AD}{h(1-D/P)} - \frac{(\pi D)^2}{h(h+\hat{\pi})}}\sqrt{\frac{h+\hat{\pi}}{\hat{\pi}}} \tag{2.8}$$

$$b^* = \frac{(hQ^* - \pi D)(1-D/P)}{h+\hat{\pi}} \tag{2.9}$$

Finally, this model is used to set optimal parameters s and S in a (s, S) ordering policy (i.e., when inventory level x falls below s, the company orders $(S - x)$): $S = s + Q^*$, where s is given by Equation 2.10 (note that $m = [\tau/T]$, where $[x]$ is the integer part of x, e.g., $[1.9] = 1$):

$$s = \begin{cases} \tau D - mQ - b & \text{if } \tau - mT \leq T_3 + T_4 \\ ((m+1)(Q/D) - \tau)(P - D) - b & \text{else} \end{cases} \tag{2.10}$$

More details about this formulation of the EOQ model can be found in Section 2-2 of Johnson and Montgomery [1974]'s book. As we can see, the goal of this model is to optimize logistic costs for a single company. Unfortunately, this model is ineffective in a supply chain, because local optimizations with the model EOQ ignore other companies, which can lead to inefficiencies at the supply chain level. In particular, it was proven that such local optimizations without global vision induce the bullwhip effect [Kahn, 1987; Naish, 1994; Shen, 2001; Chen et al., 2000; Simchi-Levi et al., 2000]. In order to be able to consider a whole supply chain in this dissertation, we do not rely on a mathematical model, but on agent-based simulations. We also try to coordinate the supply chain as a multi-agent system, instead of optimizing costs. Hence, multi-agent systems are now introduced.

2.2 Multi-Agent Systems

We now focus on the second field addressed in this work, i.e., multi-agent systems. We first define the concept of agents, next we compare this concept with another concept from software engineering, i.e., the concept of object. Then, we outline some architectures of agents and some arguments in favour of the use of agents in general. Finally, we illustrate projects involving multi-agent systems in different areas.

2.2.1 The Concept of Agents

Intelligent agents are a new paradigm of software system development. They are used in a broad and increasing variety of applications [Chaib-draa et al., 2001, 1992; Chaib-draa, 1995]. For a long time, there was no single definition of an agent and a multi-agent system: several definitions have cohabited in the past [Ferber, 1995]. Nowadays, it seems that researchers agree on the following definition proposed by Wooldridge and Jennings [1995]:

> The term "agent" denotes a hardware or (more usually) software-based computer system, that has the following characteristics:
>
> **Autonomy:** agent operates without the direct intervention of humans or others, and has some kind of control over its actions and internal state;
>
> **Social ability:** agents interact with other agents (and possibly humans) via some kind of agent-communication language;
>
> **Reactivity:** agents perceive their environment, (which may be the physical world, a user, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;
>
> **Pro-activeness:** agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative.

2.2.2 Comparison with Objects

Based on this concept of agent, Shoham [1993] proposed a new programming paradigm called Agent-Oriented Programming (AOP) to replace the current Object-Oriented Programming (OOP). The difference between agents and objects is sometimes missed by programmers familiar with object-oriented languages as C++ [Free Software Foundation, 2004a] or Java [Sun microsystems, 2003]. The main difference between these two concepts is the *autonomy* of agents. In fact, while objects encapsulate some state on which their methods can perform actions, and in particular the action of invoking another object's method, an object has control over its behaviour. That is, if an object is asked to perform an action, it always does so, while an agent may refuse. Concerning this point, Wooldridge [2001] recalls the slogan "Objects do it for free; agents do it because they want to".

Of course, some sophisticated objects may be very similar to agents. In fact, Wooldridge [1999] noted that there are clear similarities, but obvious differences also exist. Let us consider the case of objects in Java that can easily be transformed into threads exhibiting some behaviour. Such active objects have some autonomy like agents, but their behaviour is only procedural in reaction to message requests. On the other hand, autonomy of agents make them perform activities without external intervention [Guessoum and Briot, 1999]. In short, object-based concurrent programming has some relationships with distributed artificial intelligence [Gasser and Briot, 1992].

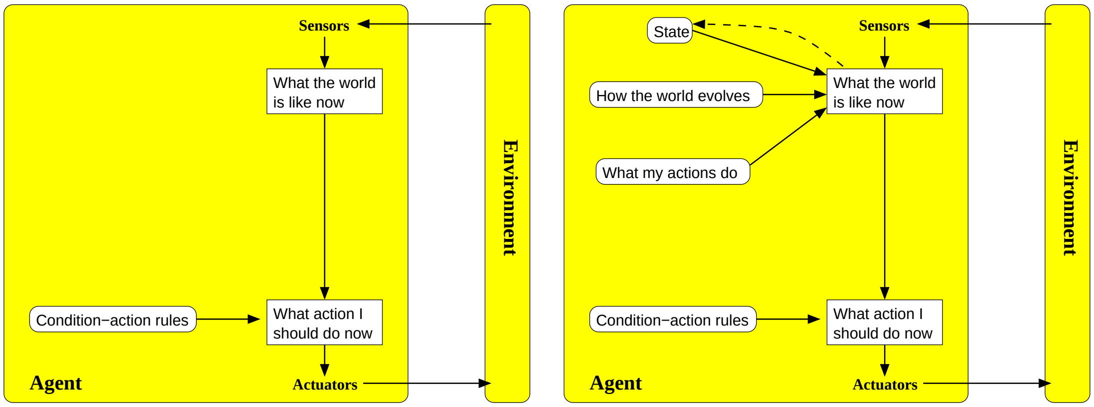
But objects and agents also present differences. In particular, object state is much simpler than agent state. In fact, an object state is only a data structure, i.e., an aggregation of variables of different types (integers, booleans, character strings...) in a common structure, while an agent state consists of components such as beliefs, decisions, capabilities and obligations. As an agent state is more sophisticated, it is also referred to as a mental state [Shoham, 1993].

Finally, it is important to note that agents have been programmed in C++ or Java, i.e., with an OOP language, but AOP languages have appeared. For example, JACK™ designed by the Agent Oriented Software Group [2003] (Melbourne, Australia) is an AOP language. This language implements concepts from AOP upon a OOP languages. That is, JACK™ provides an AOP compiler transforming JACK™ code into Java code. Next, the JACK™ compiler calls the Java compiler to transform the generated Java code into a runnable Java bytecode that works on any Java Virtual Machine. We will describe JACK™ further in Subsection 6.3.1.

2.2.3 Agent Architectures

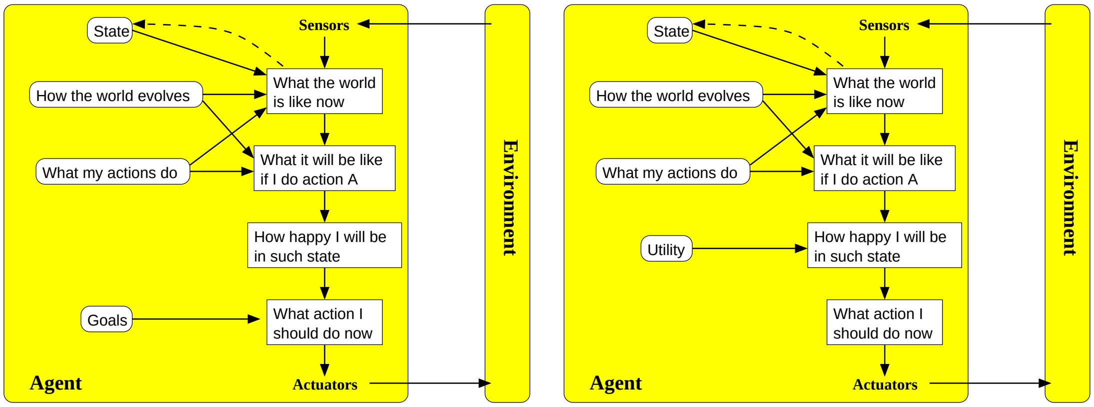
In the same manner that there are several languages to implement agents, there are also different levels of complexity of this implementation. Such complexity depends on the task that agents have to carry out and on the environment surrounding them. Russell and Norvig [2003] propose the following classification of agent architectures:

- *Simple reflex agents*: This type of agent is the simplest, because agents act on the basis of their current perceptions, ignoring what has occurred in the past, because they have no memory. Figure 2.4(a) describes how they select their actions according to condition-action rules, e.g., if sensors state that it-is-raining then actuators do take-umbrella. We use this type of agent in Chapters 7 and 8.
- *Model-based reflex agents*: As agents cannot perceive their whole environment, model-based reflex agents, presented in Figure 2.4(b), keep track of the part of their environment they cannot currently observe. To achieve this, they have an internal representation of their environment, called a “model of the world”. Like simple reflex agents, they select their action according to condition-action rules, but now, the condition only depends on the model of the world, and not on the current perception from Sensors. We do not use this type of agent, because our agents do not have any model of their world. Note that this world would be modelled in company-agents by some forecasting techniques predicting the future state of their environment, i.e., their future incoming demand.



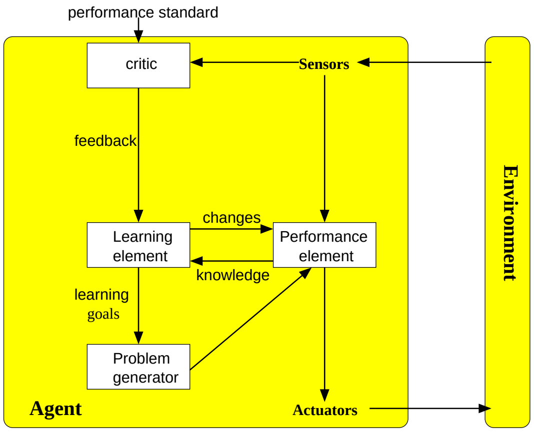
(a) ***Schematic diagram of a simple reflex agent.***

(b) ***A model-based reflex agent.***



(c) ***A model-based, goal-based agent.***

(d) ***A model-based, utility-based agent.***



(e) ***A general model of a learning agent.***

Figure 2.4: Architectures of intelligent agents [Russell and Norvig, 2003].

- *Goal-based agents*: As illustrated in Figure 2.4(c), this type of agent has goal information describing desirable situations, because the current state of the model of the world is not always enough to select an action efficiently. Conversely, to the two previous agent types, condition-action rules are no longer used, because the agent considers the possible futures of the world (cf. "What it will be like if I do action A" in Figure 2.4(c)) to decide which action it should do to achieve its goal.

- *Utility-based agents*: In order to improve the quality of agent behaviour, the agent is given in Figure 2.4(d) a utility function mapping its state (or a sequence of states) in the model of the world, onto a real number describing the associated degree of agent's happiness. In comparison with goal-based agents, utility-based agents do not decide which action to do in order to achieve a goal, but which action to do to increase utility. This difference implies that both types of agents find which actions to do to achieve their goals, but utility-based agents find the best actions according to some given metrics. This agent architecture is hence the nearest to the definition of Economics agents, that only maximize their utility.

- *Learning agents*: Turing [1950] has noted the huge amount of work it takes to program an intelligent machine, and has concluded that it would be easier to build learning machines and then to teach them. Another advantage of learning agents is their adaptability to unknown environments, and the improvement of their behaviour with time. The learning agents presented in Figure 2.4(e) use a feedback, called critic, to learn which perceptions of the environment are desirable, and in consequence, how to behave. Precisely, agents' learning consists in improving their future performance based on their past critic, by optimizing their behaviour such as to maximize their utility when the world continues evolving as it has been. This kind of learning makes agents discover that some kind of (but not exactly) condition-action rules always do the same thing, based on their current knowledge.

  A problem arises here: after some learning time, agents are always going to do the same things because of these discovered rules, though the agents are not sure that these actions are optimal, while they might have a better performance if they had a wider knowledge of their environment. In fact, they should try to do very different actions than those prescribed by their learning process. This *exploration* of new actions is insured by the problem generator.

2.2.4 Motivations for Multi-Agent Systems

Huhns and Stephens [1999] noted that multi-agent systems are generally less efficient than centralized solutions, because the distribution restrains optimization. But these authors also gave several advantages of multi-agent systems. First, multi-agent systems are easier to understand and implement, when the problem itself is distributed. This allows the multi-agent system to give more flexibility when taking into account the modularity of the real, modelled system. Next, the distribution may force programmers to propose new algorithms to solve problems. In particular, the concurrency can be used to accelerate problem solving. Finally, a centralized solution may be impossible, because systems and data are in independent organizations. We develop this latter argument in Section 2.3, because it is the main one in favour of multi-agent systems in supply chains.

Jennings [2000] pointed out the flexible, high-level interactions of agents, that make the engineering of complex systems easier. This author recalls that *complex* systems are always distributed, and from his point of view, agent decomposition is very important to manage complexity. It follows from this, that designers need a means to reduce the complexity of the system control, in order to enhance their ability to model, design and build complex, distributed systems. Multi-agent systems provide designers with this means through the control decentralisation. In particular, the system complexity makes it very difficult to know every possible interaction in the system, because the system only has partial control and observability over its environment, and thus, this environment is highly unpredictable. Multi-agent decentralisation takes this into account by letting each agent continuously coordinate its actions with other agents, instead of making this agent apply a behaviour prescribed at design time. In short, some advantages of multi-agent systems is the fact that modelling with agents:

- partitions the problem space of a complex system efficiently;
- is a natural way to modularise complex systems;
- focus on the organizational relationships in complex systems.

Similarly, Wooldridge [2001] says that *interaction* is now seen by most programmers as an important characteristic of complex softwares. For this reason, interactions, and thus multi-agent systems, take a growing part in software engineering. Moreover, multi-agent systems are an interdisciplinary field. For example, interactions in multi-agent systems are also interesting to model dynamics in human societies.

We should note that there are also objections to multi-agent systems. We now review such objections.

2.2.5 Differences between Multi-Agent Systems and Other Fields

In general, objections to multi-agent systems are due to their similarity with some other fields. To respond to these objections, Wooldridge [2001] points out the difference between this field and some others:

Distributed/concurrent system

- *Similarity*: By definition, multi-agent systems are a special case of distributed/concurrent systems. Therefore, experience in this field has to be kept by the multi-agent system community, in particular to avoid discovering again how to manage mutual exclusion over shared resources, how to avoid dead- and livelocks...
- *Differences*: First, agents are autonomous, and therefore, synchronization and coordination are not structured at design time, as they are in distributed/concurrent systems. In fact, agent synchronization and coordination is achieved at run time. Second, agents are in general self-interested, while components in a distributed/concurrent system have the common goal of maximizing the overall system efficiency. For these two reasons, negotiation is important in multi-agent systems, while it is unknown in distributed/concurrent systems.

Artificial intelligence

- *Similarity*: Historically, multi-agent systems were born from Distributed Artificial Intelligence, which is a subfield of Artificial Intelligence [Jennings et al., 1998].
- *Differences*: First, the main topic of artificial intelligence has been the study of components of intelligence (learning, planning, understanding images...), while the goal of research about agents is the integration of these elements. Therefore, during agent implementation, much more time is spent with computer science and software engineering, than with artificial intelligence. Second, social ability in systems has been ignored by artificial intelligence, while this is as important in an intelligent behaviour, as learning or planning.

Economics/Game Theory

- *Similarity*: Like multi-agent systems, these theories also deal with self-interested agents, and more precisely with their interactions. Some well-known reseachers have contributed to both computer science and economics/game theory, such as von Neumann and Turing. However, these two fields have been dissociated since these beginnings. Now, the situation is changing because game theory has more and more applications in multi-agent systems, and economists are interested in multi-agent simulations to understand inter-agent interactions.
- *Differences*: First, concepts in economics/game theory are descriptive, and thus, indicate nothing about how to compute a solution. Such computing is often very hard [Papadimitriou, 2001]. Secondly, game theory is built on the notion of rationality, but some debates are concerned with the question of its validity and/or utility for artificial agent societies. Thirdly, Boutilier et al. [1994] propose another difference, that is also related with rationality. This difference is about the assumption in economics/game theory that agents are rational (the research questions concern the social consequences of this hypothesis), while programming this rationality is the problem itself in multi-agent systems.

Social Science

- *Similarity*: Social sciences study the dynamics of human societies, while multi-agent systems are concerned with artificial societies.
- *Difference*: It is not certain that the best way of building artificial societies is to base them on human societies. Moreover, other tools, such as game theory, also models human societies, and may thus be applied.

Because of similarities between multi-agent systems and other fields (distributed/concurrent systems, artificial intelligence, economics/game theory and social science), agents have been applied in some of these fields. Furthermore, they have also been applied in many real-world applications, that are, in general, functionnally or geographically distributed. We now present some of these applications of multi-agent systems.

2.2.6 Some Applications of Multi-Agent Systems

Multi-agent systems have been used in many fields, as presented by [Chaib-draa, 1995], Jennings and Wooldridge [1998], Weiss [1999], and Wooldridge [2001]. As an illustration, we now outline some of these applications. Jennings et al. [1998] classify these applications in four classes:

Industrial applications: Industry was one of the earliest users of agent technology, especially in the following areas:

- *Manufacturing:* For example, the *Holonic Manufacturing Systems* (HMS) project [Gruver et al., 2003; Parunak, 1999; Shen and Norrie, 1999a] aims at standardizing architecture and technologies for open, distributed, intelligent, autonomous and cooperating systems in industry. Each component of these systems is controlled by agents, called "holons" for the combination of "holos" (the whole) and "on" (a particle) [Shen and Norrie, 1999a]. Each holon's goal is to work with the other holons, in order to control a production system in an efficient, scalable, open way. Applications of holons are, for instance, concurrent engineering, collaborative engineering design, and manufacturing enterprise integration [Shen et al., 2001].
- *Process control:* Process control is at a lower level than manufacturing, because manufacturing aims at controlling several workstations, while process control focuses on a single workstation. In fact, the complexity of a workstation may require the decomposition of its control into agents.
- *Telecommunications:* Telecommunication networks are geographically spread over a large area. Using agents to manage such networks is thus a natural metaphor. For instance, British Telecommunications plc [2004] has developed the ZEUS Agent Building Toolkit for this purpose.
- *Air-traffic control* [Jennings and Wooldridge, 1998; Ljungberg and Lucas, 1992]: For example, OASIS is an air-traffic control system used at Sydney airport in Australia. Aircraft and the various air-traffic control systems are seen as agents. Agents are created when they approach Sydney airport. Their behaviour is both goal-directed ("I want to land"), and reactive to take real-time constraints into account. Some similar air-traffic control systems were designed for NASA [Callantine, 2003], or by Cammarata et al. [1983].
- *Transportation systems:* Like telecommunication networks, the geographical distribution of transportation makes that agents are a natural metaphor. For example, *Automated Highway Systems* [Hallé et al., 2003] unites several projects aiming at fully automatizing vehicle driving. Several goals are addressed, such as driving a vehicle without human intervention and collaborative driving. This second example consists of forming platoons of vehicles on roads, in order to improve the fluidity of traffic. Each vehicle is seen as an agent that tries to form a team with other vehicle-agents sharing the same part of trip.

Commercial applications: While agents for industry are quite often designed for a single, specific application depending on the company, commercial agents tend to be designed for a widespread diffusion. Among the areas of commercial agents, we can find:

- *Information management:* Since the users of Internet are more and more overloaded by information, agents can help them by filtering and gathering accurate information.
- *Electronic commerce:* Since Internet takes up a growing place in our everyday life, e-commerce promises to be more frequently used in the near future. In fact, agents can:
  - look in our place for the products that best fit our needs;
  - bid for products on auctions sites, such as eBay, Inc. [2004], following a given strategy [Ben-Ameur et al., 2002b; Rosenschein and Zlotkin, 1994];
  - try to form a coalition with agents buying a similar product, in order to have a price reduction due to the higher bought quantity [Asselin, 2002; Asselin and Chaib-draa, 2002; Breban and Vassileva, 2002].

  In particular, the Trading Agent Competition [2004] aims at confronting agents to find the best buying strategy in situations close to real-life [Cheng et al., 2003; Vetsikas and Selman, 2003].
- *Business process management:* Information systems are spread among the different departments in a company, in order to bring information together. Using agents can make this information collection easier and more efficient. The collected information is useful for company managers when they make business decisions.

Entertainment applications: Although this industry is not seen as serious in computer science, it is currently growing. Specific areas of entertainment agents are:

- *Games:* For example, the concept of agents was applied in the Game "Creatures" by Grand and Cliff [1998] to build artificial pets living together in a simulated environment. These animals are built to resemble real-life animals, and in particular, their "brain" is also a neural network.
- *Interactive theatre and cinema:* In these systems, users 'enter' in the movie to play a role in this movie, and to interact with other characters played by artificial agents. Programming these agents so that they resemble real people is an issue, because they have to look like human beings, to behave like them…

Medical Applications: Agents are more and more used in medical applications, for instance:

- *Patient monitoring:* For instance, Hayes-Roth et al. [1989]'s system Guardian is distributed to respect the fact that a team in a Surgical Intensive Care Unit is made up of people who have different expertise and who collaborate. Guardian has a hierarchical structure, in which a control agent controls perception/action agents and reasoning agents, in order to help manage patient care.
- *Rescue team management:* RoboCupRescue [Hiroaki, 2000; Paquet, 2003] is a competition involving the simulation of an earthquake similar to Kobe (Japan) in 1995. Agents model teams of firemen, policemen and ambulances, that have to be coordinated in order to minimize both the number of dead civilians and the number of destroyed buildings. The idea is that an earthquake scenario cannot be studied in real-life, and thus, has to be simulated in order to find which behaviour rescue teams should have.

Some other applications cannot be put in these four classes. For instance, *interface agents* assist users in software, like the paper clip in Microsoft Corp. [2004a]'s Office, even though this is currently a mono-agent system. *Simulations of ecological and social* systems are another kind of application of multi-agent systems. For example, Franchesquin and Espinasse [2000] programmed a multi-agent simulation, that takes into account both ecological and social dynamics, in order to study the hydraulic management of the Camargue (south of France).

Since we focus in the next chapter on agents in industry, and more precisely, in supply chain management, we will describe additional multi-agent systems in industry. It is worth noting here, that the HMS project, presented above, looks similar to multi-agent systems for supply chain management that we describe in Chapter 3, but they are indeed different, because the HMS project addresses problems at a lower level, that is, intra-company, while supply chains are made up of several companies.

2.3 Multi-Agent Systems in Supply Chain Management

The first section of this chapter introduced supply chain management, and the second one multi-agent systems. We now focus on the merging of these two fields. This thesis uses the multi-agent system paradigm to model a supply chain in order to apply multi-agent system tools to simulate supply chain management. In return, some coordination mechanisms can be proposed to multi-agent systems. Therefore, we first show how computers are currently used in supply chains, then we give some arguments justifying the use of multi-agent systems in supply chains, and finally some examples illustrate this section.

2.3.1 Information Technologies in Supply Chain Management

According to Simchi-Levi et al. [2000], "information technologies is an important enabler of effective supply chain management. Much of the current interest in supply chain management is motivated by the possibilities that are introduced by the abundance of data and the savings inherent in sophisticated analysis of these data". We see that information technologies pursues three goals represented in Figure 2.5:

- collecting information on each product from production to delivery or purchase point, and providing complete visibility for all parties involved;
- accessing any data in the system from a single-point-of-contact, e.g., from a PDA linked to the company information system through a wireless link;
- analyzing data, planning activities, and making trade-offs based on information from the entire supply chain.

To achieve these activities, information technologies use certain means:

- information technology infrastructure (network, databases…);
- e-commerce;
- supply chain components, which are the various systems directly involved in supply chain planning, i.e., Decision Support Systems (DSS).

The standards gathering these three means are, for example, the protocol for Electronic Data exchange (EDI). Although regarded as a success because it is used by large corporations, EDI was never accepted by the majority of the communities of the business world as a means of trading electronically, because of its cost that is a barrier for little companies [Cingil and Dogac, 2001]. This explains why new Internet-based standards currently emerge. In particular, the eXtended Markup Language

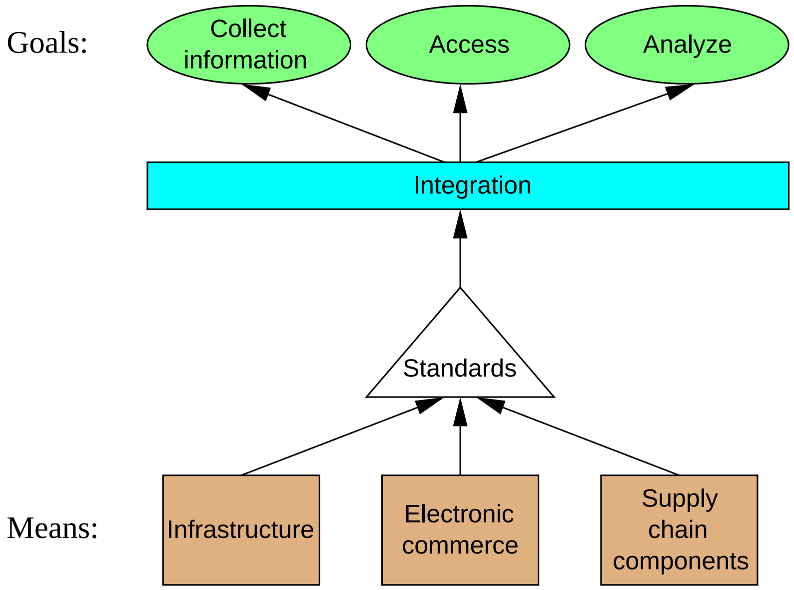


Figure 2.5: Goals and means of Information Technologies for Supply Chain Management [Simchi-Levi et al., 2000].

(XML) [World Wide Web Consortium, 2003b] is used in more and more applications on the Internet. But XML is too generic to enable collaboration in supply chains. Therefore, some XML-based standards were proposed, such as:

- Resource Description Framework (RDF) [World Wide Web Consortium, 2003a] to define a common vocabulary for describing resources;
- Common Business Library (CBL) [OASIS, 2003] for describing documents such as orders or catalogues;
- DARPA [2003] Agent Markup Language (DAML) to give semantics to Web pages.

Concretely, information and decision technologies take the form of:

- Enterprise Resource Planning (ERP) is a class of software systems organizing and managing companies [Garé, 2000], e.g., PeopleSoft, Inc and J. D. Edwards & Co. [2003], or Baan AG [2003];
- E-commerce, and in particular marketplaces, such as Commerce One Operations, Inc. [2003] and Ariba, Inc. [2003];
- Advanced Planning and Scheduling (APS) is a class of software for Decision Support System (DSS) in supply chains.

According to Shapiro [2000]'s decomposition of information technologies, the first two applications (ERP and e-commerce) belong to "Transactional Information Technologies", because they are concerned with acquiring, processing and communicating raw data. On the other hand, APS belongs to "Analytical Information Technologies" because they allow analyzing raw data in order to help managers, which is a task at a higher level. In practice, companies first install transactional tools, because analytical tools need them to be fed with raw data. More and more, multi-agent systems are seen as a new technology for improving, or replacing, technologies used in transactional and analytical information technologies. We now explain why agent technology seems so promising in the context of supply chains.

2.3.2 Using Multi-Agent Systems in Supply Chain Management: Motivations

Some arguments in favor of using multi-agent systems in supply chain management can be found in the literature. In fact, researchers have already applied agent technology in industry to concurrent engineering, collaborative engineering design, manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control, material handling, and holonic manufacturing systems [Shen et al., 2001].

Concerning supply chains, Dodd and Kumara [2001] think that Mark Fox et al. [1993] was propably the first to organize the supply chain as a network of intelligent agents. Indeed, supply chains are made up of heterogeneous production subsystems gathered in vast dynamic and virtual coalitions. Intelligent distributed systems, e.g., multi-agent systems, enable increased autonomy of each member in the supply chain. Each partner (or production subsystem) pursues individual goals, while satisfying both local and external constraints [Maturana et al., 1999]. Therefore, one or several agents can be used to represent each partner in the supply chain (plant, workshop, etc.). Moreover, the agent paradigm is a natural metaphor for network organizations, since companies prefer maximizing their own profit than the profit of the supply chain [Viswanathan and Piplani, 2001]. In fact, the distributed manufacturing units have the same characteristics as agents [Cloutier et al., 2001] (based on Wooldridge and Jennings [1995]'s definition of agents, quoted previously):

- *autonomy*: a company carries out tasks by itself without external intervention and has some kind of control over its action and internal state;
- *social ability*: a company in the supply chain interacts with other companies, e.g., by placing orders for products or services;

- *reactivity*: a company perceives its environment, i.e., the market and the other companies, and responds in a timely fashion to changes that occur in it. In particular, each firm modifies its behaviour to adapt to market and competition evolutions;
- *pro-activeness*: a company not only simply acts in response to its environment, it can also initiate new activities, e.g., launching new products on the market;

Moreover, multi-agent systems offer a way to elaborate production systems that are decentralized rather than centralized, emergent rather than planned, and concurrent rather than sequential. Therefore, they allow relaxing the constraints of centralized, planned, sequential control [Parunak, 1996]. Unfortunately, an agent-based approach is not the panacea for industrial softwares. Like other technologies, this approach has advantages and disadvantages: it must be used for problems whose characteristics require its capacities. According to Parunak [1999], five characteristics are particularly salient. In fact, agents are best suited for applications that are:

- modular;
- decentralized;
- changeable;
- ill-structured;
- complex.

To judge relevance for supply chains of autonomous agents, Parunak [1996] compares this approch with conventional technologies in Table 2.1, thus highlighting differences between these two philosophies. To this end, multi-agent systems are identified as biological (ecosystems) and economical (markets) models, whereas traditional approaches are compared with military patterns of hierarchical organization. Table 2.1 summarizes the main disadvantages of multi-agent systems:

1. theoretical optima cannot be guaranteed, because there is no global view of the system;
2. predictions for autonomous agents can usually be made only at the aggregate level;
3. in principle, systems of autonomous agents can become computationally unstable, since, according to System Dynamics, any system is potentially unstable.

	Issue	**Autonomous agents**	**Conventional systems**
	Model	Economics, biology	Military
	Issues favouring conventional system		
1	Theoretical optima?	No	Yes
2	Level of prediction	Aggregate	Individual
3	Computational stability	Low	High
	Issues favouring autonomous agents		
4	Match to reality	High	Low
5	Requires central data?	No	Yes
6	Response to change	Robust	Fragile
7	System reconfigurability	Easy	Hard
8	Nature of software	Short, simple	Lengthy, complex
9	Time required to schedule	Real time	Slow

Table 2.1: Agent-based vs. conventional technologies [Parunak, 1996].

But on the other hand, the autonomous, agent-based approach has some advantages too:

4. because each agent is close to the point of contact with the real world, the systems's computational state tracks the state of the world very closely...
5. ... and without need for a centralized database;
6. because overall system behaviour emerges from local decisions, the system readjusts itself automatically to environmental noise ...
7. ... or to the removal or addition of agents;
8. the software for each agent is much shorter and simpler than would be required for a centralized approach, and as a result is easier to write, debug and maintain.
9. because the system schedules itself as it runs, there is no separate scheduling phase of operation, and thus no need to wait for the scheduler to complete. Moreover, the optima computed by conventional systems may not be realizable in practice, and the more detailed predictions permitted by conventional approaches are often invalidated by the real world.

All these reasons show the relevance to use agents in supply chain management. In other words, thanks to their adaptability, their autonomy and their social ability, agent-based systems are a viable technology for the implementation of communication and decision-making in real-time. Each agent would represent a part of the decision-making process, hence creating a tight network of decision makers, who react in real-time to customer requirements, in opposition to the flood of current processes, which is decided before customers place an order [Dodd and Kumara, 2001].

2.3.3 Using Multi-Agent Systems in Supply Chains: Examples

We now illustrate the use of agents in supply chains by presenting various projects. These projects can be separated into two broad families: supply chain management projects [Chen et al., 1999] and supply chain design projects. Moreover, the manner of solving problems also differs depending on projects, e.g., the number and the role of agents vary considerably, depending on the particular point under study. To highlight these differences, Table 2.2 summarizes the projects that are now described.

1. DragonChain was implemented by Kimbrough et al. [2002] at the University of Pennsylvania (Philadelphia, PA, USA) to simulate supply chain management, and more particularly to reduce the bullwhip effect. For that, they base their simulation on two versions of the *Beer Game*, the *MIT Beer Game* (i.e., the original game) and the *Columbia Beer Game*, and they use agents that look for the best ordering scheme with genetic algorithms. We have further described this work as a related work in Section 1.5, because it is the closest approach to ours.

2. Agent Building Shell at the University of Toronto (Ontario, Canada) is a library of software classes providing reusable elements for building agent systems. These agents have four layers: a layer for knowledge management, an ontology layer, a layer of cooperation and conflict solving, and a layer of communcation and coordination. This latter layer is insured by the COOrdination Language (COOL). This project has involved several researchers, such as Teigen and Barbuceanu [1996], Barbuceanu and Fox [1995a,b], Beck and Fox [1994].

3. MetaMorph II is an improvement of a first project called MetaMorph. Agents form a federation centered around mediators that have two roles: they allow agents to find each other, and they coordinate these agents. These two projects were developed at the University of Calgary (Alberta, Canada) by Maturana et al. [1999] and others.

4. NetMan (NETworked MANufacturing) formalizes networked organizations and production operations in order to obtain agile manufacturing networks in a dynamic environment. Conversely to DragonChain, this multi-agent system manage

Project	Studied problem	Approach	Number and role of agents
1. DragonChain Kimbrough et al. [2002]	Management (Bullwhip effect)	Genetic algorithm seeking the best ordering scheme	1 agent/company
2. Agent Building Shell Teigen and Barbuceanu [1996] Barbuceanu and Fox [1995a,b] Beck and Fox [1994]	Management (Coordination)	COOrdination Language (COOL)	1 agent/company
3. MetaMorph 1 & 2 Maturana et al. [1999]	Management (Coordination)	Mediator-agents	1 agent/company +mediator-agents
4. NetMan Cloutier et al. [2001] Lyonnais and Montreuil [2001]	Management (Intra- and inter-company operations management)	Contract driven coordination in Convention, Agreement and Transaction (CAT) formalism	1 agent/workshop
5. BPMAT &SCL Swaminathan et al. [1998, 1994] IBM Research [2003]	Modelisation (Which elements are common to all supply chains?)	Comparison of three very different supply chains	BPMAT models companies & SCL intercompany streams
6. MASCOT Sadeh et al. [1999]	Management (Agility increase)	Comparison of several coordination policies	1 agent/company
7. DASCh Parunak and VanderBok [1998] Baumgaertel et al. [2001]	Management (supply chain modelisation techniques)	Delays and uncertainties on streams modelled as agents	2 agents/company + 1 agent/stream
8.Task dependency network Walsh [2001] Walsh and Wellman [1999, 1998]	Design & management (Partner selection)	Comparison of auction protocols	1 agent/company
9. MASC Labarthe [2000]	Design (Partner selection)	Auction-based protocol under constraints	1 agent/company + 2 directory agents
10. OCEAN Bournez and Gutknecht [2001]	Management (Global cooperation emerging from local competitions)	Negotiation system in a multi-agent contract network	1 agent/company (1 agent = system of 6 agents)

Table 2.2: Some projects applying agents to supply chains.

an actual supply chain, rather than the Beer Game. Each company is cut in NetMan centers, i.e. units of independent, collaborating business units. The NetMan centers of a company coordinate with each other and with other customers' and suppliers' NetMan centers. This coordination is based on contracts and conventions, which are formalized according to the model *Convention, Agreement, Transaction* (CAT). This work was carried out at Université Laval (Québec City, Québec, Canada) [Cloutier et al., 2001; Lyonnais and Montreuil, 2001].

5. BPMAT is a software library developed by IBM Research [2003] to model company activities. SCL is the addition to this library for modelling inter-company flows. These two tools are based on Swaminathan et al. [1998, 1994]'s work at Carnegie Mellon University (Pittsburgh, PA, USA), which sought elements common to any

supply chain by comparing three chains from distinct industrial sectors.

6. MASCOT (Multi-Agent Supply Chain cOordination Tool) is a reconfigurable, multilevel, agent-based architecture for planning and scheduling aimed at improving supply chain agility. It coordinates production among multiple (internal or external) facilities, and evaluates new product/subcomponent designs and strategic business decisions (e.g., make-or-buy or supplier selection decisions) with regard to capacity and material requirements across the supply chain [Sadeh et al., 1999]. Like BPMAT and SCL, this work was also accomplished at Carnegie Mellon University (Pittsburgh, PA, USA).

7. DASCh was developed by ERIM (Ann Arbor, MI, USA) by Parunak and VanderBok [1998] and Baumgaertel et al. [2001] to explore the modeling techniques of networks of suppliers and suppliers' suppliers. In particular, flows of products and information flows are viewed as agents to model imperfections in these flows.

8. The Task dependency network is an asynchronous, decentralized market protocol (auctions) for allocating and scheduling tasks among agents that contend for scarce resources, constrained by a hierarchical task dependency network [Walsh, 2001; Walsh and Wellman, 1998]. An additional paper [Walsh and Wellman, 1999] extends this protocol to model supply chain formation. This work is a Ph.D. thesis defended in 2001 at the University of Michigan (Ann Arbor, MI, USA). In similar ways, other works use market mechanisms to coordinate supply chains [Eymann, 2001; Fan et al., 2003].

9. MASC studies coordination modes between companies in supply chains. These coordination modes are calls for submissions, which submitters answer according to their capacity and production load. Companies winning this auction next take part to the supply chain carrying products to the consummer. This work was completed at the University of Aix-Marseilles 3 (Marseilles, France) [Labarthe, 2000].

10. OCEAN (Organization and Control Emergence with an Agent Network) is a control system with an open, decentralized and constraints-based architecture in which there is responsiveness, and distribution of production resources and technical data. This system was designed to react to environment dynamics, in order to show that cooperation at the global level may emerge from competitions at the local level. This work was completed at INSA of Lyon (Lyon, France) and at the University of Montpellier 2 (Montpellier, France) [Bournez and Gutknecht, 2001].

Note that the work presented in this dissertation is at a higher level than what has just been presented. In fact, we do not focus on simulating supply chains in detail, but we rather focus on simulating the decision making in supply chains.

2.4 Conclusion

This chapter has presented the two research fields addressed in this dissertation. First, supply chain management provides an illustration of stream fluctuations, which is the problem studied in this thesis. A supply chain is the set of companies producing and distributing products to the end-customer. Some industrial issues were first introduced, then the concepts of supply chains and collaboration were proposed as solutions to these issues. After that, we have presented the model of the Economic Order Quantity (EOQ), because this is a modelisation ruling order placement, while the bullwhip effect affects orders.

The other field addressed in this dissertation, i.e., multi-agent systems, was presented in the second section of this chapter. We first defined agents, next we presented their differences with another concept from computer science, i.e., the concept of object. Then, agent architectures were outlined, arguments pro agents were defended, and differences between multi-agent systems and some other fields were underlined.

Finally, we have reviewed the literature about the intersection of the two fields addressed in this thesis. After a recall of information technologies in supply chains, we advocated the use of multi-agent systems for a new generation of information technologies. Such use of agents was illustrated by projects involving multi-agents systems in supply chain design or management.

The next chapter continues this literature review by a description of the problem of stream fluctuations in distributed systems.

Chapter 3

The Bullwhip Effect

The previous chapter presented the background of this research, that is, the fields of Supply Chain Management and Multi-Agent Systems, and their analysis. This review is continued in this chapter by introducing the problem studied in this disseration, that is, the bullwhip effect.

For that purpose, one of the contributions of this dissertation is to introduce the bullwhip effect by taking a wide point of view, by outlining the problem of stream fluctuations in distributed systems. We illustrate through examples how this problem occurs in many fields, and in particular multi-agent systems. This illustration is given in Section 3.1. Next, we focus on this problem when the considered distributed system is a supply chain. We detail the known consequences and causes of this effect in Section 3.2. Finally, because of the similarities of this effect and stream fluctuations in other fields, we show how this effect has been studied with tools from other fields. This point is another contribution of this dissertation, and is presented in Section 3.2.

3.1 Stream Fluctuations in Distributed Systems

As stated in Chapter 1, distributed systems are travelled by several flows during their operation. When these flows are linked together, the disturbance of a flow induces the disturbance of other streams, which in turn, has other impacts on the system. Eventually, stream fluctuations may increase or decrease with time because of numerous factors that are more or less easy to understand. In general, such stream fluctuations decrease the efficiency of the distributed system, when it achieves the task for which

Distributed system	**Stream**	**Nodes**	**Instability**
1. Supply chain	Orders and products	Companies and end-customers	Bullwhip effect
2. Computer network	Information	Computers	Congestion
3. Road network	Vehicles on roads	Cities	Traffic jam
4. Electronic circuit	Intensity	Electronic components	Roasting
5. Human organizations and multi-agent systems	Tasks	Human, physical and agent resources	Resource bottleneck...
6. Economic system	Money, tasks and information	People, companies, states...	Crises, inflation...

Table 3.1: Distributed systems affected by stream fluctuations.

it was designed. This shows that dynamics in distributed systems are, in general, very complex. In this section, we illustrate the problem of stream fluctuations in distributed systems by giving some examples drawn from very different fields. These examples are summarized in Table 3.1, and described below:

1. The bullwhip effect, which is studied in this dissertation and described in Section 3.2, is an instance of stream fluctuations, when the considered distributed system is a *supply chain*. Indeed, what we call a supply chain is almost always a network of companies, that is, there may be cycles in its structure, but most supply chains in the literature have none, like in Figure 2.1. Therefore, such "supply network" may be affected by fluctuations on its ordering and product stream, i.e., by the bullwhip effect, as presented in this chapter. As was previously pointed out, streams in a distributed system may be linked together. In the case of supply chains, this link works as follows: the product stream is pulled by the ordering stream, and conversely, disturbances on product stream, e.g., delays, also impact on the ordering stream.

Besides supply networks, graphs (i.e., the generalization of networks, because networks are particular graphs having a cycle) also provide other examples of distributed systems, in which stream fluctuations may appear, e.g., computer and road networks:

2. Nodes in *computer networks*, such as the Internet, are computers, and streams are information travelling between these computers. Here, fluctuations of information

stream may generate congestions, because such fluctuations may saturate some computers in the network, in which case no information can transit by these computers.

3. Cities are nodes in *highway networks*, or buildings in street networks, and edges represent roads. In this context, congestions may appear due to the high density of vehicles. Such fluctuations alternate between full and empty roads, instead of constantly maintaining an average traffic density. Empty roads are not a problem, but full roads result in traffic jams.

4. *Electronic circuits* are another type of network, where electricity circulates through electronic components. In this case, electricity fluctuations alternate between low and high intensity in the circuit. The problem occurs when the intensity becomes too great, creating roasting.

   From a more general point of view, electronic circuits are also related to grids. The term grid is also used in other fields, such as electricity transportation (power grid) or distributed computing. In computer science, grids refer to information grids that allow sharing computing power. More precisely, when a machine does not use its whole available computing power, this machine could compute something for another user. For example, some projects, such as United Devices Grid [2004] or Seti@Home [Search for Extra-Terrestrial Intelligence, 2004], propose short programs downloading data from a central server on your computer. This data is used by this short program to compute a piece of a much larger calculation. The idea is to build a virtual supercomputer with the aggregation of many computers using this short program. But information grids are not distributed systems for the moment, because the server providing information acts as a center in this system. Nevertheless, they could become so in the future, when every computer will delegate some of its tasks to underloaded computers, which will make the center disappear. In this case, stream fluctuations may also appear in such decentralized computation grids.

The two last examples in Table 3.1, human organizations and economic systems, are not always seen as graphs or networks. In fact, interactions between the nodes in these two types of distributed systems are often ill-structured, and thus, cannot be represented by a graph.

5. *Human organizations*, such as companies, and *multi-agent systems* may be organized as a hierarchy, i.e., as a particular type of graph called a tree, in which, nodes are resources (people, machines...), and edges are streams of tasks. In fact, streams of tasks appear between nodes, e.g., Alice asks Bob to program an

agent on the Computer to do a sequence of actions, and this agent asks Alice to do another action as a consequence of this sequence of actions. If fluctuations appear on such task streams, a person may be *overloaded*, while another person has no task to accomplish.

When there are a few nodes, graphs can be used to model this organization, in order to study such stream fluctuations. On the contrary, it is much harder to use graphs when there are a huge number of nodes.

6. *Economic systems* may also be represented by a graph in which nodes are people, companies, banks, manufacturers, states..., and edges are streams of money, tasks, information... But in general, because of the complexity of this system, such regularity cannot be easily modelled as a graph in economic systems. Such systems are distributed however, because their global behaviour is the consequence of the individual behaviour of their nodes.

   Evidently, such systems may also suffer from stream fluctuations. For example, the world economy was known to work with 25 year growths, followed by 25 year crisis. Here, economic growth is an acceleration of (money, products, tasks...) streams in the system, and crisis a deceleration. Therefore, there are stream fluctuations, that are roughly cyclic with a period of 50 years. Nowadays, we can note that such cycles are shorter and shorter, and crisis are less and less important. This shows that some nodes in this system have learnt how to act on (a part of) the rest of the system. For example, states can change taxes, grants and interest rates, and launch major projects to stop a crisis. This shows that stream fluctuations can be controlled even in very complex systems.

As we can see, these examples are drawn from very different fields. Although, all of them can be modelled as distributed systems travelled by streams that may fluctuate. In every example, such stream fluctuations *reduce the efficiency* of the considered system.

3.2 The Particular Case of the Bullwhip Effect

We now describe further the first example in Table 3.1: the bullwhip effect. This name was given by Lee et al. [1997a,b] to the amplification of order variability in a supply chain. To this end, Figure 1.1 (recalled in Figure 3.1) shows how the bullwhip effect propagates in a simple supply chain with only three companies: a Retailer, a Wholesaler and a PaperMill. The ordering patterns of the three companies are similar, in the way that the order variability by an upstream site is always greater than by a downstream

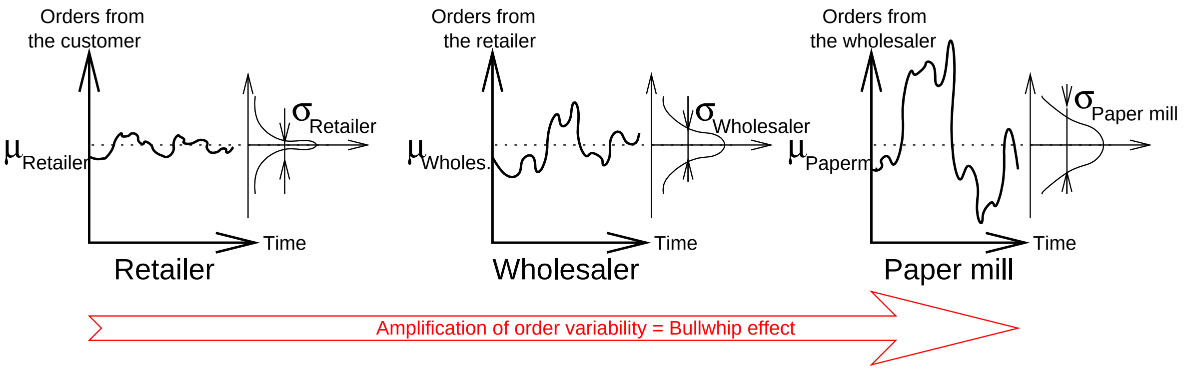


Figure 3.1: The bullwhip effect [Lee et al., 1997a,b] (recall of Figure 1.1).

site [Lee et al., 1997a]. As the bullwhip effect is a varibility, it is measured as the standard deviation σ of orders. Note that the means μ are all equal in Figure 3.1, because every company has only one client and one supplier.

3.2.1 Consequences of the Bullwhip Effect

As we have said in Section 3.1, stream fluctuations are undesirable in any distributed systems, because they reduce the efficiency of the considered system. In the case of supply chains, the efficiency reduction due to the bullwhip effect incurs costs[1] due to the consequences of this effect. In particular, for the North European forest industry, Carlsson and Fullér [2001] estimate that this cost is 200-300 MFIM (million Finland Markkaas), i.e., 50-70 million CAD, for a 300 kton paper mill. These costs may occur due to the following consequences:

- *Higher inventory levels*: Every participant in the supply chain has to stockpile because of a high degree of *demand uncertainties and variabilities* induced by the bullwhip effect [Lee et al., 1997b];
- *Supply chain agility reduction*[2]: As inventory levels are high (cf. consequence "higher inventory levels"), the supply chain should sell products in inventory, before it sells the new products demanded by end-customers, which generates inertia in following end-customer demand. Moreover, demand uncertainties induced by

---

[1] Costs are an indirect metrics of the bullwhip effect; the unique direct metrics is the standard deviation of placed orders.

[2] The Iaccoca Institute [1991] (cited by Rigby et al. [2000]) defines agility as the "ability of an organisation to thrive in a constantly changing, unpredictable business environment".

the bullwhip effect make more difficult for the supply chain to understand which product is demanded by end-customers;

- *Decrease of customer service levels*: Demand variabilities may incur stockouts, in which case, no products are available to be sold, and thus, no service can be given to customers. In this dissertation, we only consider this cause of decrease of customer service levels, but there is another one. In fact, demand uncertainties also make that the supply chain has more difficulties to understand what products end-customers wish, which is the consequence "supply chain agility reduction" presented above;

The two last consequences of the bullwhip effect are related to the difficulties of planning under uncertainties, and consequently lead to:

- *Ineffective transportation*: Transportation planification is made more difficult by demand uncertainties induced by the bullwhip effect;
- *Missed production schedules*: Similarly to transportation, production planification is made more difficult by demand uncertainties induced by the bullwhip effect;

3.2.2 Known Causes and Some Solutions to the Bullwhip Effect

After the consequences of the bullwhip effect, we now present its causes, as reflected in the literature. Table 3.2 summarizes the proposed causes and solutions of the bullwhip effect. Lee et al. [1997a,b] proposed the four first causes: (1) demand forecast updating, (2) order batching, (3) price fluctuation and (4) rationing and shortage gaming. We detail these four proposed causes:

Demand forecast updating: Every company places orders based on a forecast of the future demand, and the history of incoming orders is used to forecast future orders. The problem is that only retailers know the actual market consumption, and deform this information when they transmit it as orders to their suppliers. In fact, as these suppliers also want to fulfill end-customer demand, they should also base their orders on the market consumption. But these suppliers cannot do so, because they only have their incoming orders to estimate end-customer demand. As a consequence, retailers make a quite accurate forecast, because they are in contact with the market, while their suppliers make worse forecasts, because they have only their incoming orders to do their forecasts. For example, a supplier

Causes	**Proposed Solutions**	**Authors**
Demand forecast updating	Information sharing (e.g. VMI, CRP...), echelon-based inventory and leadtime reduction	Lee et al. [1997a,b]
Order batching	EDI and Internet technologies	Lee et al. [1997a,b] Taylor [1999]
Price fluctuation	EDLP (Every Day Low Pricing)	Lee et al. [1997a,b]
Rationing and shortage gaming	Allocation based on past sales	Lee et al. [1997a,b]
Misperception of feedbacks	Giving a better understanding of the supply chain dynamics to managers	Daganzo [2003] Forrester [1958, 1961] Sterman [1989] Dejonckheere et al. [2002]
Local optimization without global vision	None	Kahn [1987] Naish [1994] Shen [2001] Chen et al. [2000] Simchi-Levi et al. [2000]
Variabilities due to company processes	None	Taylor [1999]

Table 3.2: The bullwhip effect: causes and solutions.

may think there is a trend in market consumption even if it is steady, because the ordering policy and exponential smoothing used by the retailer make this trend appear. Such forecasting issues can lead to a bullwhip effect, because companies order more or less than the quantity required to fulfill the actual demand.

The longer the ordering and shipping delays are, the worse this situation is, because companies have to forecast further into the future. As a consequence, ordering and shipping delays are an *aggravating factor* to this cause of the bullwhip effect. We will see in Chapter 5, that we propose to see ordering and shipping delays as a cause of the bullwhip effect, rather than only as an aggravating factor of another cause.

A solution proposed to demand forecast updating is information sharing [Lee et al., 1997a,b]: each client provides more complete information to its supplier in order to allow the supplier to improve its forecasting. As stated in Section 2.1.1, information sharing is part of industry practices such as VMI (Vendor-Managed Inventory), CRP (Continuous Replenishment Program) and CPFR (Collaborative Planning, Forecasting and Replenishment) presented in Section 2.1.

Order batching: (lot sizing in a more general way) Let us consider the following example. We assume truck capacity is 10 items, a company needs 8 items each week (steady demand) and orders are placed each Monday. Therefore, the company orders 10 items the first Monday in order to have a full truckload (i.e. a lower shipping price) and puts 2 items in inventory. Next, demand remains at 8 items: the company orders 10 items in the second, third and fourth weeks, but nothing in the fifth week because the company has the 8 needed items in inventory. If this scenario is repeated for a year, the supplier of this company may believe no products are consumed by the market every fifth week. If this supplier understands that market consumption is steady, it takes several weeks order information to calculate this market consumption by means of orders. This example illustrates how order batching can generate the bullwhip effect [Lee et al., 1997a,b].

The proposed *solution* to lot sizing is electronic transactions (e-commerce, EDI...) to reduce transaction costs, and thus make companies order more frequently for smaller quantities of products.

Price fluctuation: When a company proposes a promotion, its clients buy more products in order to fill their inventory. When the price is back to its usual level, the clients stop buying, and consume their inventory instead. As we can see, changing the price of products also induce the bullwhip effect, because companies buy more or less than their actual requirements.

The proposed *solution* is EDLP (Every Day Low Pricing) policy, where the price is kept steady at the promotion level [Lee et al., 1997a,b].

Rationing and shortage gaming: We focus here on the strategic behaviour of companies. For example, when product demand exceeds supply (e.g., a machine breakdown reduces the quantity of available products), some clients might order more than their actual needs, because they try to have a bigger proportion of available products by "gambling", in order to receive a quantity closer to their actual needs. This amplifies order variability, because companies exaggerate their real requirements during rationing, and then cancel orders when this rationing stops. As a consequence, judging the real market consumption is difficult. This behaviour occurs when the manufacturer allocates the amount in proportion to the ordered amount.

Instead of that, it is preferable as a *solution* to allocate the few available products in proportion to the history of past orders [Lee et al., 1997a,b].

Other authors have extended Lee and his colleagues' causes and solutions to the bullwhip effect:

Misperception of feedback: Sterman [1989] analyzed Beer Game players and proposed that the bullwhip effect can be caused by players, who do not understand the whole dynamics of the system. For example, players who do not correctly interpret their incoming orders may induce a bullwhip effect. In particular, they may smooth their orders when they should order more because the market consumption has changed.

A solution to this cause consists in giving a better understanding of the supply chain dynamics to the players [Sterman, 1989].

Local optimization without global vision: Local optimization means companies maximize their own profit without taking into account the effect of their decisions on the rest of the supply chain [Shen, 2001; Naish, 1994; Kahn, 1987], that is, externalities[3] are ignored. In particular, some companies use an ordering policy, such as (s, S), which is the operationalization of a local optimization. It has been formally proven that some of these policies induce the bullwhip effect [Chen et al., 2000; Simchi-Levi et al., 2000].

No *solutions* were proposed to this cause.

Variabilities due to company processes: Company processes can also induce a bullwhip effect, and they were proposed by Taylor [1999] with the LEAP Project. In this project, these authors have studied the causes of the bullwhip effect in the upstream automotive component supply chain in the U.K. These authors proposed two *causes*: variability in machine reliability and output, and variability in process capability and subsequent product quality. In these two causes, which are summarized as "Variabilities due to company processes" in Table 3.2, production problems at each workstation are amplified from one workstation to another. This cause recalls that intracompany problems and uncertainties may affect each company's behaviour, which in turn may make them change the way they place orders.

No *solutions* were proposed to avoid the bullwhip effect induced by such variablities due to company processes.

---

[3] Externalities are defined in page 87 as the lack of taking into account the effect of an agent on other agents.

3.3 Different points of view on the Bullwhip Effect

The presentation in the previous section of the bullwhip effect, and its proposed causes and solutions is a synthesis of many works belonging to several fields. In particular, Scholl [2001] noted that many of these works belong to system dynamics and agent-based modeling, which are two prominent nonlinear modeling schools. In order to take into account works in other fields, one of our contribution is to extend Scholl's classification of the literature about the bullwhip effect into three broad classes: (1) formal studies relative to the bullwhip effect, i.e., theoretical approaches, (2) empirical studies of the bullwhip effect, i.e., empirical approaches, and (3) simulation approaches of the bullwhip effect. We now detail these three classes.

3.3.1 Formal studies relative to the bullwhip effect

In the first class of approaches, mathematics tools from different fields have been used. We present tools from system dynamics, classic inventory management, economics, and traffic flow and control theories:

System Dynamics

This is the field in which the bullwhip effect was first described by Forrester [1958, 1961] in 1958[4]. This explains why this phenomenon is sometimes called Forrester's effect. Much later, Angerhofer and Angelides [1999] gave an overview of system dynamics and its application to supply chain management related issues. For their part, Min and Bjornsson [2000] described a construction of supply chains with system dynamics in order to develop an agent-based supply chain management model.

Classic Inventory Management

It is not surprising to see, that most of theoretical studies of the bullwhip effect belong to the field of inventory management. In this field, models aim at demonstrating when and why ordering policies induce the bullwhip effect. Precisely, these models focus on

---

[4] However, Scholl [2001] says Sterman points out that this phenomenon was described at least as early as the 1920s and 1930s in Economics and Management Science literature.

the stabilization of product and ordering streams in the supply chain. Conversely to the Economic Order Quantity model (EOQ) presented in Subsection 2.1.2, companies do not try to maximize their profit in this type of study of the bullwhip effect, they only apply an ordering policy to manage their inventory. Of course, cost or profit optimization is the basis of such ordering policies, as it is the basis of EOQ. This optimization is used to design an ordering policy, but this ordering policy is taken into account in such studies without reconsidering the underlying optimization [Chen et al., 2000, 1994; Ryan, 1997; Simchi-Levi et al., 2000; Yao, 2001].

Many years after Forrester, in the field of inventory management, Lee et al. [1997b] gave a more complete understanding of the amplification of demand variability; they termed it the *bullwhip effect*. In particular, they proposed the four causes of this phenomenon that we have just seen: (1) demand forecast updating, (2) order batching, (3) price fluctuation, and (4) rationing and shortage gaming; however, some other causes have been identified [Taylor, 1999; Sterman, 1989; Forrester, 1958].

In the same way, Chen et al. [2000] and Simchi-Levi et al. [2000] studied the impact of delays and two particular forecasting techniques, and how the bullwhip effect is reduced with information centralization, i.e., a particular form of information sharing in which retailers multi-cast the market consumption in the rest of the supply chain. In fact, such centralization of information allows each company in the supply chain to create more accurate forecasts, rather than relying on downstream company demand, which can vary significantly more than the actual market consumption. In our validation in Chapter 7, we also consider this centralization.

Work similar to Simchi-Levi and his colleague has been done by Dejonckheere et al. [2002] but with exponential smoothing algorithms, and Kelle and Milne [1999] also did similar work to Simchi-Levi and his colleagues, but with the (s, S) ordering policy.

Economics

The third formalism used to study the bullwhip effect is taken from the field of economics. Indeed, they have studied how local optimizations, done by companies without taking into account the rest of the supply chain, cause the bullwhip effect [Shen, 2001; Naish, 1994; Kahn, 1987]. The difference with the previous approach is the fact that optimization is explicitely taken into account in economics models, while it disappears in inventory management models of the bullwhip effect: the latter approach proposes its ordering policies on local optimization too, but the proposed policies are used without reconsidering optimization. The idea in economics models is to represent a single inventory system under an incoming demand, which is modelled as a statistic distribution. In this context, researchers show that optimization of costs increases the standard deviation of placed orders in comparison with incoming orders. In other words, the distribution of incoming demand at the input of such models has a lower standard deviation than the distribution of placed order at the output of the model. In particular, Cachon studies such questions under different scenarios [Cachon, 1999].

Traffic Flow Theory

Another interesting approach is Daganzo [2003]'s "Theory of Supply Chains" where the author focuses on ordering policies that stabilize flows in the supply chain. To achieve this, he used a specific formalism based on traffic flow theory. That is, he proposed a wide mathematical framework based on this theory to study many different kinds of ordering policies. In particular, he gave properties that ordering policies should have in order to avoid the bullwhip effect, and shows that all currently used policies lead to the bullwhip effect.

Control Theory

In a very similar approach to traffic flow theory, Dejonckheere et al. [2004, 2003, 2002] and Disney and Towill [2003] also focus on stabilizing streams in the supply chain, except that they use another formalism, called control theory. Precisely, studies of supply chain using either traffic flow theory or control theory are only different from the view point of the mathematical tools that are used.

3.3.2 Empirical studies of the bullwhip effect

We now present some empirical studies of the bullwhip effect. First, Lee et al. [1997a] gave a non-formal description of their paper [Lee et al., 1997b] that we have seen above. In a more practical way, Fransoo and Wouters [2000] proposed a method for the measurement of the bullwhip effect in real-life. Similarly, The LEAP (LEAn Processing) Project has also studied the bullwhip effect in practice. The supply chain studied in this project was made up of three echelons of the automotive component supply chain in the U.K. [Hines and Holweg, 2000; Taylor, 2001, 2000, 1999]. Conversely to these two practical studies, Wilding [1998] explained from a conceptual view point that uncertainty is

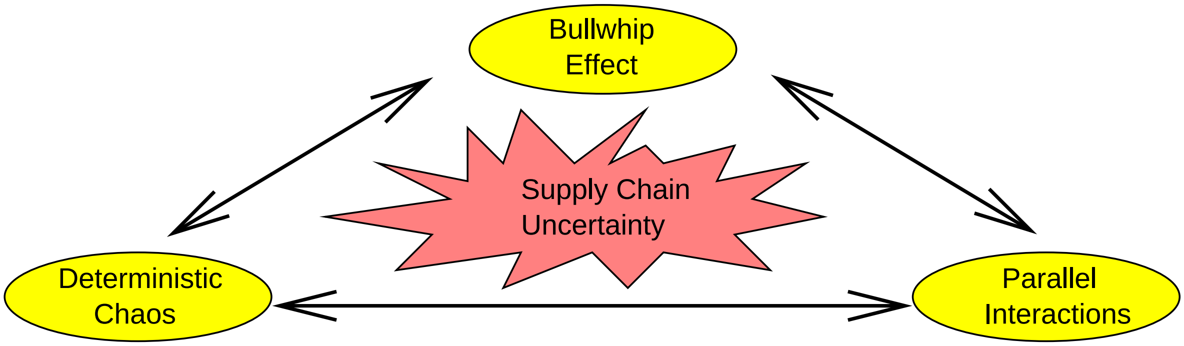


Figure 3.2: The supply chain complexity triangle [Wilding, 1998].

generated in the supply chain by three interacting sources, as illustrated in Figure 3.2:

1. The bullwhip effect;
2. Deterministic chaos (chaos appears while the system is deterministic, i.e. chaos appears in the system, while there is a definite rule with no random terms governing the dynamics of this system);
3. Parallel interactions (interactions between companies in the same echelon may appear: a retailer has an influence not only on its suppliers, but also on other retailers).

As previously stated, the bullwhip effect has also been studied using the Beer Game: the question was to find the managers' cognitive limitations that cause this demand variability [Croson and Donohue, 2002; Sterman, 1989]. In fact, although there are mathematical tools to manage inventories, some managers still use their intuition when placing orders. The problem lies in the fact that managers have trouble understanding the dynamics of a supply chain, because there are complex feedback loops, time delays and past orders to consider together.

In addition, the original Beer Game has been modified in order to take more realistic considerations into account. For instance, Chen and Samroengraja [2000] changed some parameters (market consumption...), and Haartveit and Fjeld [2002] and Fjeld [2001] adapted it to the North European forest industry to study how the structure of the game can result in a mismatch between supply and demand.

3.3.3 Simulation-Based Studies of the Bullwhip Effect

We now present simulation-based studies of the bullwhip effect, and in particular those that are based on the Beer Game. For example, Simchi-Levi et al. [2000]'s book is distributed with a Beer Game software in which the player plays with three other companies. These three companies can be set to apply one of six ordering policies, such as the (s, S) policy. In the same way, Jacobs [2000] proposed playing the Beer Game on the Internet.

As stated in Section 1.5, Kimbrough et al. [2002] have used software agents as players in the Beer Game to find the best ordering scheme with a genetic algorithm. In a less theoretic approach, Chatfield [2001] was also inspired by this game to design SISCO (Simulator for Integrated Supply Chain Operations). Similarly to the two latter approaches, we now focus on simulations of supply chains that involve agents.

In fact, we have shown in Subsection 2.3.2 that agent technology is widely recognized as a promising paradigm for the next generation of design and manufacturing systems. In this context, Yung and Yang [1999b] represented each company as an agent that minimizes its costs subject to some constraints. As agents work in parallel, the optimization of the supply chain is done concurrently. In a similar approach, Carlsson and Fullér [2001, 2000] used fuzzy logic to estimate demand for the upcoming period, and thus, reduce the bullwhip effect due to the cause called "demand forecast updating". Like Chen et al. [2000] but with the multi-agent paradigm instead of a formal model, Yan [2001] has studied the impact of delay distribution on the bullwhip effect to verify that increasing delays increases the bullwhip effect. Finally, Davidsson and Werstedt [2002] have implemented a multi-agent system to coordinate production and distribution in supply chains. Their architecture is general, and has been applied to the case study of district heating systems. They do not specifically study the bullwhip effect, but note that this phenomenon reduces the efficiency of their system. In spite of the fact that the bullwhip effect is not the point addressed in their work, they used information centralization, which is a means to reduce the bullwhip effect.

3.4 Conclusion

This chapter has presented the general problem of stream fluctuations in distributed systems. Some examples of such problems have illustrated this presentation, e.g., traffic jams are problems of vehicle stream fluctuations on roads, information congestions are problems of information stream management on computer networks, economic growth and crisis are fluctuations of money streams… One of the contributions of this dissertation is to point out this generalization of the bullwhip effect. Next, we have focussed on the case of supply chains, in which there is a particular case of stream fluctuations called the bullwhip effect. The known cause, consequences and solutions of this effect were presented. Finally, we have shown that a wide variety of approaches have been used to study this phenomenon. This variety is a witness to the similarity between the bullwhip effect and stream fluctuations in other fields. Another contribution of this dissertation is the wide literature review about the bullwhip effect.

Chapter 4

Formalizing Interactions with Game Theory

The previous chapter has presented the problem of stream fluctuations in distributed systems, and has focussed on the case of supply chains, in which these stream fluctuations are called the bullwhip effect. According to certain points of view, this effect is due to a lack of coordination in the supply chain [Cooper et al., 1997]. For this reason, in this chapter, we present how coordination is carried out in the two fields addressed in this thesis, i.e., supply chains and multi-agent systems. Since coordination deals with interactions between agents, this concept is introduced in this chapter also.

First, we outline some ways of coordinating streams in distributed systems. To this end, we see that most coordination mechanisms in supply chains focus on money (which promotion should we propose in order to make clients buy in coordinated ways with our processes? How to optimize jointly inventory systems in several companies, so that our activities are synchronized? Which contracts make companies work in a coordinated way, so that the whole supply chain earns the largest amount of money?), except one class of mechanisms which focusses on stream control in the supply chain (e.g., Responsibility Tokens, system PAC. . .). Since our proposition to reduce the bullwhip effect is concerned with such stream control, we develop our review of the literature around this class. First, we introduce coordination mechanisms in multi-agent systems. After that, we compare both types of coordination, i.e., coordination in supply chains and coordination in multi-agent systems, and to our knowledge, this is the first time when such comparison is made. Section 4.1 reflects all these aspects around the notions of coordination.

Since coordination takes place by means of interactions, we also introduce game theory as a formalism to study interactions and coordination. To this end, we present some notations and concepts from this theory, and we illustrate with examples how this theory is used to analyze interactions. Game theory is presented in Section 4.2.

Finally, the application of game theory to computer science requires algorithms, and as a consequence, we outline the current knowledge about the computational complexity of these algorithms in Section 4.3.

4.1 Coordination

As we focus in this dissertation on decentralized coordination mechanisms for reducing the bullwhip effect, we present some similar mechanisms in supply chain management first, and in multi-agent systems after. But first of all, Malone and Crowston [1994] note that coordination is of interest in many disciplines, such as computer science, organization theory, operations research and economics. They have proposed an interdisciplinary definition for coordination, which is the "management of dependencies between activities". From a conceptual viewpoint, research about coordination has two goals: the identification of these dependencies, and the proposition of processes to manage these dependencies.

In this context, Mintzberg [1978] has studied the structure of organizations and proposed a classification. Figure 4.1 shows the Frayret [2002]'s extension of this classification. In this figure, A and B are two broad classes that distinguish the forms of coordination by standardization from other forms of coordination, where class B is the extension of Mintzberg [1978]'s classification. One interesting feature of Figure 4.1 is the separation of direct and indirect forms of coordination, which allows distinguising third party coordination (a Superior or a Mediator helps A and B to coordinate) from mutual adjustment (A adjusts with B, while B also adjusts with A).

We now focus the presentation of coordination on two particular cases of organizations: supply chain, then multi-agent systems.

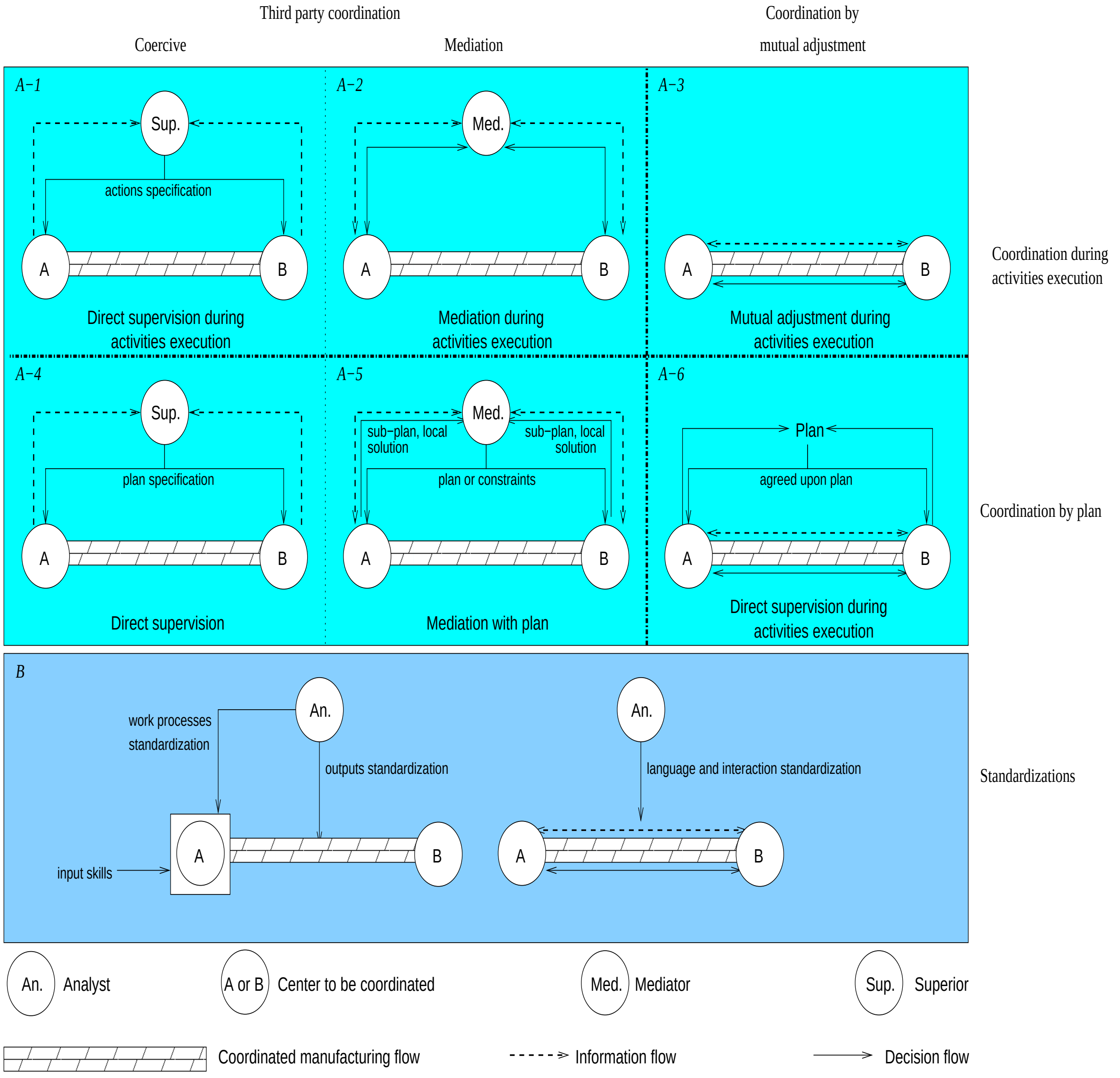


Figure 4.1: Generic classes of coordination mechanisms [Frayret, 2002].

4.1.1 Decentralized Coordination Mechanisms for Supply Chains

Overview of All Coordination Mechanisms for Supply Chains

According to Viswanathan and Piplani [2001], coordination in supply chains can be divided into three classes of mechanisms:

1. *Discounts:* In this approach, each client coordinates the way it places orders

with its supplier, in order to profit from quantity discounts made by this supplier. Precisely, the supplier proposes some discounts to the client, so that the client has incentives to order products in the way that best suits the supplier, for example, by ordering bigger batches of products [Rubin and Benton, 2003; Corbett and de Groote, 2000; Dolan, 1987; Crowther, 1964].

2. *Joint Optimization:* We have seen in Subsection 2.1.2, that the Economic Order Quantity (EOQ) aims at minimizing the cost of the inventory system in a single company. Extensions of this model to several companies are known as "Joint Economic Lot Size" (JELS) or "Multi-Echelon Inventory" models. Most of these models, but not all, only consider two levels in a supply chain, i.e., clients and their suppliers, but not a larger supply chain. The goal is to minimize the overall logistics cost of the considered supply chain by centralizing decisions. Yang and Pan [2004]; Kosadat and Liman [2002]; Lee and Whang [1999]; Cachon and Zipkin [1999]; Banerjee [1986]; Clark and Scarf [1960]

3. *Contracts:* Every company can commit itself to perform actions, and pay penalties if it does not do these actions. Such commitments are fixed in contracts. According to Cachon [2004], a contract coordinates a supply chain, when this contract has the set of companies' optimal actions as a Nash equilibrium. In other words, a contract coordinates the supply chain, when every company is worse off when it unilaterally deviates from the contract (because it has to pay penalties) because non-deviating is the best for the whole supply chain. Much work [Corbett et al., 2004] exists about contracts in the supply chain. A good review of this work has been proposed by Tsay et al. [1999].

As we can see, these three classes of mechanisms consider the coordination of the supply chain from the point of view of profit optimization. We now add a fourth class of coordination mechanism, that considers stream stabilization instead of money saving:

4. *Stream control:* The goal of this approach is to control flows of products through the production system. Money is no longer taken into account, because the metrics is now the fluidity of these flows [Buzacott and Shanthikumar, 1992; Liberopoulous and Dallery, 2000]. Of course, money is also taken into account secondarily, because a more fluent supply chain is more efficient, and thus, earns more money.

Since our proposition to reduce the bullwhip effect is a coordination mechanism controlling streams in a supply chain, we now extend this fourth class.

Focus on Supply Chain Coordination through Stream Control

Several different approaches have been proposed, and are used nowadays in some companies to coordinate production entities in manufacturing systems. We restrict ourselves to the approaches using *tokens* to coordinate entities, because they are decentralized. The most known and used of these approaches is known as *Kanban*, which means signboard or placard in Japanese. *Kanban* is a tool related to the philosophy of "just in time" manufacturing, and is used to control production lines in a company. It was born in Japan in 50's, and was first successfully applied to Toyota by Taiichi Ohno. The principle of Kanban operation between a consuming work centre and a supplying work centre (or supplier) is as follows: the consuming work centre sends a *kanban* (signal) to the supplier (i) to order products (each *Kanban* represents a fixed quantity of products) that will be moved into a buffer stock, and (ii) to make the supplier produce this quantity of products. This operation is simple, and only works between every consuming work centre and its direct supplier(s). Since *Kanbans* are hard to apply in many contexts, many mechanisms were proposed to extend it:

- *At the company level*[1]: Here, PAC (Production Authorization Cards) System [Buzacott and Shanthikumar, 1992], Kanban, Extended Kanban and Generalized Kanban [Liberopoulous and Dallery, 2000] have been used to control the production of one company.
- *At the supply chain level*: Here, Porteus [2000] has propposed Responsability Tokens to operationalize Lee and Whang [1999]'s decentralized supply chain management scheme.

We now detail such coordination devices, and we start by the PAC system.

The PAC system is a decentralized approach to the coordination and control of material and information flows in multiple cell manufacturing systems. This approach generalizes other approaches, such as MRP (Material Requirements Planning), Kanban (Japanese card system) and OPT (Optimized Production Technology), among others. However, we know neither any comparison of this system with these other approaches, nor any real application.

Figure 4.2 shows a production cell (circle at the centre), two stores (dashed boxes at the left and at the right) and the minimal components of the PAC system. The production cell is the workstation itself, that is, the place where manufacturing operations

---

[1] Even if these coordination mechanisms are designed for a single company, their scalability permits extending them over several companies.

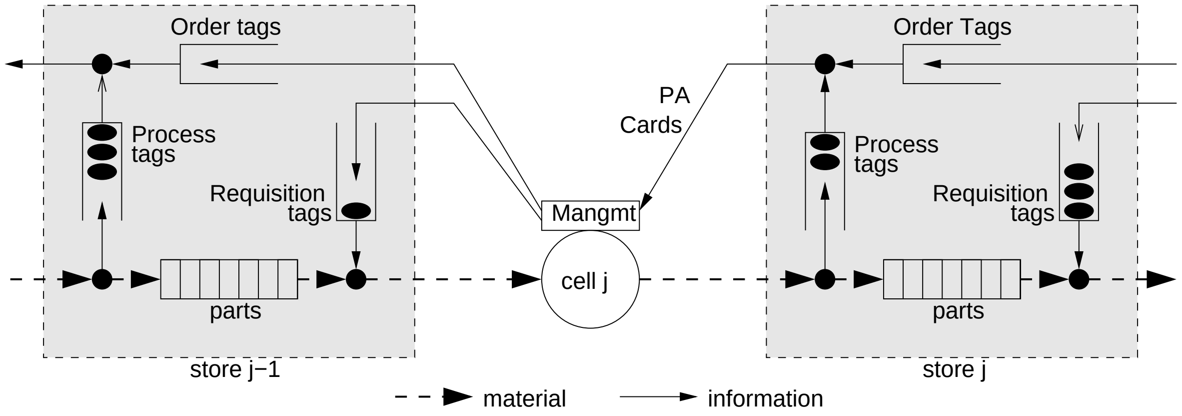


Figure 4.2: The PAC system [Buzacott and Shanthikumar, 1992].

are performed, and the stores are inventories in which products wait before and after their processing by the cell. The control of product streams in the cell and in the stores is insured by different types of tokens:

- *Requisition tags:* They are sent by cell j to store $j-1$ to ask store $j-1$ to ship an item to cell j *immediately*, or, if the store is empty, requisition tags wait in a queue at the store, until there is a unit of product available (so, this queue is filled with backorders).
- *Order tags:* They are sent by cell j to store $j-1$ to inform store $j-1$ that there will be a demand by the cell for a product *in the future*: for each order tag, there would be a requisition tag. These tokens allow long-term scheduling by propagating demand information in the production system.
- *Process tags:* They are sent by cell j to store j, when cell j ships an item to store j. When an order tag arrives at a store, it is matched with a process tag, and the match generates the PA card.
- *PA (Production Authorization) cards:* They are sent by store j to cell j to allow this cell to process a part. Moreover, when cell j receives a PA card, it sends order and requisition tags to store $j-1$.

Now, we give only a brief description of material and information flow control. The complete PAC system has more components: each type of product has its own set of tags, tags can have priority and be added to take stream convergences into account (e.g., for cells having many entry flows and only one exit stream) and stream divergences, order cancelations, treatment of defective products. . . .

The second example of a decentralized coordination mechanism is Responsibility Tokens, that were proposed by Porteus [2000] to further operationalize Lee and Whang [1999]'s decentralized supply chain management scheme, which is itself an operationalization of the decentralized management scheme made implicit by Clark and Scarf [1960]. It is a simpler mechanism than the PAC system, and is designed to coordinate several companies, whereas the PAC system is designed to coordinate manufacturing workstations in the same company. Responsibility Tokens are used as a mechanism for administering the transfer payments required to implement upstream responsibility. The idea is to base reimbursement on actual consequences of processing/delivering/shipping less than what was requested, rather than predicting these consequences in advance. The system works as follows: whenever an upstream company cannot meet the entire order placed by its customer company, this company will substitute Responsibility Tokens in place of the missing units. Customer companies will treat these tokens as physical units, and the financial consequences of the fact that these units are not real are assigned to the issuing player, when this non-reality of units incurs harmful consequences. Precisely, when a retailer has a real item, this item is sold to end-customers demanding it, but if this retailer only has tokens, it transforms them into penalties if end-customers want some of these items. Thus, companies are incited by financial penalties to deliver to their downstream companies as completely as possible, but these penalties are only paid when *stockouts arise*.

We now outline coordination in multi-agent systems.

4.1.2 Multi-Agent Coordination

Definition

Wooldridge [2001] defined coordination in multi-agent systems with the same definition as Malone and Crowston [1994], that was previously cited, i.e., coordinating is managing the dependencies between the activities of agents. Jennings [1993b] proposed a more precise definition, that is, the "process by which an agent reasons about its local actions and the (anticipated) actions of others, to try and ensure the community acts in a coherent manner". Durfee [2001] proposed a definition similar to Jennings, since he defined the coordination as an "agent's fundamental capability to decide on its own in the context of the activities of other agents around it".

This similarity is underlined by the fact that all authors agree to see coordination as a key concept in the field of multi-agent systems [Boutilier, 1999; Durfee, 2001].

Furthermore, coordination is a large problem, as stated by Durfee [2001], who thinks that a unique technique is not possible under every circumstances. In fact, if such a technique was possible, it would substitute for many constructs, such as companies, states, markets, teams. . ..

It is also important to note that cooperation and coordination are two different things, although closely related, because cooperation implies coordination, while coordination does not imply cooperation. For example, a chess-player agent coordinates its actions to maximize its utility against its opponent, but it does not cooperate with its opponent [Durfee, 2001].

We present now some techniques that have been developed to coordinate the agents' activities. First, we rely on Boutilier [1996]'s three classes of coordination techniques, which are used by many authors. These three broad classes are communication-based coordination, convention-based coordination and learning-based coordination. We now detail these three classes.

Communication-Based Coordination

Agents can coordinate by communicating together in order to find the best way, either not to disturb each other, or better, to work together. In this approach, agents negotiate to find an agreement on actions that each agent has to perform [Beck and Fox, 1994]. This negotiation may take, for example, the form of an auction, such as the Contract Net Protocol [Smith, 1980]. This negotiation may also be formalized as a DCSP (Distributed Constraint Satisfaction Problem), that has to be solved to find each agent's schedule of actions [Yokoo and Hirayama, 2000].

Durfee and Lesser [2001] proposed the Partial Global Planning (PGP) as another coordination technique. Since PGP interleaves coordination with action in order to coordinate in dynamic environments, this technique applies an iterative process to create, coordinate and execute plans. Moreover, decisions are taken despite incomplete and possibly obsolete information about environment. To this end, the idea of PGP is to make agents generate plans that are both partial, i.e., such plans are not for the overall system, and global, i.e., such plans are based on a non-local view of the problem. The non-local view required to build global plans is obtained by communication: each agent exchanges its local plan, and cooperates with other agents, so that to build this non-local view to have a better view of the system than the view the agent can perceive directly with its sensors [Wooldridge, 2001].

The solution to stream fluctuations that we propose in this dissertation belongs to this class of communication-based coordination mechanisms, because we propose that companies share information in order to limit these fluctuations by providing each agent with the required output of the multi-agent system.

Convention-Based Coordination

With such coordination techniques, agents have to or should follow social laws. An example in which agents have to obey is vehicle drivers on roads, who have to follow social laws, e.g., drive on the right side of the road, or do not pass on red light, because of the police, and of the danger of not doing so. But there are also conventions that agents do have to follow, but only should follow. For example, a person can go to the airport to pick up another person, without agreeing on the place where they will find each other. Only the arrival time of the flight is required, because they both know that they will meet at the baggage claim exit.

In general, if every agent follows the social laws, the system is supposed to be well coordinated. Hence, the problem for the designer of the multi-agent system is to find these "good" laws [Cachon, 2004; Cloutier et al., 2001; Cachon, 2004; Jennings, 1993b]. Some tools, such as game theory [Rosenschein and Zlotkin, 1994] described in Section 4.2, and the COOrdination Language (COOL) [Barbuceanu and Fox, 1995b] can help the designer in this job. According to Delgado [2002], there exist two ways to find these laws:

- *Off-line design*: Here, social laws are given *a priori* to agents by the designer of the multi-agent system.
- *Emergent design*: In this case, agents interact to decide together which laws to use in the current context. That is, agents have to agree on some common laws, but these laws are given *a priori* to the agents by the designer of the multi-agent system.

Similarly to emergent design, a common agreement also has to be reached in the next class of coordination techniques, i.e., in learning-based coordination. The difference with emergent design is that some laws are given *a priori* to agents in convention-based coordination, while agents have to find these laws in learning-based coordination. We now detail this latter class.

Learning-Based Coordination

In this class, agents learn to live together, that is, they learn which action should maximize their utility depending on the current context. This learning process is carried out with a learning process, as presented in Subsection 2.2.3. During this process, agents learn the social laws that were applied in the class of convention-based coordination. Hence, learning is a way for the designer to simplify his/her task, because agents find some good laws at his/her place. Furthermore, it is also a way to have a more adaptive coordination mechanism, because this mechanism follows environment changes [Paquet, 2003; Prasad and Lesser, 1999; Sugawara and Lesser, 1998].

Of course, learning-based coordination is based on learning techniques from artificial intelligence. As a first consequence, learning methods, such as Q-learning, Bayesian networks, or the extensions of Markov Decision Process (MDP), e.g., Partially Observable MDP, Multi-agent MDP and DECentralized MDP, have been applied to multi-agent systems [Paquet, 2003; Chalkiadakis and Boutilier, 2003; Boutilier, 1999, 1996]. As another consequence, the two main problems from learning in artificial intelligence also arise:

- Will the learning process of an isolated agent converge towards some behaviour, instead of having an agent that changes its behaviour all the time?
- When should an isolated agent stop exploring every possible strategy?

In addition to that, an additional problem arises, due to the fact that agents are not isolated, but interact together. In fact, agents have to coordinate their action with other agents, which greatly increases the complexity of the coordination problem [Chalkiadakis and Boutilier, 2003]. In fact, since the other agents change their behaviour through their learning process, it is much more complex to make a given agent learn to coordinate. In other words, the considered agent not only has to learn how to behave, but it also has to adapt its behaviour in order to adapt to the change in other agents' behaviour. In particular, agents may enter some cycles, in which, for example, an agent A coordinates with an agent B, then an agent C coordinates with the new behaviour of A, but because of this adaptation, the agent B changes its behaviour to coordinate again with C. We can see that the learning process may never stabilize on a stable coordination of the multi-agent system.

In addition to the above approaches of coordination, some others do not fit in Boutilier's classes. In particular, we now present two other techniques.

Commitments and Conventions as the Foundation of Coordination

Jennings [1993b] gathered every coordination technique in order to propose a unified one. To this end, he proposed to see commitments and conventions as the base of every coordination mechanism. Here, a commitment is defined as "a pledge, or a promise, to undertake a specified course of action" and a convention as "a means of monitoring commitments in changing circumstances". Note that convention does not have the same meaning here, as in convention-based coordination. The idea of this approach is that a commitment taken by an agent A provides other agents with predictability about A, so that, the other agents can predict what A could do in the future, and thus, the other agents can act according to this prediction. On the other hand, conventions provide A with flexibility in face of the changes in its environment, because conventions allow A, for example, to abandon an obsolete commitment, or to pay a penalty in exchange for changing a commitment.

Jennings assumed that the two concepts of commitment and convention are central for coordination, because all coordinations would be ultimately reduced to (joint) commitments and their associated (social) conventions. To support this hypothesis, he illustrated how some well-known coordination techniques, that do not mention the two concepts of commitment and convention, can be reformulated, so that they are compliant with this hypothesis.

Coordination through Joint Intentions

The above Jennings' hypothesis deals with joint commitments, which can be used to introduce coordination through joint intentions. In some circumstances, a joint intention can be defined as a joint commitment to perform a collective action in a certain mental state [Jennings, 1993a]. The mental state is important in this definition, because, when two groups of agents share a joint commitment, one group cooperates because its members also share a certain mental state, while the other group does not cooperate. In other words, the first group wants to cooperate, while the other group cooperates in a fortuitous way. We can note that we refer here to cooperation rather than to coordination, because, as we have just stated, joint intentions assume that agents want to work together [Wooldridge, 2001].

We now compare these coordination mechanisms with their equivalents in supply chain management, that were introduced in Subsection 4.1.1.

4.1.3 Comparison of Both Types of Coordinations

As a contribution, we propose to examine the differences and similarities in the coordination mechanisms that have just been outlined, in order to compare coordination in supply chain management with coordination in multi-agent systems. First, from the viewpoint of vocabulary, “cooperation” is used in these two fields, and the two words “collaboration” and “cooperation” refer to the same concept, but not in the same field. That is, companies in supply chains are said to collaborate, while agents cooperate.

Besides this little difference in vocabulary, we take the classes of multi-agent coordination techniques proposed by Boutilier [1996] as a basis of comparison of these two fields:

1. *Communication-based coordination:* Every coordination mechanism presented in Paragraph “Focus on Supply Chain Coordination through Stream Control” is a communication-based mechanism, because they use *tokens*, which are little pieces of information used to coordinate activities. We have presented in Subsection 4.1.1 such token-based approaches, because they are decentralized like multi-agent systems, which respects the autonomy of companies. Nevertheless, there also exist centralized approaches, which use communication too, but we do not insist on them, because centralization of decision requires companies to follow an external leader. In both cases of centralization and decentralization, we only note that coordination through stream control in supply chain always uses communication, like communication-based coordination in multi-agent systems.
2. *convention-based coordination:* This class of coordination mechanism corresponds to laws and contracts in supply chains:
    - *Laws:* Laws imposed on a company by the society correspond exactly to social laws in multi-agent systems, because both are given from the exterior. Furthermore, in both cases, agents (respectively companies) may propose to the rest of the system (respectively society) to change these laws.
    - *Contracts:* The little difference between convention-based coordination in multi-agent systems and contracts in supply chains, is that contracts have to be found by companies, while they are given *a priori* to agents. Therefore, contracts in a supply chain are mostly similar to learning-based coordination in multi-agent systems, because finding contracts can be seen as a learning process.
3. *learning-based coordination:* The link between learning-based coordination and contracts in the supply chain is stronger than the link outlined with convention-

based coordination: when agents learn to coordinate, their learning finds the best contracts ruling their interactions with other agents. In the same way, managers in supply chains "learn" which contracts are needed by their supply chains. When the environment changes, agents learn in order to find new contracts, which corresponds to contract updating in supply chains.

As we can see, there are some similarities between the coordination mechanisms in multi-agent systems and two (among four) of the mechanisms in supply chains. The two other mechanisms ("discounts" and "joint optimization") in supply chains also have correspondances in multi-agent systems:

- *Discounts:* Discounts are specific to systems managing money, and thus, are applied in some specific applications of multi-agent systems, e.g., e-commerce. More generally, market mechanisms are used to coordinate some multi-agent systems, and in particular, mechanism design may propose discounts in order to make the overall system works in the best way [Eymann, 2001]. For example, some research studies the way to price some shared resources, such as a networks [MacKie-Mason and Varian, 1995], in order to avoid congestions.
- *Joint optimization:* This approach proposes centralizing decision making, which is the contrary of the multi-agent philosophy, since this philosophy is to decentralize decision and control. At first glance, companies also prefer decentralization, because they prefer to keep their autonomy, but studying centralized decision making is interesting for at least two reasons:
  - companies may accept centralized decision making, i.e., they may accept to loose some control of their activities, if it can be proven that decentralization is much worse for them.
  - centralizing decision making allows using mathematical tools, such as Operations Research [Hillier and Lieberman, 1997], to find the best possible decision, which can be used as a reference to measure the efficiency of a decentralized approach to coordinate companies.

    Similarly, in the field of multi-agent systems, Durfee [2001] proposed optimality as one of the possible metrics of coordination. That is, an optimization is said to be optimal, if we can show that no other coordination leads to better results. Even though optimality is desirable, it is rarely feasible, because it requires a great deal of computation and many communications. Moreover, Durfee also proposed some other characteristics of coordination, that are also interesting for coordination in the field of supply chains. Here are some examples of such characteristics:

  - ∗ *scalability:* how does the performance of the system change when agents are added?
  - ∗ *heterogeneity:* how does the performance of the system change when the differences of agent types increase?
  - ∗ *robustness:* how does the performance of the system evolve when there are changes in the world, while some assumptions about this world were stated to design the coordination mechanism?
  - ∗ *overheads:* what are the computational and the communication overheads of the coordination mechanism?

> Although centralized decision making is not desirable neither in supply chains, nor in multi-agent systems, both fields also consider it. Indeed, we have seen that agents can make joint decisions in a similar way to joint optimization in supply chain management. Such a joint decision was referred as "coordination through joint intentions" in Subsection 4.1.2. We have noted that this coordination is in fact cooperation, because agents want to work together. In the context of supply chains, "joint optimization" also refers to cooperation, and more specifically to collaboration, since this word is preferred in supply chain management. As a consequence, the aims of joint intention and joint optimization are the same, i.e., to collaborate/cooperate, even if the means are different, i.e., centralized coordination based on mathematics vs. sharing of a certain mental state.

This section has introduced and compared coordination in two particular types of distributed system. We now present game theory, a formal tool to study coordination, and more generally, interaction between agents.

4.2 Game Theory

Game theory is a mathematical theory that studies *interactions* among self-interested agents ([Binmore, 1991] in [Wooldridge, 2001]). Precisely, game theory is the "systematic study of how rational agents behave in strategic situations, or in games", that is, in situations in which "the actions any one agent may take, will have consequences for others" [Jehle and Reny, 2000]. Such consequences for others are called *externalities*. This theory is used in the multi-agent systems field, because it gives a high point of view on such interactions. Wooldridge [2001] also noted how interesting is the fact that:

> John von Neumann, one of the founders of computer science, was also one of the founders of game theoy [von Neumann and Morgenstern, 1944]; Alan

> Turing, arguably the other great figure in the foundation of computing, was also interested in the formal study of games, and it may be that it was this interest that ultimately led him to write his classic paper *Computing Machinery and Intelligence* [Turing, 1950], which may be seen as the foundation of Artificial Intelligence (AI) as a discipline. However, since these beginnings, game theory and computer science went their separate ways for some time. Game theory was largely - though by no means solely - the preserve of economists, who were interested in using it to study and understand interactions among economic entities in the real world. Recently, the tools and techniques of game theory have found many applications in computational multi-agent systems research.

As example of multi-agent research, game theory was used to design protocols in electronic commerce [Asselin, 2002; Asselin and Chaib-draa, 2002; Ben-Ameur et al., 2002a; Sandholm, 1999]. But such trends towards game theory are very recent. To explain why economics in general, and game theory in particular, has not had more impact on artificial intelligence (AI) in general, and on multi-agent systems in particular, Boutilier et al. [1994] proposed two fundamental reasons. The first reason is substantive: "Economics is primarily concerned with explaining the decisions and interactions of rational self-interested agents (or communities thereof), or designing policies that influence these interactions to further certain global objectives", while "AI is largely (though not exclusively) concerned with constructing self-interested agents, the very entities economic theory takes for granted". The second reason is cultural, because economics and AI have a very different cultural heritage: philosophical epistemology for AI, e.g., the logicist McCarthy and the psychologists Newell, Simon and Minsky, and a strict bayesian setting for economics. We now introduce some game-theoretic concepts and notations, that are illustrated through examples.

4.2.1 Assumptions

When using game theory in this thesis, we make the following hypothesis:

- We consider games in the *strategic form*, that is, games in which all players make their decisions simultaneously.
- Players decide independently.
- Players' *rationality is infinite*. In general, rationality means that players decide in a way that maximizes their utility. In the context of games, it is not straight-

forward to apply this definition, because others' decision impacts on each player (presence of externalities). In consequence, players reason about their actions and others' actions. In particular, we also assume that infinite rationality allows players to repeat a reasoning an infinite number of times [Yildizoglu, 2003].

- There is no communication between players, and therefore, they cannot coordinate.
- We consider games of *complete information*, that is, players have a common knowledge of the game structure. In other words, players make their decisions with the same game, but with a different role.

4.2.2 Basic Definitions

With these assumptions, we can now give the definitions in game theory that are usefull to understand our work. The following definitions are from Jehle and Reny [2000], except when the contrary is stated:

Strategic form game: A strategic form game is a tuple $G = (R^i, u^i)_{i=1}^N$, where for each player $i = 1, \ldots, N$, R^i is the finite set of strategies available to player i, and $u^i : \times_{j=1}^N R^j \rightarrow \Re$ (i.e., $u^i : R^1 \times \ldots \times R^N \rightarrow \Re$) describes player i's utility/payoff as a function of the strategies chosen by all players.

As an illustration of this definition of games, consider the well-known game "Rock, Paper, Scissors", in which $N = 2$ players Player1 and Player2 are opponents. Each player has the same set of strategies, thus $R^1 = R^2$. The strategies in R^1 and R^2 are to play either Rock, Paper or Scissors, i.e., $R^1 = R^2 =$\{Rock, Paper, Scissors\}. In this case, each strategy is a single action, but in general, a strategy is a complete plan of actions. The utility functions $u^1 : \times_{j=1}^2 R^j = R^1 \times R^2 \rightarrow \Re$ and $u^2 : R^1 \times R^2 \rightarrow \Re$ describe the fact that:

- Rock beats Scissors;
- Scissors beats Paper;
- Paper beats Rock;

For example, for any $a > 0$, these relations correspond to the following utilities: u^1(Rock, Scissors) $= a$ and u^2(Rock, Scissors) $= -a$, and conversely, u^1(Scissors, Rock) $= -a$ and

		Player2		
		Rock	Paper	Scissors
Player1	Rock	$0; 0$	$-a; a$	$a; -a$
	Paper	$a; -a$	$0; 0$	$-a; a$
	Scissors	$-a; a$	$a; -a$	$0; 0$

for any $a > 0$

Table 4.1: The game "Rock, Paper, Scissors".

		Player2		
		Rock	Paper	Scissors
Player1	Rock	0; 0	-1; 1	1; -1
	Paper	1; -1	0; 0	-1; 1
	Scissors	-1; 1	1; -1	0; 0

Table 4.2: An instance of the game "Rock, Paper, Scissors", when $a = 1$ in Table 4.1.

u^2(Scissors, Rock) $= a$, as reflected in Table 4.1. What is important in this table is the relation between each player's utility when these players choose either Rock, Paper or Scissors. We can instanciate this general form of "Rock, Paper, Scissors" by choosing a positive value for a. We take $a = 1$ in Table 4.2.

We can note this is a zero-sum game, which is a game where players' payoffs add up to zero, i.e., $u^1 + u^2 = 0$ whatever is played by both players. For example, u^1(Rock, Scissors) $+ u^2$(Rock, Scissors) $= a - a = 1 - 1 = 0$. We do not give more details about this type of game, as games studied in this thesis neither sum to zero, nor to any other constant. We now present another concept.

Joint strategy: the set $r = \times_{i=1}^{N} r^i$ is a joint strategy in which each player i plays $r^i \in R^i$. The symbol $-i$ denotes "all players except player i", such as $r^{-i} \in R^{-i} = R^1 \times \ldots R^{i-1} \times R^{i+1} \times \ldots R^N$.

For example, in the game "Rock, Paper, Scissors", when Player1 plays Rock and Player2 Scissors, the joint strategy is $r =$(Rock, Scissors), in which case, $u^1(r) = u^1$(Rock, Scissors) $= 1$ and $u^2(r) = u^2$(Rock, Scissors) $= -1$. Precisely, each entry in Table 4.2 gives the outcome of a joint strategy. Concerning the notation "$-i$" introduced in the previous definition, r^{-1} represents what is played by Player2, because there are only two players in "Rock, Paper, Scissors".

4.2.3 Definition of Dominance Relations

At this point, we have a formalism describing games, and we know that agents maximize their utility, since they are rational. But we do not know how agents achieve this maximization, because they have to take others' decisions into account. Therefore, we now focus on tools for analyzing games.

We first look at two relations in a company's decision. The first presented relation is domination. It is strong, because Jehle and Reny [2000] say of such domination that:

> "whenever we attempt to predict the outcome of a game, it is preferable to do so, without requiring that players know a great deal about how their opponents will behave. This is not always possible, but when it is, the solution arrived at, is particularly convincing."

We introduce strictly dominant strategies first. Such a strategy is strictly superior to all other strategies, that is, it is the best one whatever is played by all other players:

Strictly dominant strategies: A strategy $\hat{r}^i$ for player i is strictly dominant if $u^i(\hat{r}^i, r^{-i}) > u^i(r^i, r^{-i})$, for all $(r^i, r^{-i}) \in R$ with $r^i \neq \hat{r}^i$.

Let us consider the new example depicted in Table 4.3. Now, strategies available by Player1 and Player2 are different: R^1 ={U, C, D} and R^2 ={L, M, R}. Playing M (middle) is a dominant strategy for Player2, because it always incurs the highest payoff (utility), whatever Player1 plays as strategy:

- If Player1 plays U (up), Player2 earns 4, which is the maximum between 0, 4 and 3, that it can have when it plays L, M, or R;
- If Player1 plays C (center), Player2 looses 1, which is the lowest loss between -2, -1 and -4;
- If Player1 plays D (down), Player2 earns 8, which is the maximum between 4, 8 and 6.

In all cases, M has the highest payoff for Player2: M is therefore a dominant strategy. As M is dominant for Player2, Player1 knows for sure that his/her opponent is going

		Player2		
		L	M	R
Player1	U	3; 0	0; 4	2; 3
	C	1; -2	3; -1	4; -4
	D	2; 4	-1; 8	2; 6

Table 4.3: Strictly dominant strategy.

to play only M. Player1 knows that about Player2, because both players have common knowledge of the game, since we consider games of complete information, and reason in the same way, since they both have an infinite rationality. In conclusion, Player1 chooses C in order to win 3, because s/he knows that only (U, M), (C, M) and (D, M) may be played, and s/he prefers (C, M).

We can also consider the latter game to show that stricly dominant strategies do not always exist. For example, if Rock was strictly dominant in Tables 4.1 and 4.2, both players would only play this strategy, and never Paper or Scissors. This is not the case, because the game "Rock, Paper, Scissors" has no strictly dominant strategy, and thus, players choose any of the three possible strategies.

In general, strictly dominant strategies are very rare. When there are none, we consider the second relation, i.e., strictly dominated strategies, because "it may still be possible to simplify the analysis of a game by ruling out strategies that are clearly *un*attractive to the player possessing them [Jehle and Reny, 2000]". This is the motive for strictly dominated strategies:

Strictly dominated strategies: Player i's strategy $\hat{r}^i$ strictly dominates another of his strategies $\bar{r}^i$, if $u^i(\hat{r}^i, r^{-i}) > u^i(\bar{r}^i, r^{-i})$ for all $r^{-i} \in R^{-i}$. In this case, we also say that $\bar{r}^i$ is strictly dominated in R.

Again, the game "Rock, Paper, Scissors" has no strictly dominated strategy in Tables 4.1 and 4.2. For example, if Rock was strictly dominated by Paper, players would never play Rock, which is not true.

Let us now consider the game in Table 4.3(a) [Jehle and Reny, 2000]. Neither player has a strictly dominant strategy, but each player has a strictly dominated strategy, i.e., an unattractive strategy. In fact, Player1 always has a higher outcome in D than in C, whatever is decided by Player2. In the same way, Player2 always has a higher outcome in R than in M, whatever is made by Player1. Therefore, strategies C and M are

Player2

Player1

	L	M	R
U	3; 0	0; -5	0; -4
C	1; -1	3; 3	-2; 4
D	2; 4	4; 1	-1; 8

(a) ***Original game.***

Player2

Player1

	L	R
U	3; 0	0; -4
D	2; 4	-1; 8

(b) ***Table (a) after the first turn of eliminations of strictly dominated strategies.***

Player2

Player1

	L	R
U	3; 0	0; -4

(c) ***Table (a) after the second turn of eliminations of strictly dominated strategies***

Player2

Player1

	L
U	3; 0

(d) ***Table (a) after the third turn of eliminations of strictly dominated strategies***

Figure 4.3: Strictly dominated strategy [Jehle and Reny, 2000].

dominated by another strategy for both players: they can be removed, because they should not be played by both players, that are assumed as rational. The reduced form of game 4.3(a) obtained by removing these two strictly dominated strategies is presented in Table 4.3(b).

In this second table, we can see that U becomes a dominant strategy for Player1. This strategy was not dominant at the beginning (Table 4.3(a)), but this dominance appeared when C and M were eliminated, because they are dominated. As U is dominant for Player1, D is dominated by U. Thus, we can remove D in a second turn of elimination of strictly dominated strategies. The result of this elimination is given in Table 4.3(c). As Player1 will only play U in Table 4.3(c), R is strictly dominated by L for Player2, because Player2 earns 0 instead of -4. As a consequence, R can be ignored in Table 4.3(c), which gives Table 4.3(d). Finally, we know exactly what both players are going to play: the solution of the game in Table 4.3(a) is (U, L), i.e., the only remaining joint strategy in Table 4.3(d).

4.2.4 Definition of Concepts of Solution

Up to now, we have only compared strategies for individual players. In the definition of Pareto-improvement, Pareto-efficient, social welfare and Nash equilibrium, joint strategies are compared together. As these concepts define the possible outcomes of a game, they are called *concepts of solution*. The first two of these concepts of solution are now introduced.

Pareto-improvement: the joint strategy $\hat{r}$ Pareto-improves the joint strategy r, if for each player i, $u^i(\hat{r}) \geq u^i(r)$, and there is at least one player for which $u^i(\hat{r}) > u^i(r)$.

This definition means that a Pareto-improvement can be made only when it is possible to make someone better off and no one worse off. If there is no way at all to make a Pareto-improvement from a given joint strategy, then we say that this joint strategy is Pareto-efficient. That is, a situation is Pareto-efficient if there is no way to make someone better off without making someone else worse off:

Pareto-efficiency: the joint strategy $\hat{r}$ is Pareto-efficient if for each player i and for any joint strategy $r \neq \hat{r}$, $u^i(\hat{r}) \geq u^i(r)$.

In consequence, there may be several Pareto-efficient joint strategies. For instance, in zero-sum games, i.e., games where the sum of all players' payoff is zero, such as "Rock, Paper, Scissors" in Tables 4.2, no Pareto-improvement can be made, and therefore, in Table 4.2, i.e., the nine joint strategies of this game, are Pareto-efficient.

On the contrary, some entries in Table 4.3 are not Pareto-efficient. In particular, (U, R) can be Pareto-improved, because if Player1 switched from U to D, it would increase other player's payoff without changing his own payoff. This second entry, (D, R) cannot be Pareto-improved, because in all other entries, either one player is better off and the other is worse off, or both players are worse off. Therefore, (D, R) is Pareto-efficient, even if we have seen that R is strictly dominated for Player2. Similarly, (U, L) Pareto-improves (C, M) and (C, L). Finally, we can check that (U, L), (U, M), (D, L), (D, M) and (D, R) are Pareto-efficient.

Let us consider now another concept of solution, called the maximum of the social welfare. In general, it is difficult (and even impossible, when we want to have some desirable properties for this function, as proven in Arrow's impossibility theorem [Arrow, 1951]) to build a function representing the social welfare, that is, the aggregation of a

		Player2		
		L	M	R
Player1	U	$3_{=3+0}$	$-5_{=0-5}$	$-4_{=0-4}$
	C	$0_{=1-1}$	$6_{=3+3}$	$2_{=-2+4}$
	D	$0_{=2+4}$	$5_{=4+1}$	$\mathbf{7}_{=-1+8}$

Table 4.4: A possible measure of the social welfare in Table 4.3(a).

group of agents' utilities, because an agent's utility cannot be compared with another agent's utility. In fact, agents have different points of view about a situation, and thus, their utility is measured in different units. For example, money, happiness and health cannot be compared together [Jehle and Reny, 2000]. Therefore, in this thesis, we focus on player's utilities that can be compared by using money as the common unit. Conversely to other game-theoretic definitions presented in this section, the following definition is from Sandholm [1999]:

Social welfare : the set of players playing the joint strategy r has a social welfare $U(r) = \sum_i u^i(r)$.

With this definition, the social welfare measures the global good of the agents as the sum of all agent's payoffs/utilities, when these agents use a common metrics for their utility. If we assume that Player1's utility is equivalent to Player2's, we can calculate the social welfare in all entries in the game in Table 4.3(a); this calculation is achieved in Table 4.4. We saw earlier that the solution of this game is (U, L), that is, when we eliminate recursively every strictly dominated strategy, (U, L) is the only joint strategy surviving this elimination. Though, the issue is that (U, L) incurs a social welfare of $3 + 0 = 3$, which is very far from the maximum of $-1 + 8 = 7$ incurred by (D, R). Player2 would earn 8 in this joint strategy, and the Player 1 would loose 1: Player2 could therefore give 5 to Player1 to give him an incentive to reach this other joint strategy. In this case, both players would increase their income: Player1 would have 4 instead of 3, and Player2 would have 3 instead of 0.

Unfortunately, such arrangements are not allowed by game theory; if such transfers are possible, they have to be included in the definition of strategies, i.e., *before* writing the outcomes in Table 4.3(a). This design of interaction rule that will, in principle, yield such a desired outcome is called *Mechanism Design* [Varian, 1995]. In other words, this field aims at designing the rules of the agents' environment to reach some desirable joint strategies, i.e., mechanism design defines "the game that the agents must play, so that the collective good of all agents is maximized when each agent adopts the game-theoretic solution that maximizes its own utility". One basic requirement of mechanism design is "to ensure that *externalities* are made explicit, where externalities are the "effects on other agents' utility that are not recognized in the individual agents' transactions" [Russell and Norvig, 2003].

The bullwhip effect studied in this thesis is an externality, because each company tries to minimize the cost of its inventory system while ignoring that this optimization has effects (i.e., order variability) on the rest of the supply chain. However, we do not consider mechanism design as such in this thesis, because no economics principles were used to propose our coordination mechanism. In fact, though we design some rules of interaction to reduce the bullwhip effect, i.e., some ordering schemes, these rules are designed to stabilize ordering and shipping streams in the supply chain, rather than make company-agents internalize an externality, by making this externality explicit.

Finally, the last concept of solution that we use is the Nash equilibrium in pure strategy[2]. This well-kwown concept was proposed by John Nash [1951] in 1951. It is the single most important equilibrium concept in all of game theory, because it is the most general equilibrium that is known. Jehle and Reny [2000] defines this equilibrium as:

Pure strategy Nash equilibrium : Given a strategic form game $G = (R^i, u^i)_{i=1}^N$, the joint strategy $\hat{r} \in R$ is a pure strategy Nash equilibrium of G, if for each player i, $u^i(\hat{r}) \geq u^i(r^i, \hat{r}^{-i})$ for all $r^i \in R^i$.

Jehle and Reny [2000] makes the following comment about this concept of solution:

> Informally, a joint strategy $\hat{r} \in R$ constitutes a Nash equilibrium as long as each individual, while fully aware of the others' behaviour, has no incentive to change his own. Thus, a Nash equilibrium describes behaviour that can be rationally sustained.

This is true as long as the other player does not change his/her decision, that is, as long as deviances are *unilateral*. No player would like to change what he plays, regarding what is played by the other player. A Nash equilibrium is therefore a *"shaft state"*,

---

[2] Pure strategy means the player i has no probability distribution over his set of strategies R^i. The contrary is mixed strategies, where the player i may play for example $\hat{r}^i$ during 25% of the time and $\bar{r}^i$ during the rest of the time, i.e., the player i randomizes among his choices. We do not consider such probability distribution in this thesis, because we will see in Subsection 8.4.5 that our simulation does not take into account the switching of strategy during a simulation

or, from a mathematics point of view, a fixed-point of a function describing the game [Nash, 1951]. In general, there can be several pure strategy Nash equilibria or none[3]. These two cases can be a problem, because either we do not know which equilibrium will occur, or we do not know which decision players will prefer.

We have previously established that players play the joint strategy (U, L) in Table 4.3(a). We can also find this result with the concept of Nash equilibrium, in order to illustrate the meaning of this equilibrium. To do so, we test the nine entries in Figure 4.3(a), to check if each player has no incentive to leave the considered entry. For example, for the entry (U, L):

- The Player1 has an outcome of 3, which is better than 1 if he plays C instead of U and better than 2 if he plays D instead of U;
- The Player2 has an outcome of 0, which is better than -5 if he plays M instead of L and better than -4 if he plays R instead of L.

Let us come back to Table 4.3(a) in order to describe one of the sophistications of the Nash equilibrium: the *iterated dominance equilibrium*. In Table 4.3(a), (U, L) is the solution obtained by recursive eliminations of dominated strategies of the game: (U, L) is an iterated dominance equilibrium of this game. The concept of iterated dominance equilibrium is stronger and encompasses the Nash equilibrium concept. That is, an iterated dominance equilibrium is a Nash equilibrium with the particularity of being preferred by players to any other equilibria. Rasmusen [1994] defines such iterated dominance equilibria as:

Iterated dominance equilibrium: An iterated dominance equilibrium is a strategy profile found by deleting a dominated strategy from the strategy set of one of the players, recalculating to find which remaining strategies are dominated, deleting one of them, and continuing the process until only one strategy remains for each player.

Nevertheless, even though iterated dominance equilibria are stronger than Nash equilibria, they are not the panacea. In particular, iterated dominance equilibria may not be Pareto-efficient. To see this, recall that we have shown that (U, L) in Table 4.3(a) is the Nash equilibrium remaining in Table 4.3(d) after the recursive elimination of strictly dominated strategies, while (D, M) is a Pareto-improvement on this equilibrium.

---

[3] In mixed strategies, there is always at least one Nash equilibrium [Nash, 1951].

		Girl	
		Bach	Stravinsky
Man	Bach	$\zeta; \epsilon$	$\eta; \eta$
	Stravinsky	$\eta; \eta$	$\epsilon; \zeta$

subject to $\zeta > \epsilon > \eta$

Table 4.5: The game “Battle of the Sexes” in general.

Finally, note that Pareto-efficient joint strategies are different from Nash equilibria, because some players may have an incentive to deviate from a Pareto-efficient joint strategy, which would increase the utility of such player, but also decrease the utility of some other players. For example, (C, M) is a Pareto-improvement on (U, L) in Table 4.3(a). Moreover, (C, M) has a higher social welfare ($3 + 3 = 6$), than (U, L) ($3 + 0 = 3$). Although, (C, M) is not an equilibrium, because both players have an incentive to unilaterally deviate from it.

4.2.5 An illustration: The Battle of the Sexes

Consider a last example of game, called the “Battle of the Sexes”. In this game, two lovers have to decide what they are going to do tonight. The Man would like to listen a Bach symphony, while his girlfriend prefers a Stravinsky symphony. But they wish above all to be together. Table 4.5 summarizes these preferences, where, $\zeta > \epsilon > \eta$, to respect players’ preferences. For example, the Girl prefers Stravinski than Bach, when the Man also chooses Stravinski.

Table 4.6 is an instanciation of Table 4.5, i.e., the values of ζ, ϵ and η are chosen such as the constraints $\zeta > \epsilon > \eta$ are respected. On the one hand, people’s utility is zero in this instance of the “Battle of the Sexes”, when they make different choices, because they listen to the music they like, but they are alone. On the other hand, people’s utility is superior to zero when they make the same choice as the other player, but only one of the two characters listens to the music s/he likes, and has therefor a utility of 2 instead of 1.

The analysis of this game shows that there are neither dominant strategies, nor dominated strategies, but two Nash equilibria: (Bach, Bach) and (Stravinsky, Stravinsky). Nothing differentiates these two equilibria, and in particular, none of them is an iterated dominance equilibrium. Moreover, a Pareto-improvement can by made from

		Girl	
		Bach	Stravinsky
Man	Bach	2; 1	0; 0
	Stravinsky	0; 0	1; 2

Table 4.6: An instance of the game "Battle of the Sexes" in Table 4.5.

entries (Stravinsky, Bach) and (Bach, Stravinsky) to entries (Stravinsky, Stravinsky) and (Bach, Bach). Therefore, the two Nash equilibria are Pareto-efficient, but none of these equilibria Pareto-improves the other equilibria.

This kind of game is called a *coordination game*. For example, the choice of television standards or disk drives for Macintosh and PC corresponds to this kind of game. Each manufacturer would like to impose the use of its own standard, but in case of disagreement with the competition, consumers may refuse to buy the product [Yildizoglu, 2003, pp. 27].

4.2.6 Remarks

All these definitions from game theory will be adapted to our needs later in Section 8.3. Another classic example, called the "Prisoners' Dilemma", will then illustrate these adaptations in Subsection 8.3.2.

We finish this presentation of game theory with a general remark about its limits. In fact, Rosenschein and Zlotkin [1994] notes the issues related to the study of human beings, that do not apply to agents:

> Game theory tools have been primarily applied to human behaviour, but in many ways they fall short: humans do not always appear to be rational beings (i.e., utility maximizers), nor do they necessarily have consistent preferences over alternatives. Automated societies, on the other hand, are particularly amenable to formal analysis and design. Automated agents can exhibit predictability, consistency, narrowness of purpose (e.g., no emotions, no humor, no fears, clearly defined and consistent risk attitude, and an explicit measurement of utility.

Nowadays, game-theoretic agents can be used to model intelligent agents and business entities, even if they are not very efficient for human beings. In fact, Cachon and Netessine [2003] noted that the application of game theory to supply chain management is beginning, even if a few early papers also exist [Shubik, 1960], and much work is still possible. The next section outlines the computational complexity raised by game theory.

4.3 Computational Complexity and Game Theory

The main solution concept in game theory is the Nash equilibrium [Lipton and Markakis, 2004], which explains why most studies on computational complexity focus on this concept. This explains why the computation of such equilibria is currently a very active research field. Papadimitriou and Roughgarden [2004] presented a polynomial-time algorithm for computing (finding in a given game) a Nash equilibrium as the "holy grail" of this research field. In fact, Papadimitriou [2001] thinks that this problem is not easy, i.e., it is harder than P (the class of problems considered as easy, because they can be solved in a polynomial time), even if it must be easier than NP-hard (NP for Non-deterministic Polynomial, one class of hard problems, because they cannot be solved in a polynomial time).

Similarly, it was shown that determining if Nash equilibria with certain natural properties (e.g., is the equilibrium Pareto-efficient? is there more than one equilibrium? is there an equilibrium where player one never plays `A`?) exist is NP-hard, and the counting of Nash equilibria is $\#P$-hard (another class of problems considered as hard) [Conitzer and Sandholm, 2003].

As we focus in this dissertation only on pure strategies, instead of on all Nash equilibria, we can wonder if this hardness remains. Gottlob et al. [2003] answers "yes" to this question. In fact, they have shown that determining the existence of a pure Nash equilibrium is NP-hard, even in very restrictive settings. Fortunately, we can find examples in which a Nash equilibrium can be computed in a polynomial time [Fabrikant et al., 2004]. The good news is that determining the existence of a pure Nash equilibrium and computing all such equilibria is feasible in logarithmic computational space [Gottlob et al., 2003], but Gottlob and his co-workers have said nothing about computational time.

To have an insight into the computational time for computing a Nash equilibrium, let us consider the method used in Section 4.2, in which each entry of the game is checked by hand. For instance, in the game "Battle of the Sexes" in Table 4.6, two players each have two possible strategies. Therefore, the game is represented by a 2×2

matrix in which the $2 * 2 = 4$ entries have to be checked. But when a third player with two possible strategies is added, the game is represented by a $2 \times 2 \times 2$ matrix in which $2 * 2 * 2 = 8$ entries have to be checked. This algorithm thus incurs a combinatorial explosion, because there is an exponential relation between the number of entries to check and the number of players. This shows that the algorithm used to find Nash equilibria by hand is exponential, that is, NP, while Papadimitriou [2001] hopes that a quicker algorithm exists. This better algorithm is required if we want to analyze games with many players.

From a more practical point of view, the above results imply that when we enumerate all pure Nash equilibria, the calculation may last a very long time (because time complexity is thought to be harder than P), without requiring an excessive memory (because required space is logarithmic [Gottlob et al., 2003]). Therefore, we need a means to accelerate the enumeration of pure Nash equilibria in Chapter 8. This is the reason why one would first simplify the game by removing all strictly dominated strategies. In fact, removing all strictly dominated strategies accelerates the enumeration of Nash equilibria without loosing any of them [McKelvey et al., 2004]. McKelvey et al. [2004]'s Gambit 0.97.05 achieves these tasks of eliminating recursively strictly dominated strategies and enumerating pure Nash equilibria for us. Gambit is a free software licensed under the Free Software Foundation [2004b]'s GNU General Public License for analyzing games according to game theory principles. Gambit can be used to find pure Nash equilibria. Here, Gambit applies an algorithm based on a method called Simplicial Subdivision [McKelvey and McLennan, 1996], but its complexity is unknown [Lipton and Markakis, 2004]. We only know from some experiments that it is a complex algorithm.

4.4 Conclusion

As the bullwhip effect can be seen as a problem of coordination in supply chains, this chapter has presented how to coordinate supply chains and multi-agent systems. With respect to coordination in these two fields, we have pointed out some similarities and some differences between these two fields, which is a contribution of this dissertation.

As coordination is related to interactions, we have next presented game theory as a tool to study these interactions. In particular, we have introduced this theory, and presented its essential concepts. Next, we have illustrated this theory with some examples.

Finally, we have outlined the difficulty of applying game theory to computer science. Such application is recent, but researchers think that computing game-theoretic concepts cannot be considered an easy problem, i.e., a problem that can be solved by a polynomial-time algorithm.

The next chapter is the first one describing the core of our research. To this effect, we show how stream fluctuations are induced in distributed systems, and we propose a coordination technique to limit this cause of stream fluctuations.

Chapter 5

Delays as a Cause of The Bullwhip Effect

The previous chapter has concluded the introduction of the background of this research with the presentation of coordination and game theory. In this chapter, we present the theory on which our solution to stream fluctuations is based. For that purpose, we show in Section 5.1 why delays lead to an increase of stream fluctuations in the particular case of supply chains. Two principles are next suggested to reduce this cause of stream fluctuations in Section 5.2. These two principles are then instanciated in two ordering schemes for the Québec Wood Supply Game (QWSG) in Section 5.3. The supply chain behaviour induced by one of these two ordering schemes is then presented in Section 5.4. Section 5.5 illustrates the use of this ordering scheme with a more realistic example than the QWSG.

Since we introduce and illustrate how delays incur stream fluctuations, and how we propose to reduce this issue on the case of supply chains, we show in the last part of this chapter, how to adapt this solution to any multi-agent system. To this end, we show in Section 5.6, that a simple replacement of words translates the content of this chapter into a multi-agent context.

5.1 Why Delays Cause the Bullwhip Effect

The Québec Wood Supply Game (QWSG) was designed to teach the phenomenon of stream fluctuations in supply chains. We recall that stream fluctuations in supply chains are also known as the bullwhip effect. In the QWSG, each player takes the role of a company that places orders according to its incoming orders and inventory level. This game simulates product and ordering streams, and the only decision taken by players concerns the placement of orders. Hence, the bullwhip effect in the QWSG is due to the method used by players to place orders. We present the QWSG in details in Subsection 6.1.2.

It is important to note, that in our research we replace human players with intelligent agents, and orders placed by these agents are ruled by their ordering scheme. Our problem in this chapter is to propose a behaviour to a company, so that the bullwhip effect is reduced in the QWSG. Before doing that, we have to understand what makes the bullwhip effect appear. In this section, we first show that only ordering and shipping delays can explain the bullwhip effect our model and with intelligent agents, then we detail how delays induce this effect.

Several causes of the bullwhip effect were proposed in the literature for real supply chains, but the QWSG is so simple, that few of these causes occur in it. From our point of view, only two of the proposed causes of the bullwhip effect, recalled in Section 3.2, can be found in the QWSG and in the Beer Game: demand signal processing and misperception of feedback.

1. *Demand signal processing:* This cause of the bullwhip effect was proposed by Lee et al. [1997a,b]. They explain that each company uses its incoming orders to forecast its future demand, while this demand can be very different from market consumption because of clients' forecasts. Since no forecasts were used in the ordering rules used by our intelligent agents playing the QWSG[1], this can explain the bullwhip effect in the board version of the QWSG, because human players intuitively forecast their future incoming demand, but not in our simulation.

2. *Misperception of feedback:* The second possible cause of the bullwhip effect in the QWSG, was proposed by Sterman [1989], who studied the behaviour of human players in the "father" of the QWSG, i.e., in the Beer Game. According to Sterman, players do not understand supply chain dynamics, and thus, do not exhibit the best behaviour when they place orders.

Only the above second cause could explain why we still had a bullwhip effect when software agents replaced human players. In fact, players' understanding of the supply chain dynamics was not used directly in our experiments, but the ordering policies that

---

[1] In fact, we assume there is no forecast, because we base the current order on the last demand, but we can also see this method as a forecast based only on the last demand.

they apply could be designed so that these dynamics are taken into account. When we looked for efficient ordering policies, we found that the cause "misperception of feedback" can be detailed as "ordering and shipping delays". In other words, we can see ordering and shipping delays as one of the *refinements* of the cause "misperception of feedback".

Since only ordering and shipping delays can explain the existence of the bullwhip effect when agents play the QWSG, we now show why ordering and shipping delays induce the bullwhip effect in our model. In deed, the ordering scheme that we give to agents has to take this cause into account in order to reduce the bullwhip effect, and if possible, to minimize also company-agents' individual cost and/or the overall supply chain cost. We tried several ordering schemes (that are not presented here), but these schemes either induced the bullwhip effect, and/or did not manage inventory. The problem of designing these schemes is now outlined. Outlining this problem allows understanding why delays incur the bullwhip effect.

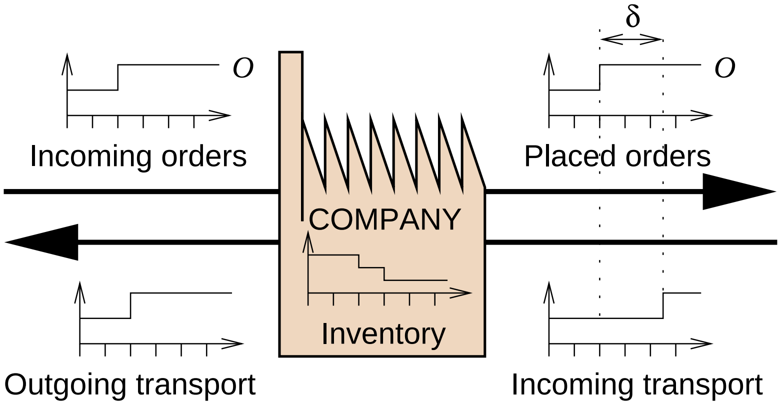
The basic idea to avoid the bullwhip effect in the QWSG is that, *if companies' orders follow their clients' demand* with a lot-for-lot ordering policy, *there is no bullwhip effect, but inventories fluctuate greatly.* In other words, either there is a bullwhip effect or inventories fluctuate greatly. This fact is illustrated in Figures 5.1(a) and 5.1(b), that represent a company travelled by an ordering and a product stream. In these figures, we assume each company places orders strictly equal to its demand, following a strict lot-for-lot ordering policy. We now detail Figure 5.1(a) in four points, to show why companies prefer the bullwhip effect, rather than use the lot-for-lot policy.

1. The lot-for-lot ordering policy eliminates the bullwhip effect, because each company has the same ordering pattern as its client and thus, as the market consumption. Therefore, the two curves Incoming orders and Placed orders are identical. Since the bullwhip effect is measured as the standard-deviation of placed orders, we can see that the standard-deviation of each company's orders is exactly the same as the standard-deviation of its client's orders, and therefore, as the standard-deviation of the market consumption[2]. This explains why a lot-for-lot ordering policy eliminates the bullwhip effect.

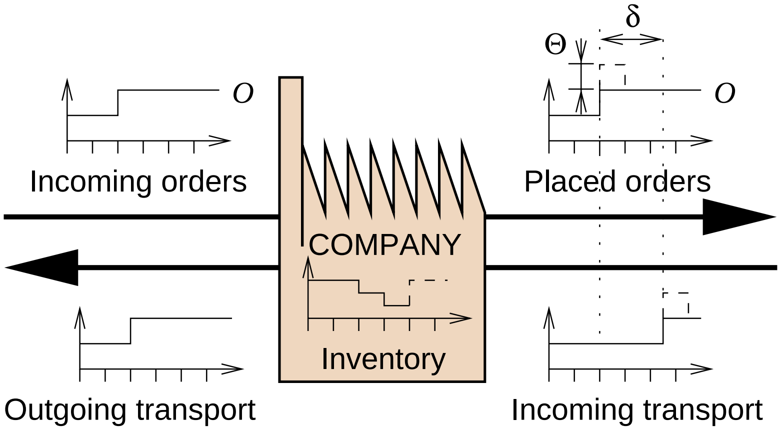
2. The considered company tries to fulfill its entire demand, and thus, the two curves

---

[2] We can notice here that companies could smooth the market consumption when they place orders, that is, companies could reduce the bullwhip effect by ordering in a more steady way than the market consumes. In this case, at least one company would absorb order fluctuations by allowing its inventory to fluctuate, and thus, this company would have a higher inventory level than what is made in this dissertation. We do not focus on such a smoothing technique, whereas it could be an interesting future work.



(a) ***Lot-for-lot ordering policy***



(b) ***Lot-for-lot ordering policy, market consumption***

Figure 5.1: Lot-for-lot ordering policy with (O, Θ) orders.

Incoming orders and Outgoing transport are the same, that is, as many products are shipped as ordered. The curve Outgoing transport in Figure 5.1(a) is valid as long as no stockouts occur by the considered company.

In short, these two points say that the three curves Incoming orders, Placed orders and Outgoing transport are similar. The third point below says that the fourth curve also has the same pattern, but with a temporal shift.

3. The fourth curve Incoming transport has the same pattern as the three other curves, except that it is delayed by δ in comparison with the three other curves. This curve represents the reception of products by the company. This temporal

shift corresponds to the ordering and shipping delays, because items ordered by the company are not immediately received. The problem with lot-for-lot orders is that the inventory is not managed, because this temporal shift makes inventory decrease (respectively increase, when we inverse the pattern of "Incoming Orders"). In fact, the company ships more (respectively less) products than it receives during the ordering and shipping delays.

Note that Incoming transport has the same pattern as the three other curves only when the supplier has no stockouts, because the supplier is assumed to want to fulfill its entire demand, like the considered company.

4. Since every company wants to avoid stockouts (respectively huge inventory), rather than eliminate the bullwhip effect, it does not use the lot-for-lot ordering policy. Instead, it overorders (respectively underorders) to stabilize its inventory, which amplifies the demand variabilities, because the company overorders (respectively underorders) when the demand increases (respectively decreases). This shows that the *bullwhip effect always appears* each time the market consumption has an infinitesimal change, if companies want to *keep a steady inventory*[3].

   We can note here, that some of the other causes of the bullwhip effect presented in Section 3.2, induce the bullwhip effect even with a steady demand, while delays amplify order fluctuations, but do not induce fluctuations when the demand is steady. Specifically, the bullwhip effect amplifies, because if a retailer overorders to stabilize its inventory, a worse phenomenon takes place with its suppliers: since the demand variation is now bigger, their inventories decrease much more, and thus they must overorder more.

As we can see, our problem is not only to reduce the bullwhip effect, because company-agents in the QWSG only have to apply the lot-for-lot ordering policy to eliminate this effect, but we also have to manage inventories in order to avoid stockouts and high inventory levels. In our solution, we propose an ordering policy with a unique order amplification for each change in market consumption. Since companies have to know the market consumption, this solution is the same as the one proposed by Lee and his colleagues to improve demand forecasting updating, because companies have to share their incoming orders information with their suppliers. Precisely, companies signal to their suppliers when they over- or underorder. This information sharing is presented in Figure 5.1(b), in which each company uses a vector (O, Θ) of two orders:

---

[3] This is true when smoothing is not considered. Figure 5.1(a) shows that if companies smooth their demand when they transmit it to their suppliers (in placed orders), their inventory will fluctuate, as stated in the previous footnote. Therefore, a company reducing the bullwhip effect with some smoothing technique has to increase its inventory to avoid stockouts. In other words, either inventory fluctuates to place steady orders, or orders fluctuate to stabilize inventory. This increase of inventory costs to the company money, but only its suppliers directly profit from the bullwhip effect reduction.

1. orders O follow the lot-for-lot policy to avoid the bullwhip effect;
2. orders Θ are used to order more or less products than O to stabilize inventory level.

We now present the two principles ruling the use of O and Θ.

5.2 Two Basic Principles for Reducing the Bullwhip Effect Caused by Delays

The bullwhip effect can be considered as a coordination problem between autonomous companies [Cooper et al., 1997], and these companies can be considered as agents. Therefore, we have looked for a coordination technique in multi-agent systems [Durfee, 2001; Durfee and Lesser, 1989; Lizotte and Chaib-draa, 1997; Shen and Norrie, 1999b; Paquet, 2001; Barbuceanu and Fox, 1995b], and industrial management [Porteus, 2000; Liberopoulous and Dallery, 2000; Chandra and Fisher, 1994; Buzacott and Shanthikumar, 1992] fields to coordinate the supply chain, so that the bullwhip effect can be reduced. The required coordination technique needs to:

1. preserve the autonomy of each entity, that is, the supply chain should not have a central coordinator;
2. avoid companies transmitting their own information to others, in order to reduce the communication load and keep information *secret*.

The coordination mechanism that we now propose meets the first need, but only partially fulfills the second requirement. In fact, we will see soon that our mechanism needs the market consumption information to be transmitted by companies, but this is the only shared information and companies do not have to share anything else, such as their inventory level information.

To design this coordination mechanism, we were inspired by Porteus [2000]'s Responsibility Tokens presented in Subsection 4.1.1, even if our ordering mechanism is eventually very different from this. In particular, we used tokens to share information supporting coordination, while a responsibility token represents an unfulfilled order, that may eventually lead to a financial penalty. This means that our tokens represent orders (order stream), while responsibility tokens represent products that were not shipped (product stream).

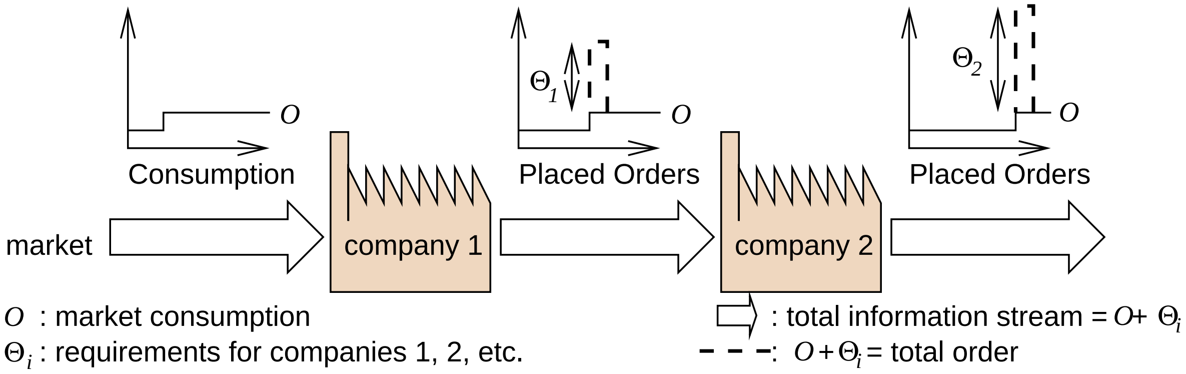


Figure 5.2: Information streams cut into two parts.

As stated with Figure 5.1(b), we also think the most appropriate technique to reduce the bullwhip effect is based on information sharing with (O, Θ) orders. By doing so, we focus on communication-based coordination in accordance to Boutilier [1996]'s classification of coordination techniques presented in Subsection 4.1.2 (convention-based coordination, communication-based coordination and learning-based coordination).

In short, we proposed a token-based coordination mechanism, which, to our knowledge, is unknown as such in the multi-agent field [Nwana et al., 1996; Boutilier, 1996], even if it belongs to communication-based coordination. Since an ordering scheme can act as a coordination mechanism in a supply chain, our problem is to find some ordering schemes which lead to a stable supply chain (steady streams and steady inventory levels) after a perturbation in the market.

We now present two principles that the two ordering schemes that we propose follow in order to reduce the bullwhip effect. As presented in Figures 5.1(b) and 5.2, each company using one of our ordering schemes places a vector of orders (O, Θ), instead of a unique number O that englobes and hides these two numbers[4]. In (O, Θ), O is the market consumption transmitted from each company to its supplier(s) with the lot-for-lot policy, and Θ is chosen such as $O + \Theta$ represents what the company needs and such as $\Theta = 0$ when O is steady. As O follows a lot-for-lot policy, the bullwhip effect cannot occur in it. Unfortunately, it may occur in Θ, that is, there may be a bullwhip effect in Θ. This is the reason why our (O, Θ) ordering schemes lay on the following two principles, and not only one of them.

First Principle: *The lot-for-lot ordering policy eliminates the bullwhip effect, but does*

---

[4] O like *O*rder and Θ like *T*oken, as these two pieces of information were called in our previous papers. Moreover, O and Θ also have the advantage of looking similar; while they have a very similar meaning: they are both *O*rders.

not manage inventories.

This first principle rules the way of choosing O in (O, Θ). Lot-for-lot orders means that each company orders what is demanded from it; if its client wants 10 products, the company places an order for 10 products. As previously stated in Figure 5.1(a), the problem is the bullwhip effect is eliminated with the lot-for-lot policy, but inventory level is not managed. Therefore, we keep lot-for-lot orders for ruling O, but we add another piece of information Θ to manage inventory level. Note that lot-for-lot orders allow O to share the market consumption information, as illustrated in Figure 5.2.

Second Principle: *Companies should react only once to each market consumption change.*

This second principle rules the way of choosing Θ in (O, Θ). Θ is equal to zero all the time, except when the market consumption changes, in which case companies react to this change by sending non-zero Θ, in order to stabilize their inventory to the initial level. The purpose of Θ is to trigger a product wave from the most upstream company when this company receives this Θ. This product wave will increase, or decrease when $\Theta < 0$, each company's inventory as it travels the supply chain down to the retailers. We later present this global behaviour of the supply chain incurred by our two principles.

We now illustrate these two principles with two ordering schemes for the QWSG.

5.3 Two Collaborative Ordering Schemes for Reducing the Bullwhip Effect Caused by Delays

5.3.1 Ordering Scheme B

The two previous principles are now applied in order to design two ordering schemes for the QWSG. In the first ordering scheme, which is called B in this dissertation[5], companies can know only the market consumption with O. In consequence, a method to follow the second principle, is to set Θ proportional as the variation of O.

Let us now introduce a few notations to present that: i is the considered company,

---

[5] This ordering scheme was also called 'Experiment B' in Moyaux et al. [2004c, 2003a,b], and β in Moyaux et al. [2004b,d].

$i-1$ its unique client, and we assume time is continuous (which is not the case in the QWSG). In this context, Equation 5.1 describes how the company i places orders (O_t^i, Θ_t^i). Note that $\frac{dO_t^{i-1}}{dt} \geq 0$ (respectively $\frac{dO_t^{i-1}}{dt} \leq 0$) represents the forecasted inventory decrease (respectively increase) during the ordering and shipping delays. This quantity has to be overordered (respectively underordered) in order to keep a steady inventory. The constant λ depends on the duration of ordering and shipping delays.

$$(O_t^i, \Theta_t^i) = (O_t^{i-1}, \Theta_t^{i-1} + \lambda \frac{dO_t^{i-1}}{dt}) \tag{5.1}$$

5.3.2 Ordering Scheme D

In the second ordering scheme, which is called D in this dissertation[6], information centralization is used, that is, retailers multicast the market consumption along the whole supply chain. Information sharing with information centralization is much quicker than information sharing with (O, Θ) orders, because the market consumption transmitted in O is as slow as orders, while information centralization is assumed to be instantaneous, reflecting thus the actual market consumption in real-time.

We use information centralization to make our second ordering scheme more efficient than B by setting Θ proportional to the variation of the market consumption: as soon as the market consumption changes, non-zero Θ are sent by all companies. Moreover, companies also base O on the market consumption transmitted by retailers, instead of on incoming O, again in order to react quicker to the market consumption change. If we note again i the considered company, O_t^{market} the market consumption, and if we assume time is continuous, Equation 5.2 presents how company i places its orders (O_t^i, Θ_t^i) when there is information centralization. Again, λ is a constant depending on delays:

$$(O_t^i, \Theta_t^i) = (O_t^{\text{market}}, \Theta_t^{i-1} + \lambda \frac{dO_t^{\text{market}}}{dt}) \tag{5.2}$$

We will see in Section 8.2 that λ is different in our two ordering schemes B and D, and thus, in Equations 5.1 and 5.2. In both ordering schemes, no company is allowed to change O; we assume in this thesis that every agent plays the game, and every company is sincere when it informs others that the market consumption is O. As future work, we could show that no company has an incentive to lie about the actual market consumption in the QWSG. It is clear this is not true in real life, because buyers have

---

[6] This ordering scheme was also also called 'Experiment D' in Moyaux et al. [2004c, 2003a,b], and γ in Moyaux et al. [2004b,d].

less negotiation power when their supplier has a better knowledge of actual buyer's requirements.

The next section presents the supply chain behaviour induced by (O, Θ) orders without information centralization.

5.4 Supply Chain Behavior under the Proposed Information Sharing Scheme

To present the global behaviour of the supply chain with (O, Θ) orders without information centralization, we assume that there is a unique change in market consumption (like in the market consumption called `Step`, later, in Figure 7.1). Precisely, this consumption is as follows: eleven products per week are bought by each market (lumber and paper) in Weeks 1, 2, 3 and 4, and then seventeen products until the end, which is a pattern similar to Figures 5.1 and 5.2.

The goal that (O, Θ) orders must reach, is to bring the supply chain back to a new stable state after the change in the market consumption, and this new state should be the same as the initial state. Since we assume that there is no capacity limit in the supply chain, and in particular, companies can stock as much as they want, the processing of Θ is done as follows. After having traveled the supply chain from one company to another from their starting point (retailer, wholesaler, etc.), the Θ piece of information arrives at the most upstream company (the forest in the forest industry), where it triggers the shipping of a single big batch of products. This batch of products is of sufficient size to fill inventory in each company to a convenient level[7], as this batch of products travels from one company to the next.

The size of this big batch is given by the number Θ, which is the sum of positive and negative company needs added successively by each company when they placed orders (each company transmits to its supplier the Θ coming from its client and adds to it its relative requirements in comparison with the market consumption O).

Concretely, this mechanism divides the supply chain into five successive states. Each of these states occurs at different times along the supply chain, that is, the initial state occurs first by the retailer, next by the wholesaler, ... and the last state occurs finally

---

[7] In this dissertation, we always assume that this convenient level is the inital level, but some objections can be made to this point.

by the wholesaler, and ultimately by the retailer.

In these five steps, we refer to the concept of *stable supply chain*. We define this as a supply chain in which the product stream and inventory levels are steady with time. This means that each week ressembles another one. We can now describe the five states incurred by a supply chain in which every company applies our Scheme B:

1. *Initial state*: Let us assume that the supply chain is stable, and under a steady market consumption. Therefore, the product stream in the supply chain is also steady, by definition of a stable supply chain, and equal to the market requirement. In fact, if we assume processes are reliable[8], each week, companies place the same order as in the previous week, and this order corresponds to the market consumption: placed O are all equal to incoming O and $\Theta = 0$.

2. *Perturbation and reaction of the supply chain*: After the single increase in the market consumption from eleven to seventeen, companies have to change their order from eleven to seventeen products in O, in order to follow the market consumption with the lot-for-lot ordering policy. As there are ordering and shipping delays, companies will still receive eleven items per week for a short period, i.e., the quantity of products they were receiving before the abrupt change in market consumption. It is as if there were an *inertia* in product flow. This leads companies to receive less than what they need during this period, and their inventories decrease. This is the reason why companies ask for more products than O, to reconstitute their inventory; $\Theta \neq 0$ are sent in a unique order.

3. *Wait for the effects of the reaction*: As O has increased since the second state, the flow of products in the supply chain has increased too. As long as this flow increase has not reached a company, this company's inventory decreases, because this company ships seventeen products each week, while it only receives eleven.

4. *First stabilization of the supply chain*: When products corresponding to the new value of O are received, the company's inventory stabilizes. Precisely, this stabilization occurs when the company receives as many products as it ships. Such inventory stabilization is in fact too low: it will be so until products corresponding to Θ sent by the most upstream company arrive, that is until the next state.

5. *Second stabilization of the supply chain*: The supply chain does not remain in the fourth state, because Θ sent in the second state have triggered a larger batch

---

[8] We add this assumption in order to avoid the only cause of the bullwhip effect in Figure 3.2 that may apply in this scenario, that is, variabilities due to company processes presented in Section 3.2. This cause is due to the uncertainties in processes that lead to uncertainties in order placement.

of products to arrive at the company, which brings inventory to a desired level. In fact, when the Θ sent in the second state have arrived at the most upstream company (the Forest in the QWSG), a big batch of products was sent down the supply chain to increase all inventories. Finally, this last state is the same as the fourth one, except that inventories are increased to the desired level.

From these five states, we can see that inventory variations last a longer time for retailers than for the most upstream supplier(s), because retailers enter state 2 first and state 5 last. Indeed, inventory stabilization comes from upstream suppliers and travels down to suppliers. In general, it is said that the bullwhip effect disturbs more upstream suppliers than retailers. Here, this situation is less clear: *retailers have longer inventory fluctuation*, and possibly longer stockouts, than upstream suppliers. Nevertheless, this inventory fluctuation is still bigger upstream suppliers, as is usual with the bullwhip effect. Finally, these five steps are incurred by the two proposed ordering schemes, but there is a difference in the inventory variation. In fact, the inventory variation is reduced in duration and in amplitude when information centralization is used.

Note that these five steps are an idealized presentation of the supply chain behaviour, because they occur when there are no stockouts. Stockouts mean that the third and the fourth steps are mixed by a company, when its supplier incurs stockouts. For example, the company orders for and receives eleven products during the four weeks in the first step. Then, the company orders for seventeen products but receives only eleven products during four weeks in the second step, because of the ordering and shipping delays. In consequence, its inventory decreases. The company eventually receives the seventeen ordered products, which means that it enters the third step. But the company receives these seventeen items only during two weeks, because its supplier is currently in the second step (the five steps do not occur at the same time by every company): a stockout occurs next by this supplier, and thus, less than seventeen products are received, and the company comes back into the second step. This phenomenon is repeated when more upstream suppliers are also considered, because stockouts may also occur later by these suppliers.

Finally, these five steps are accurate under our first proposed ordering scheme (called B), and some tiny adaptations are required to describe the supply chain behaviour under our second ordering scheme D, but we do not insist thereon for the sake of simplicity, because these adaptations are only details that are related to the use of information centralization.

The next section illustrates the use in a multi-item company of our two ordering schemes based on (O, Θ) orders.

5.5 Illustration

As an illustration, consider a company buying three types of products 1, 2 and 3, and selling two types of products 4 and 5. We note $OP = (op_1, op_2, op_3)$ the quantities of ordered products, $PS = (ps_4, ps_5)$ the quantities of products to ship. The company needs one unit of op_1 and two units of op_3 to build one unit of ps_4, and two units of op_2 and one unit of op_3 to build one unit of ps_5; we write these two relations in a matrix M such as $OP = M.PS^T$:

$$M = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix} \tag{5.3}$$

Assume that our company has two clients $Client1$ and $Client2$. The rest of the calculations can represent our two ordering schemes, except that the meaning of components of PS are different:

- *Our first ordering scheme* (Scheme B), in which information centralization is not used:
  - $Client1$ orders for $(6, 1)$ units of ps_4 (that is, the ps_4 market consumption was $O_{ps_4}^{Client1} = 6$ anytime in the past, and $Client1$ currently needs one another product to stabilize its inventory, because $\Theta_{ps_4}^{Client1} = 1$) and $(5, 0)$ units of ps_5.
  - $Client2$ orders for $(7, -2)$ units of ps_4 ($Client2$ does not want 7 items even if they are consumed by the market, and therefore it cancels 2 of them, which is written $\Theta_{ps_4}^{Client2} = -2$) and $(2, 0)$ units of ps_5.
- *Our second ordering scheme* (Scheme D), in which information centralization is used:
  - 6 units of ps_4 are currently consumed by the $Client1$'s market, $Client1$ has just ordered for $\Theta_{ps_4}^{Client1} = 1$ in addition to the market consumption, 5 units of ps_5 are consumed by the $Client1$'s market, and $Client1$ has just ordered for $\Theta_{ps_5}^{client1} = 0$.
  - 7 units of ps_4 are currently consumed by the $Client2$'s market, $Client2$ needs 2 units in less than this consumption ($\Theta_{ps_4}^{Client2} = -2$), $Client2$'s market consumes 2 units of ps_5, and $Client2$ would like to get these two products ($\Theta_{ps_5}^{Client2} = 0$).

In both cases, we note the demand PS:

$$\begin{aligned} PS = & \quad PS^{Client1} & + & \quad PS^{Client2} \\ = & \begin{pmatrix} O^{client1}_{PS_4} & \Theta^{client1}_{PS_4} \\ O^{client1}_{PS_5} & \Theta^{client1}_{PS_5} \end{pmatrix} & + & \begin{pmatrix} O^{client2}_{PS_4} & \Theta^{client2}_{PS_4} \\ O^{client2}_{PS_5} & \Theta^{client2}_{PS_5} \end{pmatrix} \\ = & \begin{pmatrix} 6 & 1 \\ 5 & 0 \end{pmatrix} & + & \begin{pmatrix} 7 & -2 \\ 2 & 0 \end{pmatrix} & = \begin{pmatrix} 13 & -1 \\ 7 & 0 \end{pmatrix} \end{aligned} \tag{5.4}$$

Our company has twelve units (= 13 − 1) of ps_4 and seven units (= 7 + 0) of ps_5 to ship. To do so, it has to order the quantity OP, where OP is a vector of (O, Θ) orders for each of the products to order 1, 2 and 3:

$$OP = M.PS^T = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix} . \begin{pmatrix} 13 & 7 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 13 & 7 \\ -2 & 0 \\ 25 & 14 \end{pmatrix} = \begin{pmatrix} op_1 & \Theta p_1 \\ op_2 & \Theta p_2 \\ op_3 & \Theta p_3 \end{pmatrix} \tag{5.5}$$

The line r in the OP matrix gives the quantity of item op_r to order. The left column of this matrix represents the market consumption O that must not be changed and the right column is the aggregation of the quantities Θ wanted by both clients in more or in less in comparison with the left column. Note that op_2 is negative, and that $op_2 + \Theta p_2$ is also negative. We can interpret this as a cancellation of past or future orders for these types of products.

Basically, the company should place the order OP, but it is allowed to change the right column if it needs to. The company is only allowed to change the line in OP when the left column (i.e., the indication of O) is different from the previous week, in order to respect our second principle, i.e., Θ can be modified only when O fluctuates. To illustrate this, we now assume the company wants and is allowed to produce two items of PS_1 and one item of PS_2 more than its demand. Its inventories are thus going to increase. The company does so, because it needs fifteen units (= 13+2) of OP_1, sixteen units (= 14+2) of OP_2 and thirty-six units (= 33+3) of OP_3, for example, because the company has to produce these additional products in order to use its whole capacity of production. To change the right column, that is, to change the Θ part of the order, the company calculates how many additional OP_1, OP_2 and OP_3 it has to order. We call this additional quantity R, which corresponds to $\lambda.dO/dt$ in Equations 5.1 and 5.2. We should note that $\begin{pmatrix} 2 \\ 1 \end{pmatrix}$ corresponds to two items of ps_4 and one item of ps_5.

$$R = M. \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 2 & 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 5 \end{pmatrix} \tag{5.6}$$

To place an order, a column of zeroes is added to this vector, to show these numbers do not represent the market consumption O but company requirements Θ. Finally, the company places the order OP':

$$OP' = OP + \begin{pmatrix} 0 & \\ 0 & R \\ 0 & \end{pmatrix} = \begin{pmatrix} 13 & 7 \\ -2 & 0 \\ 25 & 14 \end{pmatrix} + \begin{pmatrix} 0 & 2 \\ 0 & 2 \\ 0 & 5 \end{pmatrix} = \begin{pmatrix} 13 & 9 \\ -2 & 2 \\ 25 & 19 \end{pmatrix} \tag{5.7}$$

This example exposes the possibility of using this ordering scheme in more complex settings where more than one customer are purchasing and many products are needed to assemble. The calculation of Θ is however much more complex in that setting. In fact, the supplier is often counting on the positive aspects of risk pooling (i.e., the combination of the uncertainty over individual products in order to have a more steady level of the combination. For example, if you produce A_1 and A_2 with B, the inventory level of B fluctuates less than the fluctuation of A_1 plus the fluctation of A_2) for diminishing its need for inventory. As a result, inventory management is not that easy.

The rest of this thesis studies the implications of our two ordering schemes for the supply chain. Similarly, we now outline the implications for multi-agent systems in general.

5.6 Consequences for Multi-Agent Systems

We first describe the consequences of the theory presented in this chapter for information systems modelled as multi-agent systems. Concerning this point, let us recall the growing importance of multi-agent systems in industry. After that, we underline a much more important consequence, because this consequence applies to *any* multi-agent system.

5.6.1 Consequences for Information Systems Modelled as Multi-Agent Systems

When a supply chain is modelled as a multi-agent system, the direct consequence of sharing the market consumption information with (O, Θ) orders is an increase of the load of the network supporting inter-agent communications. In the example in Section 5.5, the quantity of data in placed orders doubles, that is, orders are the six numbers

$13, 9, -2, 2, 25$ and 19 in Equation 5.7, instead of the six numbers $23(= 13 + 9), 0(= -2 + 2)$ and $44(= 25 + 19)$.

But in real life, the quantity of transmitted information more than doubles, because we can assume, that the market consumption is transmitted more often than only when orders are placed. That is, companies do not place orders often, in order, for example, to profit from economies of scales, while the transmission of the market consumption should be much more frequent than these orders. In fact, if this information is shared more often, companies have a better understanding of the market demand.

Another consequence of sharing information for a supply chain modeled as a multi-agent system is the adaptation of the message format. For example, (O, Θ) orders use order messages with two fields per item instead of only one.

5.6.2 Consequences for any Multi-Agent Systems

More generally, the cause of the bullwhip effect presented in this chapter may also incur stream fluctuations in any multi-agent system. To see this link between supply chains and multi-agent systems, Table 5.1 presents a translation dictionary. Indeed, the definition of multi-agent systems in Subsection 2.1.1 is sufficiently wide to encompass supply chains, because this definition does not assume that agents are softwares, but they can also be people. As a result, the following dictionnary only reflects the specificities (and not the differences) of supply chains.

1. Of course, the word “company” refers to “agent”. . .
2. . . . and “bullwhip effect” comes from “stream fluctation” or “system instability”.

In this context, the proposed cause of fluctations, i.e., delays, have the following equivalents in multi-agent systems:

3. “Ordering delays” in supply chains correspond to delays on the network that links agents, when these agents run on different computers. Such delays are referred as “network latency” in Table 5.1. We can feel this latency, when we browse on the Internet.

   But even without this latency, delays are also incurred by agents’ message buffers. In fact, agents cannot always process incoming messages as soon as they are

	Supply chain	**Multi-agent system**
1.	Company	Agent
2.	Bullwhip effect	Stream fluctuation/instability
3.	Ordering delay	Network latency
4.	Shipping delay	Request execution time
5.	Market consumption	Requirement of the multi-agent system
6.	Inventory	Resource storage
7.	Ordering scheme	Agent's behaviour
8.	Lot-for-lot ordering policy	Behaviour in which the considered agent acts exactly as another agent

Table 5.1: Instantiation of multi-agent vocabulary into supply chain vocabulary.

received, and thus, incoming messages first arrive in a buffer, i.e., a mail box, in which they wait until the agent picks them up. This delay is not presented in Table 5.1, because in this dissertation we do not consider its equivalent in the world of supply chains, in which they are orders waiting by the company before their processing, e.g., in a mail box. In addition to ordering delays and waiting in mail box, we consider another delay:

4. "Shipping delays" in supply chains are similar to the execution time of the requests addressed to agents. That is, when an agent S (supplier) is requested to perform an action by an agent C (client), if agent S agrees to achieve this action, a certain time has to be spent by C until the action has been completely performed. It is similar to the shipping delay that a client C has indure to receive its products.

5. The "market consumption" resembles the "requirement of the multi-agent system", because both are the output, that sets the point at which the supply chain/system has to be.

6. The "inventory" of a company has an equivalent in agents, which depends on the application of this agent. We give the general term of "resource storage" to this equivalent. Of course, when this agent models a company, the stored ressource may be products, and we will see another example soon.

7. The "ordering scheme" rules our company-agent's behaviour, and thus, such a scheme implements the "agent's behaviour".

8. The "lot-for-lot ordering policy" is one particular ordering scheme in which a company orders exactly its incoming demand. More generally, it means that the company-agent performs exactly the same action as another agent, like a young child repeating what his elder brother says to bother him.

Obviously, the way that delays incur fluctuations in a multi-agent system depends on the considered application. We can only note the similarity between the product stream in a supply chain and the flow of tasks in a multi-agent system, and between the ordering stream in a supply chain and the flow of requests in a multi-agent system. With these two similarities, we can infer that the *flow of tasks* in the multi-agent system *may fluctuate*, when there are changes in the requested output of the multi-agent system. More precisely, when this requested output changes, fluctuations appear in the flow of tasks carried out by agents, when these agents act so that the multi-agent system output is correct. As previously stated, such fluctuations reduce the efficiency of the multi-agent system in carrying out its functions.

To illustrate both this point and what can be a "resource storage", let us consider the case of intelligent highway systems outlined in Subsection 2.2.6 in which some researchers [Hallé et al., 2003] are working on how to design platoons of cars. Each vehicle is modelled as an agent, and therefore, the platoon corresponds to a coalition or to a team of agents travelling toward a common destination. Except the front vehicle of a platoon, every other vehicle has to follow the preceding vehicle. The question is how to design agents' behaviour in order to carry out this functionality. In this scenario, inter-vehicle distance is one of the possible stored resources. When we give each vehicle-agent the equivalent of the Lot-for-Lot policy, i.e., when we make each vehicle have the same speed as its preceding car, curves in Figure 5.1(a) explain why the stored resource, i.e., the inter-vehicle distance, increases when the platoon accelerates, and decreases when the platoon decelerates. In other words, the inter-vehicle distance may fluctuate, leading to a "slinky effect" in the platoon [Ioannou and Chien, 1993; Chien and Ioannou, 1992; Sheikholeslam and Desoer, 1990]. This effect corresponds to variations in the resource storage.

If we want the inter-vehicle distance to be constant, each vehicle-agent must be given another behaviour. But this behaviour has to be designed carefully to avoid distance fluctuations, and in particular a translation of (O, Θ) orders to this problem of platoons. In this case, the front vehicle, which imposes the speed of the whole platoon, would transmit its velocity to the rest of the platoon. Other vehicles would accelerate or decelerate more or less than the first vehicles in order to keep a steady inter-vehicle distance[9].

Of course, this platoon problem is more complex in real-life than what has just been described, for example, because of the lack of accuracy in speed measurement

---

[9] Similarly to supply chains, in which we aim at having eventually a steady inventory in this dissertation, we can also object when we say that inter-vehicle distance should be kept steady, because the security distance should increase with speed.

by each vehicle, and because vehicle-agents' behaviour is more complex than a simple transpostion of (O, Θ) orders. Similarly, we have neglected a part of the supply chain complexity to propose (O, Θ) orders, in particular, because we do not consider all known causes of the bullwhip effect presented in Table 3.2, such as demand forecast updating, order batching, price fluctuation. . ..

5.7 Conclusion

This section has proposed to consider ordering and shipping delays as a new cause of the bullwhip effect. Up to now, delays were only seen as a factor aggravating another cause of the bullwhip effect, called the demand signal processing [Simchi-Levi et al., 2000; Chen et al., 2000; Lee et al., 1997a,b; Ryan, 1997]. Notice that demand signal processing can also cause the bullwhip effect, because forecasts made by all companies, during information processing, induce errors that add up in the supply chain, and the longer forecasts are, the greater these errors are. Therefore, the longer delays are, the bigger the bullwhip effect incurred by this cause is, because companies forecast on a longer time horizon.

In order to address the bullwhip effect, we have proposed two principles to design ordering schemes, in which this effect induced by delays would be minimized. These two principles are that the lot-for-lot ordering policy eliminates the bullwhip effect, but does not manage inventories, and companies should react only once to each market consumption change. This second principle makes that companies should collaborate, because sharing the market consumption information requires collaboration.

Then, two ordering schemes were proposed according to these two principles. The supply chain behaviour under such schemes was also outlined, and an illustration was proposed for a multi-item scenario.

Finally, we outlined the implication of this chapter for multi-agent systems. Of course, the information sharing that we propose has consequences for information systems in supply chains that are modelled as multi-agent systems, because agents have to manage this additional information. But a further implication concerns multi-agent systems in general. To see this point, we proposed a short dictionary to translate vocabulary in supply chains into vocabulary in multi-agent systems. This dictionary shows that the bullwhip effect is referred to as stream fluctuations in multi-agent systems, and that such fluctuations can also be induced by delays in multi-agent systems. In this context, our two principles can also be translated to any multi-agent system to reduce these fluctuations, and thus, to increase the efficiency of the considered system.

The next chapter presents the simulation model used to validate the efficiency of our two principles in reducing the bullwhip effect.

Chapter 6

Multi-Agent Simulation of a Québec Forest Supply Chain

The previous chapter has presented how delays in distributed systems induce stream fluctuations from a conceptual viewpoint. To this end, two principles to reduce the fluctuations induced by this cause have been presented. In the remainder of this dissertation, we study the efficiency of the two proposed principles on supply chains modelled as multi-agent systems. In this context, we verify whether our two principles reduce stream fluctuations (called bullwhip effect, since we consider supply chains) in an agent-based simulation of a supply chain.

In other words, we verify in the following chapters, whether our solution to stream fluctuations *coordinates* the way that company-agents place orders, so that the bullwhip effect is reduced. We now need a simulator to validate these principles, by verifying if our two ordering schemes actually reduce the bullwhip effect. The goal of this chapter is to introduce this simulator.

Since, in this entire dissertation, we use company and supply chain models that are based on the Québec Wood Supply Game (QWSG), we introduce this game in detail first. To this end, we show that it is an adaptation of the Beer Game to the Québec wood industry. Notice that the Beer Game has been designed to teach the dynamics in supply chains, and specifically, the bullwhip effect. These two games are first introduced in Section 6.1.

We have implemented the QWSG in a spreadsheet program. This implementation closely simulates the QWSG, and is described with time-dependent equations in Section 6.2.

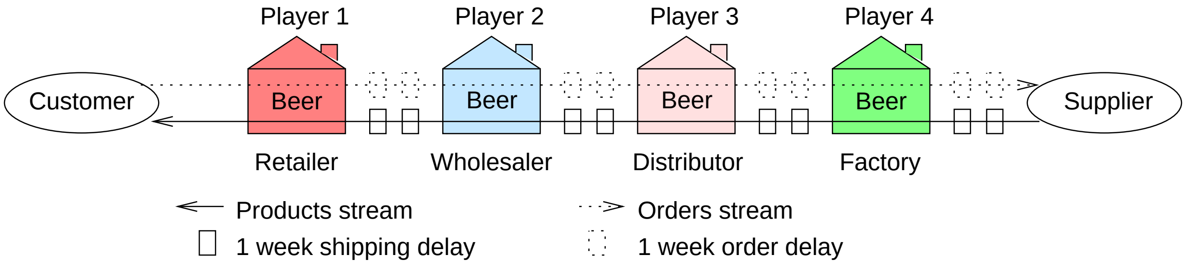


Figure 6.1: Supply chain in the Beer Game [Sterman, 1989].

Finally, we have designed and implemented a more realistic simulator with the agent-oriented language `JACK™`. `JACK™` and our second simulator is presented in Section 6.3.

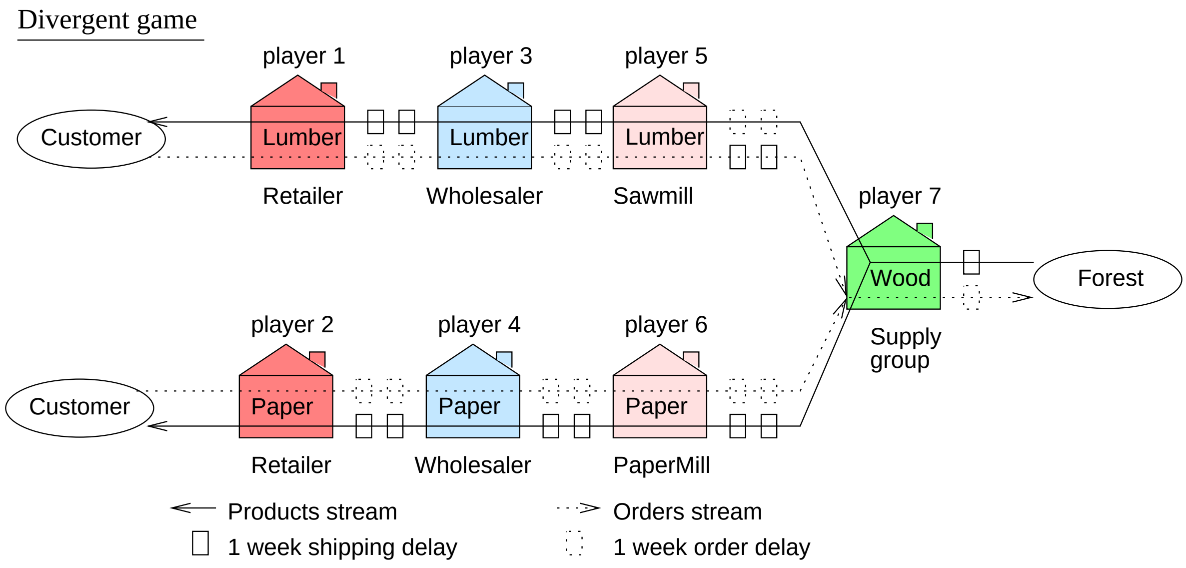
6.1 The Québec Wood Supply Game (QWSG)

6.1.1 Pedigree of the QWSG

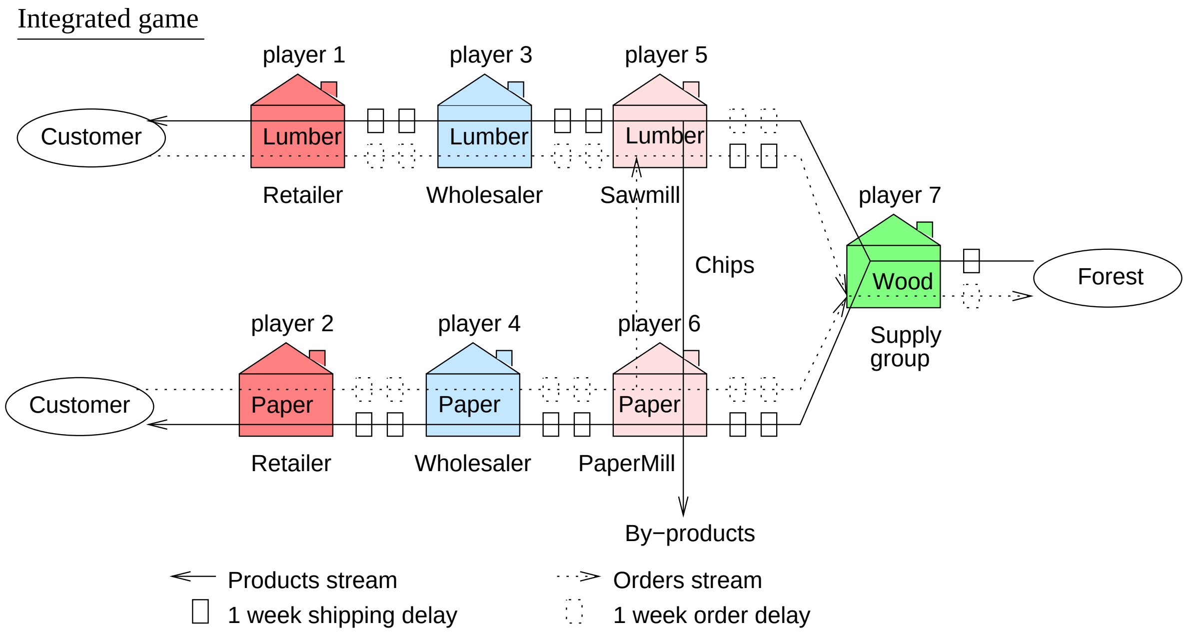
The framework used in this thesis is the "Québec Wood Supply Game" (QWSG), illustrated in Figure 6.3. The QWSG was derived from two board-games, called "Integrated and Divergent Wood Supply Games" [Fjeld, 2001; Haartveit and Fjeld, 2002], and presented in Figures 6.2. These latter two games were themselves adapted from the "Beer Game", illustrated in Figure 6.1. We now outline the relations between these four games.

The basis of these four games is the Beer Game from Sterman [1989]. It is a classroom exercise, that simulates the material and information flows in a production-distribution system, as illustrated in Figure 6.1. It was designed to make players aware of supply chain dynamics, and specifically the bullwhip effect. As a consequence, most *hypotheses* made in this work are imposed by this framework, e.g., discrete time, model of companies, method of cost evaluation (even if we adapt this method to parameters reflecting the Québec wood industry in Section 8.2)... Concerning hypotheses, the main difference between the four games above lies in the supply chain model, which describes how many companies are simulated, which company sells to another company, and which company buys from another.

In the three other games, the supply chain model is modified, so that specificities of



(a) ***Divergent game***



(b) ***Integrated game***

Figure 6.2: Forest supply chain in the two original Wood Supply Games [Fjeld, 2001; Haartveit and Fjeld, 2002].

some industries are reflected. For example, the wood industry has the particularity of dealing with wood, which is an organic material needing care (e.g., it can molder). In our

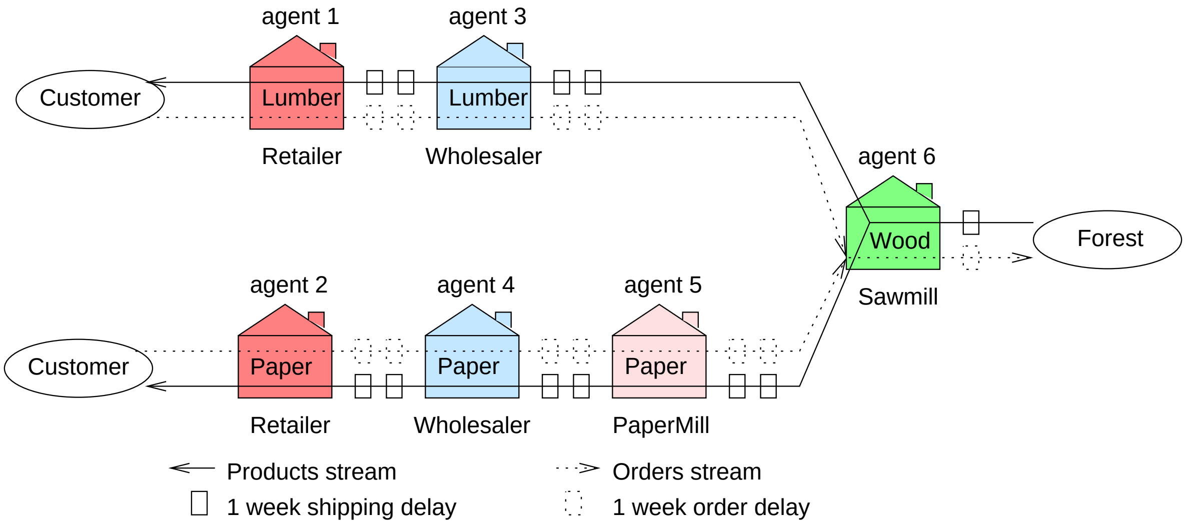


Figure 6.3: Forest supply chain in the Québec Wood Supply Game (QWSG), i.e., the one used in this dissertation.

special case of supply chain management, the relevant particularity of the wood industry is the diverging material profile, which is due to the broad variety of products (papers, books, paperboard boxes, furniture, buildings...) manufactured from a few types of raw materials (wood) [Fjeld, 2001]. Since, in this dissertation, we are interested in the Québec wood industry, we have to consider a supply chain with a divergent product flow.

The two first modifications of the Beer Game take this diverging material profile of the wood industry into account. These modifications are called Wood Supply Games [Fjeld, 2001; Haartveit and Fjeld, 2002], and introduced divergent product flows in order to increase their relevance to the North European forest sector. The difference between these two Wood Supply Games is the product stream from the Sawmill to the PaperMill in the Integrated Game in Figure 6.2(b), that does not exist in the Divergent Game in Figure 6.2(a).

Finally, our team in FOR@C [2004][1] has adapted these two games to the Québec forest sector in a game called the Québec Wood Supply Game (QWSG), displayed in Figure 6.3. The difference between the two original Wood Supply Games and this QWSG is in the supply chain model, because the length of the lumber and paper chains was the same in the original games in Figures 6.2, while it is different in the QWSG in Figure 6.3. This difference is due to differences between North European and Québec forest industries: the PaperMill is separated from the Sawmill in Québec, conversely to

---

[1] I would like to thank Jean-Marc Frayret and Philippe Marier for their work on the QWSG.

the North of Europe.

The company and supply chain models in the QWSG are now described. Precisely, we introduce the board-game version of the QWSG, which will next be the basis of our two simulators.

6.1.2 Description of the QWSG

Introduction to the QWSG

The QWSG is based on Fjeld's two Wood Supply Games [Fjeld, 2001; Haartveit and Fjeld, 2002], depicted in Figure 6.2, but the QWSG requires a different number of players, because it models a different supply chain model, as illustrated in Figure 6.3. Thus,

- two players have the role of the two retailers, sell to their respective market, and buy from their respective wholesaler,
- one of the two wholesalers, the PaperWholesaler, buys from the PaperMill, which itself buys from the Sawmill,
- the other wholesaler, the LumberWholesaler, buys directly from the Sawmill,
- the Sawmill places orders to the Forest by aggregating the orders from the LumberWholesaler and the PaperMill.

Each player (who will be replaced by intelligent agents in our two simulators) takes the role of one of these six companies, and places orders to her/his supplier, so that the inventory levels and stockouts are minimum for the whole supply chain. Indeed, the players are not opponents, but form a team.

There is a two-week delay to transmit the demand. When a supplier receives an order, i.e., two weeks after its client has placed it, this player has to fulfill it when its inventory is enough, or else all products in inventory are shipped, and unsatisfied orders are memorized as *backorders*. Backorders represent products to ship as soon as possible, and are noted as negative inventories. The shipping introduces another two-week delay.

Progression of the QWSG

The game is played by turns, where each turn represents a week in reality, and is played in five days. These five days are played in parallel by each player, according to Figure 6.3:

- *Day 1:* The player receives her/his inventory (these products were sent two weeks earlier by her/his supplier, because there is a two-week shipping delay), and advances the shipping between its suppliers and itself, in order to represent the second week of shipping delay (see lines 1 and 2 in Algorithm 6.1).
- *Day 2:* The player looks at her/his incoming order, and tries to fulfill them. This fulfilment is performed by moving products in the first week of shipping delay. This corresponds to the third line in Algorithm 6.1. For the sake of simplicity, Algorithm 6.1 does not mention the two following conditions:
    - If the player has backorders from past weeks, s/he tries to fill those as well;
    - If the player does not have enough inventory, s/he ships as much as s/he can, and adds the rest to her/his backorders.
- *Day 3:* The player records her/his inventory or backorders, as reflected by the fourth line in Algorithm 6.1.
- *Day 4:* The player advances order slips (see line 5 in Algorithm 6.1) to simulate a week of ordering delays.
- *Day 5:* The player places an order with her/his supplier(s), and records this order, which is described in lines 6, 7 and 8 in Algorithm 6.1.

After the player has finished the fifth day, the game continues with a new week, that begins in the first day, and so on.

To decide in line 6 of Algorithm 6.1 what the order to place is, the player compares her/his incoming orders with her/his inventory/backorder level. Order placement is the only decision made by players, everything else is performed mechanically in the five previous days. In this thesis, the agents, that replace human players, make this decision by applying one of seven ordering schemes called A, A', A'', B, B', C and D, introduced in Figure 6.1.2, and described now.

Ordering Scheme A: Companies can only base their orders on the incoming demand and on their inventory level. Each order is a unique number X calculated by

Scheme	O	Θ	Orders	Information centralization
A	incoming O minus inventory variation, when positive	zero	$X(=O+\Theta)$	no
A’	incoming O plus λ times order variation	zero	$X(=O+\Theta)$	no
A”	S minus inventory level when inventory is lower than s	zero	$X(=O+\Theta)$	no
B	incoming O	incoming Θ plus λ times order variation	(O,Θ)	no
B’	incoming O	incoming Θ minus inventory variation, when positive	(O,Θ)	no
C	customer consumption minus inventory variation	zero	$X(=O+\Theta)$	yes
D	customer consumption	incoming Θ plus λ times customer consumption variation	(O,Θ)	yes

Table 6.1: Experimented ordering schemes.

Algorithm 6.1 playAweek(incomingOrder, incomingShipping)

```
returns outgoingOrder, outgoingShipping
local variables: inventory, placedOrder

Add incomingShipping to inventory
Move outgoingShipping by our supplier into our incomingShipping
Move the quantity incomingOrder from inventory into outgoingShipping
Record inventory
Move outgoingOrder by our client into incomingOrder
Choose placedOrder
Put placedOrder into outgoingOrder
Record placedOrder
```

adding the inventory level variation to the client's order; when this value is negative, nothing is ordered.

For example, when the incoming order is 11, and the inventory level has decreased from 2 to 1 unit since the previous week, the company places an order for $X = 11 + (2 - 1)$ products.

Ordering Scheme A': Like Experiment A, except that each order is calculated by subtracting λ times the order variation of the client's order, where λ has the same value as in Scheme B. In our experiments, this calculation always gives a positive result. The main difference between Schemes A and A' is that A is based on order and product flows, while Experiment A' is only based on order flow.

For example, when the incoming order was for 11 products last week, and is currently for 17 products, the company places an order for $X = 17 + \lambda * (17 - 11)$ products.

Ordering Scheme A": It is a classic (s, S) ordering policy, that is, when inventory I is lower than s, the company orders for $S - I$ products, so that the inventory increases up to S.

Ordering Scheme B: This is our first ordering scheme proposed to reduce the bullwhip effect in Section 5.3. Instead of a single order X, companies now place vectors of orders (O, Θ), in order to share the market consumption information. These two numbers O and Θ are chosen according to our two principles introduced in Chapter 5. These two principles force companies to use the lot-for-lot ordering policy in O, and use Θ to manage their inventory, according to Equation 5.1. That is, client's Θ is transmitted to the supplier, and λ times the order variation is added to balance the inventory variation induced by the change in O. The choice of λ is explained in Subsection B.3.

For example, when the incoming order was for 11 products previous week, and is currently for 17 products, the company places an order for $(O, \Theta) = (17, \lambda * (17 - 11))$ products.

Ordering Scheme B′: it is similar to Scheme B, except that Θ now depends on inventory level. There is the same relation between Schemes B and B′, as between Schemes A and A′, that is, A′ and B′ only take the demand into account, while A and B also take the inventory level into account, which makes a stronger link between the ordering and the shipping streams.

For example, when the incoming order is currently for 17 products, and the inventory levels has decreased from 2 to 1 units since the previous week, the company places an order for $(O, \Theta) = (17, \lambda * (2 - 1))$ products.

Ordering Scheme C: this scheme is similar to Scheme A, except that information centralization is now used. In comparison with A, companies now base their orders on the actual market consumption, instead of on client's orders. Note that placed orders do not depend directly on incoming orders.

For example, when the market consumption is 11, and the inventory level has decreased from 2 to 1 units since the previous week, the company places an order for $(O, \Theta) = (11 + (2 - 1), 0)$ products. This is the same example as for Scheme A, except that "market consumption" replaces "incoming order".

Ordering Scheme D: this scheme is similar to Scheme B, except that information centralization is now used in addition to (O, Θ) orders. Companies now base their orders on the actual market consumption, instead of on client's orders. The choice of the constant λ is explained in Subsection B.4. Scheme D is an improvement on Scheme B, because information centralization speeds up the broadcast of market consumption information. The behaviour of D is outlined by Equation 5.2. Like C, D places orders that do not depend directly on incoming orders.

For example, when the market consumption was for 11 products the previous week and is currently for 17 products, the company places an order for $(O, \Theta) = (17, \lambda * (17 - 11))$ products. This is the same example as for Scheme B, except that "market consumption" replaces "incoming order".

The relations between these seven ordering schemes are as follows:

- As indicated in Figure 6.1.2, only companies using Schemes C and D know in real-time the market consumption (i.e., information centralization is applied), and only companies using Schemes B, B′ and D have (O, Θ) orders.

- Schemes B and D have been described in section 5.2, and Scheme D is an improvement on B, since it applies information centralization in addition to our two principles.
- Schemes A′ and B′ are two schemes derived from A and B. Their difference is to take the inventory level into account, while A and B do not do so. As a consequence, schemes A′ and B′ create a much stronger link between the ordering and the product stream than A and B. Scheme A″ has no relation to the six other schemes. Though, A″ is interesting, because it is based on the classic (s, S) ordering policy, in which there is no information sharing with (O, Θ) orders or information centralization.

Particularities of the Sawmill

Each position is played in the same way, except the Sawmill, since this position receives two orders (one from the LumberWholesaler, another from the PaperMill), that have to be aggregated when placing an order to the Forest. The Sawmill can evaluate its order by basing it, either on the lumber demand, or on the paper demand. In this thesis, the Sawmill places an order equal to the mean of these two possible orders, except in Subsection C.4 where the Sawmill places an order equal either to the maximum of these two possible orders, or to only one of these two possible orders, in order to study the impacts of this aggregation.

Moreover, the Sawmill receives one type of product and each unit of this product generates two units: a lumber and a paper unit. That is, each incoming unit is split in two: one piece goes to the Sawmill's lumber inventory, the other goes to its paper inventory.

We now present our first simulator that reflects the QWSG.

6.2 Implementation of the QWSG: The Spreadsheet Approach

We have programmed two simulations based on the QWSG. The first simulation was implemented in a spreadsheet program, and the other one, presented in Section 6.3, was programmed as a multi-agent system with the agent-oriented language JACK™. We present the first simulator in this section. This implementation has also been presented in [Moyaux et al., 2004a, 2003a].

The use of a spreadsheet program for the first simulator has two main advantages. The first advantage is the availability of many generic tools, in particular to optimize parameters, to analyze simulation outcomes, or to automatize some tasks. All these tools may also be included to our second simulation, but this is an additional task of programmation, while these tools are provided with the spreadsheet program. The second advantage is the speed, not only to design and implement the simulator and the automatization of some tasks, but also to run the simulations. Since we run a lot of simulations, the simulation speed is crucial here. To illustrate the difference of speed, a fifty week simulation takes less than a second on our first simulator, while our second simulator takes around 20 seconds. Of course, the agent-based simulator also has many advantages, but we present these arguments later.

To present our spreadsheet implementation of the QWSG, notations are first introduced, follow generic equations, then scheme-specific equations. We can note that the setting of parameters and initial conditions will be presented in Chapters 7 and 8, since such a setting depends on the considered experiment.

6.2.1 Notations

As previously stated, orders may have up to two dimensions, called O and Θ. O placed in week w by company i is noted Op^i_w, and the corresponding Θ is noted Θp^i_w. The way to calculate these two numbers depends on the ordering scheme, and will be presented in Subsection A.3. The rest of the variables required to model the company i, whose behaviour will be presented in Subsection 6.2.2, are as follows. These notations apply to most of this document, except Subsection 2.1.2 and 8.2.2 in which we use notations for the Economic Order Quantity.

To^i_w = company i's *o*utgoing *T*ransport in week w.

Too^i_w = company i's *o*utgoing *T*ransport in week w corresp. to current O.

Tob^i_w = company i's *o*utgoing *T*ransport in week w corresp. to backordered O.

$To\Theta^i_w$ = company i's *o*utgoing *T*ransport in w corresp. to backordered Θ.

Ti^i_w = company i's *i*ncoming *T*ransport in week w.

I_w^i = company i's Inventory in week w.

Oo_w^i = company i's $o$utgoing Orders O in week w.

Oi_w^i = company i's $i$ncoming Orders O in week w.

Ob_w^i = company i's $b$ackordered O in week w.

Θo_w^i = company i's $o$utgoing Θ in week w.

Θi_w^i = company i's $i$ncoming Θ in week w.

Θb_w^i = company i's $b$ackordered Θ in week w.

$D_w^{\text{lumber}} = Oi_w^1$ = lumber market consumption in Week w.

$D_w^{\text{paper}} = Oi_w^2$ = paper market consumption in Week w.

Except the inventory I and the two market consumptions D, the first letter in the name of these variables indicates if the considered variable is for the shipping stream (T for transportation) or for the ordering stream (O or Θ), and the second letter indicates if it is an $i$ncoming, $o$utgoing, $p$laced or $b$ackordered value of this stream. For the sake of simplicity, the quantity of products to ship To is plit into three parts: Too and Tob for orders O, and $To\Theta$ for orders Θ.

We now outline the equations describing one company in our simulator. We begin with the equations that are used by any company, and we finish with the equations that depend on the ordering scheme used by the considered company.

6.2.2 Simulation Model

The equations that are outlined now, implement the QWSG with the use of two sets of equations[2]:

- *Scheme-independent equations:* This first set of equations implements the animation of product and ordering streams in the QWSG. In other words, these equations represent what we have called earlier the "mechanical" part of the QWSG, that is, the actions that players have to perform without making decisions.

---

[2] The simulation begins in Week $w = 1$.

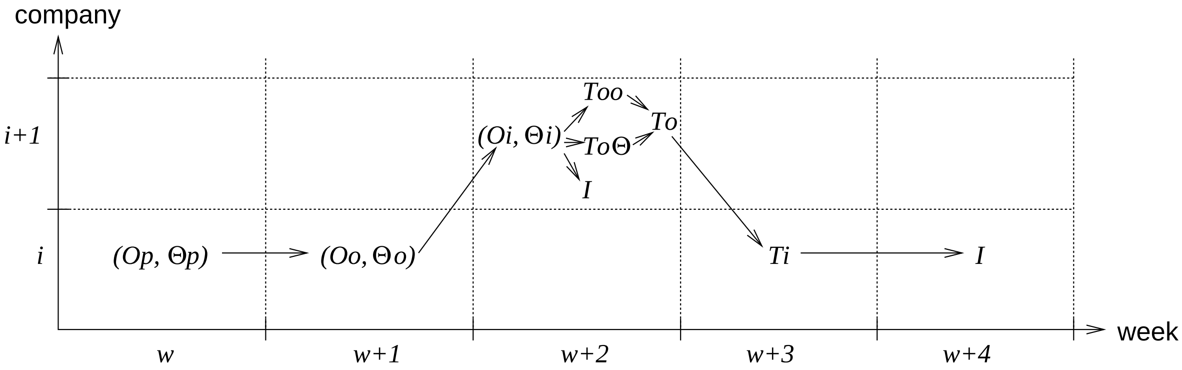


Figure 6.4: Some relations between variables.

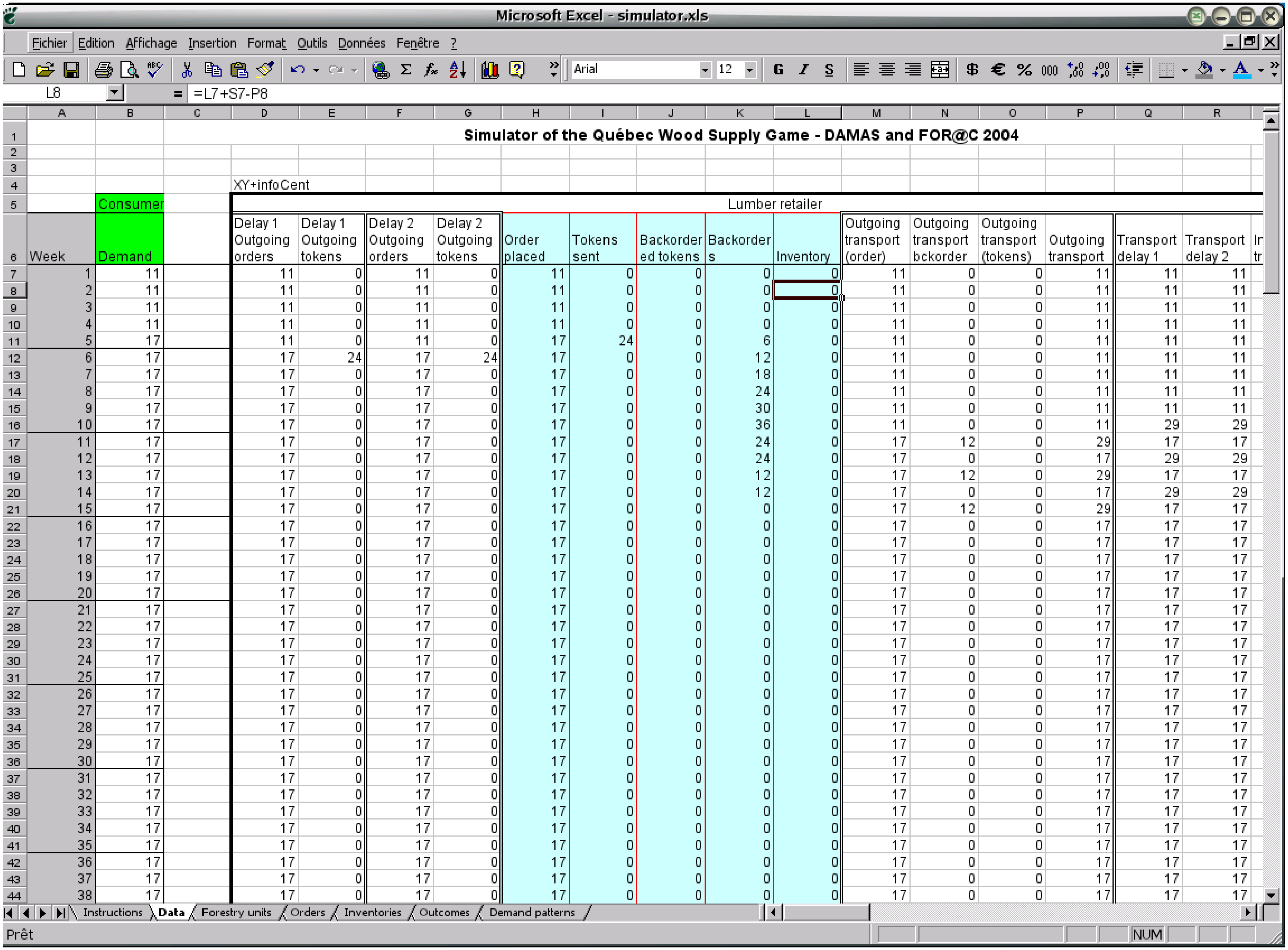
For example, Figure 6.4 presents some of the relations between the above variables, that are implemented by these equations. This figure illustrates that an order $(Op, \Theta p)$ is placed in Week w by a client i. This order is put in Week $(w+1)$ in $(Oo, \Theta o)$ to represent a first week of ordering delay (thus, $(Oo, \Theta o)$ can be seen as the order placed in previous week and that is now shipped by the postman). To represent the second week of delay, this order is received by supplier $(i+1)$ in Week $(w+2)$ in Oi. This order reception decreases supplier's inventory I in the same week, because products are shipped in Too and $To\Theta$, and thus in To. The following week, these shipped products are put in Week $(w+3)$ in client's Ti in order to represent a first week of shipping delay. To model the second week of shipping delay, these products are put in client's inventory in Week $(w+4)$. Therefore, to implement a two-week delay in information transmission and in transportation, as described in the previous section, order and transport variables are doubled, which explains the reason why there are To^i_w/Ti^i_w and $(Oo^i_w, \Theta o^i_w)/(Oi^i_w, \Theta i^i_w)$.

All these equations are described in detail in Section A.2.

- *Scheme-dependent equations:* The players' decision making is represented in our simulator of the QWSG, as their ordering scheme. More precisely, we need a pair of equations Op^i_w and Θp^i_w to implement each of the seven ordering schemes A, A', A", B, B', C and D, that were presented in Paragraph "Progression of the QWSG" in Subsection 6.1.2. In other words, the class of scheme-dependent equations is made up of seven pairs of equations, where each pair describes Op^i_w and Θp^i_w, and we use one of these seven pairs to implement a particular company-agent.

  The seven pairs of Op^i_w and Θp^i_w are described in Section A.3.

Finally, Figure 6.5 gives an overview of the implementation of the lumber market demand and of the LumberRetailer in our first simulator.



= =L7+S7-P8

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1				Simulator of the Québec Wood Supply Game - DAMAS and FOR@C 2004														
2																		
3																		
4				XY+infoCent														
5		Consumer		Lumber retailer														
6	Week	Demand		Delay 1 Outgoing orders	Delay 1 Outgoing tokens	Delay 2 Outgoing orders	Delay 2 Outgoing tokens	Order placed	Tokens sent	Backordered tokens	Backorders	Inventory	Outgoing transport (order)	Outgoing transport bckorder	Outgoing transport (tokens)	Outgoing transport	Transport delay 1	Transport delay 2
7	1	11		11	0	11	0	11	0	0	0	0	11	0	0	11	11	11
8	2	11		11	0	11	0	11	0	0	0	0	11	0	0	11	11	11
9	3	11		11	0	11	0	11	0	0	0	0	11	0	0	11	11	11
10	4	11		11	0	11	0	11	0	0	0	0	11	0	0	11	11	11
11	5	17		11	0	11	0	17	24	0	6	0	11	0	0	11	11	11
12	6	17		17	24	17	24	17	0	0	12	0	11	0	0	11	11	11
13	7	17		17	0	17	0	17	0	0	18	0	11	0	0	11	11	11
14	8	17		17	0	17	0	17	0	0	24	0	11	0	0	11	11	11
15	9	17		17	0	17	0	17	0	0	30	0	11	0	0	11	11	11
16	10	17		17	0	17	0	17	0	0	36	0	11	0	0	11	29	29
17	11	17		17	0	17	0	17	0	0	24	0	17	12	0	29	17	17
18	12	17		17	0	17	0	17	0	0	24	0	17	0	0	17	29	29
19	13	17		17	0	17	0	17	0	0	12	0	17	12	0	29	17	17
20	14	17		17	0	17	0	17	0	0	12	0	17	0	0	17	29	29
21	15	17		17	0	17	0	17	0	0	0	0	17	12	0	29	17	17
22	16	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
23	17	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
24	18	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
25	19	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
26	20	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
27	21	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
28	22	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
29	23	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
30	24	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
31	25	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
32	26	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
33	27	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
34	28	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
35	29	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
36	30	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
37	31	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
38	32	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
39	33	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
40	34	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
41	35	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
42	36	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
43	37	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17
44	38	17		17	0	17	0	17	0	0	0	0	17	0	0	17	17	17

Figure 6.5: Implementation of the Québec Wood Supply Game (QWSG) in Excel.

We now present our second simulator.

6.3 Implementation of the QWSG: The Software Agent Approach

Our second simulator is based on the first one, i.e., on the implementation in a spreadsheet of the QWSG. The difference with this first simulator is its increased realism, which makes it more complex. To address this complexity, companies are implemented with the agent-oriented language JACK™. JACK™ [Agent Oriented Software Group, 2003] gives tools to implement systems of intelligent agents, that are both goal-directed and reflex (event-driven), according to Russell and Norvig [2003]'s classification presented in Subsection 2.2.3. For the moment, we only use the reflex behaviour of JACK™ agents in our simulation. We now motivate the implementation of an agent-based simulator in four points:

- The first argument concerns the lack of realism of the model in the QWSG, implemented on a spreadsheet, because we have seen that it only addresses one cause of the bullwhip effect (the "misperception of feedback" [Sterman, 1989], that can be refined, at least, as "ordering and shipping delays", as shown in Chapter 5). We can make this simulation more realistic/complex when an agent-oriented programming language is used, which allows considering the other known causes of the bullwhip effect. Lee et al. [1997a,b] have proposed four causes, which are often seen as the main ones, and which were recalled in Subsection 3.2.2: demand forecast updating, order batching, price fluctuations and rationing and shortage gaming. We will show in Subsection 6.3.4 how these causes can be addressed with our second simulator.
- The second improvement allows companies to work with different types of products. This is a limitation of our first simulator, because companies can only manage one type of products, and extending this simulator to several types of product is very difficult.
- The third improvement allows companies to negotiate. That is, each company in the first simulator only takes into account its requirements concerning products to place orders (since there are no product prices, but only inventory holding and backorder costs), while it could, for example, try to order by big batches of products in order to profit of economies of scale. JACK™ is designed to implement agents with such negotiation protocols, while it is absolutely impossible in

a spreadsheet program.

- The last improvement is the enhanced realism of our second simulator by taking into account production activities, and inventory, shipping and production capacities. This realism could also have been applied to the first simulator, but we have preferred to put it directly into our new tool.

These four advantages of an agent-based simulator over our first simulator make it worth implementing a brand new simulator.

In the rest of this section, we first describe JACK™ and how the QWSG is adapted to agents. Next, we show how each part of a company is modelled in a more realistic way than what is used in the QWSG. Finally, we focus on the implementation of Lee and his colleagues' causes of the bullwhip effect in this second simulator.

6.3.1 The Agent-Oriented Programming Language JACK™

JACK Intelligent Agents™, or the short form JACK™, is an agent-oriented programming language built and integrated with the Java environment [Sun microsystems, 2003]. We present the framework JACK™ by basing our description on its designer's documentation [Agent Oriented Software Group, 2003]. Java provides an object-oriented language and an execution environment on several platforms, e.g., Microsoft Corp. [2004c]'s Windows, Free Software Foundation [2004b]'s GNU/Linux, Apple Computer, Inc. [2004]'s Mac OS…. As a result, JACK™ agents run on every platform for which a Java environment is available, i.e., for almost every platform. The components provided by JACK™ are:

- the JACK™ Agent Language is an extension of Java providing constructs to represent agent-oriented features;
- the JACK™ Agent Compiler translates JACK™ source code into Java, then calls the Java compiler to transform it into a runnable program;
- the JACK™ Agent Kernel is a runtime engine supporting the facilities used by agents.

Although JACK™ is built upon Java, it is much more than an enhancement of Java, because the relationship between JACK™ and Java is analogous to the language C's relationship to the language C++. That is, JACK™ (respectively C++) was developed as an enhancement to Java (respectively C) by providing agent-oriented concepts (respectively object-oriented concepts). Furthermore, modularity is a concept from Java, and more generally from object-oriented programming, that make easy design, implementation and reuse. This modularity is kept in JACK™ by programming each program as an entry point for the Java environment, i.e., the traditional "*main*()", and some or all of the following components:

1. Each component "*agent*" embodies all the functionalities of an *agent*. Precisely, an *agent* has *plans* and *capabilities* driven by events. These *plans* and *capabilities* may also post some *events*. Finally, agents use *beliefsets* to maintain their beliefs about the world.
2. Each component "*capability*" structures reasoning elements of *agents* into clusters implementing selected reasoning capabilities. From the implementation point of view, a *capability* is very close to an *agent*, because it is an aggregation of *plan(s), event(s), beliefset(s)* and other *capabilitie(s)*, except that a *capability* cannot be run.
3. Each component "*event*" models occurrences and messages that *agents* must address. *Events* may arise externally between *agents* (message) or internally in an *agent* when a *plan* has to trigger some other *plans*.
4. Each component "*plan*" is the procedural description of actions that an *agent* achieves when this *agent* receives a specific internal or external *event*.
5. Each component "*beliefset*" models an *agent*'s knowledge about the world. This knowledge is represented as first order relational tuples.

These components are divided into several classes to model the behaviour of *agents* in two ways: a reflex part and a pro-active part. The first part of behaviour is *event*-driven, because *agents* react in a more or less sophisticated manner to their environment (we include in an *agent*'s environment the other *agents*, and the messages sent by the other *agents*). The second part of *agent* models a more intelligent behaviour based on the theoretical BDI (Belief Desire and Intention) model from Artificial Intelligence. BDI takes into account the pro-activeness of *agents* in the following way:

- An *agent* believes something about its environement, which is implemented by setting some sets of data (*beliefsets*) according to the *agent*'s perception of the environment.

- This *agent* also desires to achieve its goals, or more precisely, the *agent* wishes to receive some specific perceptions of its environment.
- Therefore, the agent adopts an intention by selecting the appropriate plans according to its beliefs.

For the moment, we only use the *event*-driven part of agent behaviour, because our second simulator is still very similar to the first one. Indeed, our second simulator is more complex because it includes the activity of companies, but our agents only work in a "mechanical way", but the framework JACK™ allows the company behaviour to evolve easily towards a BDI architecture.

In fact, this flexibility in combining pro-active and reflex behaviours is the first JACK™ for a project. For our simulation, the interesting features of JACK™ are:

- The flexibility in combining pro-active and reflex behaviour, as just stated, because we can use this flexibility to make our companies evolve towards much more realism;
- The autonomy of agents that is taken into account by the JACK™ Agent Kernel;
- The high-level representation of behaviour, i.e., a level above object-oriented concepts;
- The suitability for distributed applications, because the source code of agents does not need adaptations when agents are moved from one computer to another;
- The ability for *agents* to work co-operatively in teams, which could be used in future developments of our simulation.

6.3.2 Adaption of the Québec Wood Supply Game to agents

The QWSG is the base of a model that we use to study the efficiency of different coordination mechanisms in the supply chain. More precisely, we have programmed intelligent agents, so that they can play a more sophisticated version of this game. Therefore, the task of every agent will be to simulate a company behaviour, and to decide when and how much to order. From a global point of view, Figure 6.6 represents the simulated supply chain. Each company has one or several inventories figured as triangles; the total height of a triangle represents inventory capacity (which is an addition to the QWSG) while its filling represents its level.

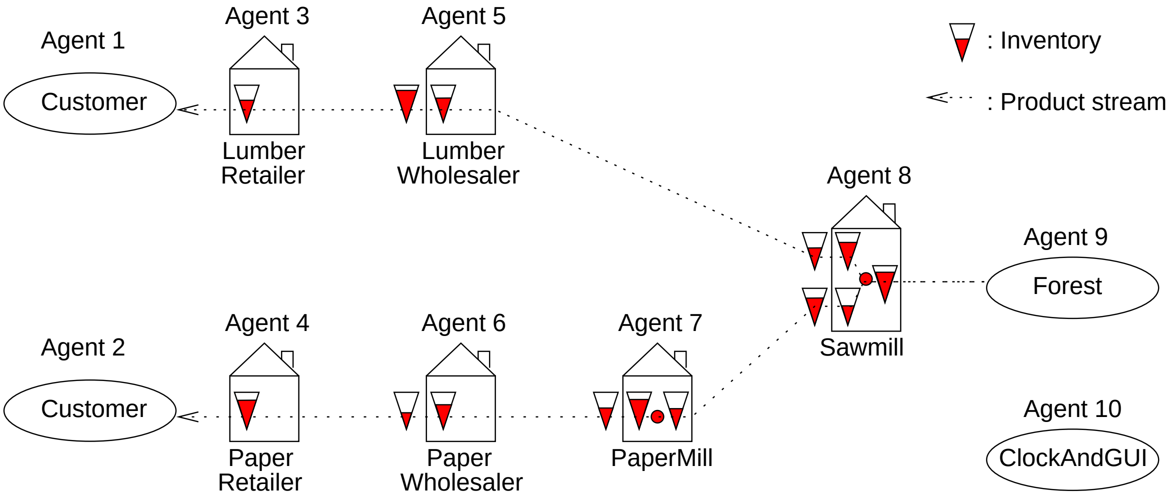


Figure 6.6: The modelled forest supply chain.

In PaperMill and Sawmill, circles represent the transformation of a quantity of material (production is another addition to the QWSG): this inventory is exclusively either full or empty. Both wholesalers are like in the QWSG: they do not have production activity and they have a truck to ship products to their retailer. The other companies differ from the QWSG: both mills manage raw material inventories due to their production activity, and none of the retailers ships products because customers come to buy them. Like in the QWSG, time is discrete[3], each company has a product inventory ready to be shipped, called _finishedProductInventory[4], and there are delays between companies modelled as a queue _productsInTruck. The PaperMill and the Sawmill are the only agents that have the basic structure shown in Figures 6.7 and 6.9 while the other agents are simplifications from this.

Figure 6.7 shows the adaptation of the PaperMill and Sawmill in the QWSG to our agent-based simulator: there are information and a product streams travelling the four functions Transport, Deliver, Make and Source of the company. We can note that these functions are based on the first level of the model SCOR from the Supply Chain Council [Supply Chain Council, 2001] illustrated in Figure 6.8, except that we add the function Transport. In fact, the model SCOR represents a company as three activities:

- Deliver: a company ships products to its clients;
- Make: a company carries out its activities;

---

[3] Time discretization makes it easier to control that simulation works as we intend it to do.

[4] The names of variables begins with a "_", and the name of JACK^{TM} plans with "Pl".

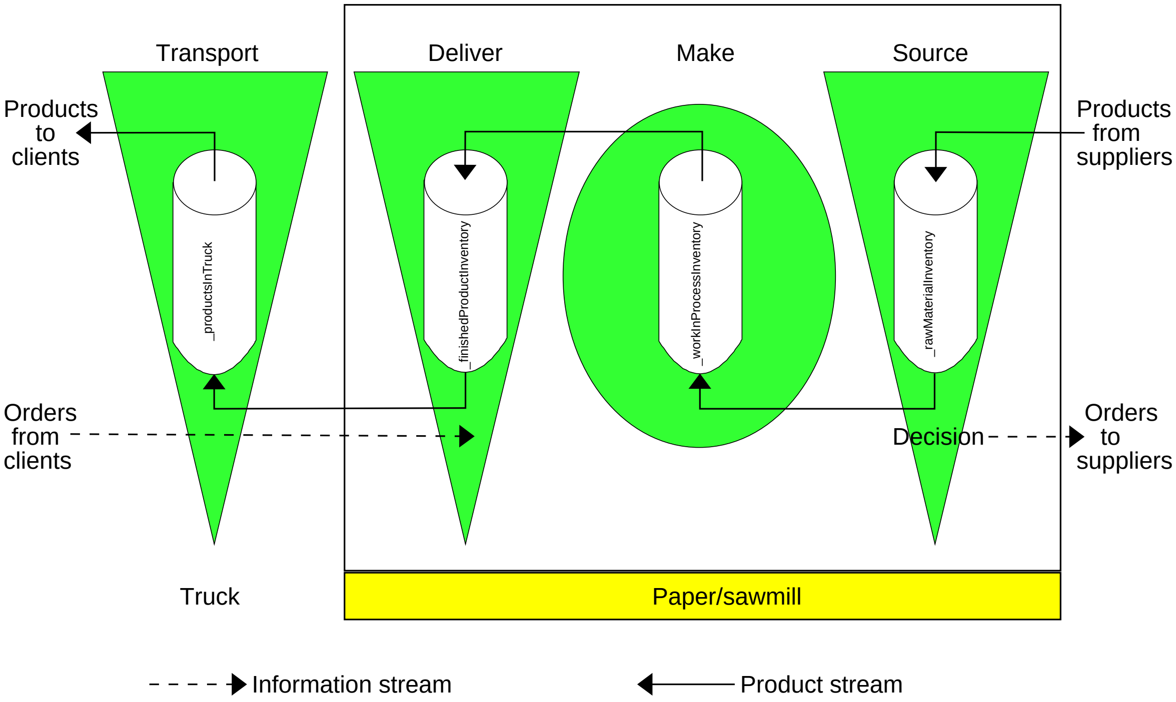


Figure 6.7: The PaperMill and Sawmill model.

- Source: a company procures products from other companies to perform its activities.

These three activities should be planned together. The additional levels of SCOR details these three activities.

We can also note that, in the first simulator, all companies were modelled with the same equations, except the Sawmill, while in the second simulation, the PaperMill and the Sawmill share the same code and other companies derive from this. The model of the PaperMill and Sawmill is based on the first level of the SCOR model from the Supply Chain Council [2001] in which we add a link Transport between the company and its client(s). This link is a truck which takes the place of the two shipping delays in the QWSG. Precisely, these delays are represented in the simulator as an inventory at the visual level and as a queue called _productsInTruck at the logical level. Product batches go through this queue and are given to the client's _rawMaterialInventory as soon as the shipping delay has elapsed. We add capacity to this queue, whereas shipping delays in the QWSG can ship as many products as needed. We add capacity to inventories too. The inventory in the QWSG, like I_w^i in the first simulator, corresponds to _finishedProductInventory in Figure 6.7.

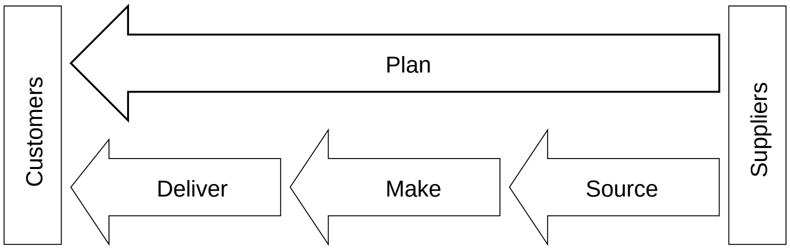


Figure 6.8: The model SCOR [Supply Chain Council, 2001].

The QWSG and our first simulator do not make any distinction between companies in the distribution network (the retailers and the wholesalers) and production companies (the PaperMill and the Sawmill). On the contrary, we make the model more realistic by adding the Make function to the PaperMill, which forces us to add the Source function. Both of these functions contain a limited inventory, called _workInProcessInventory in Make, and _rawMaterialInventory in Source. The _workInProcessInventory in the Make function represents the batch being processed, that is, it is a small inventory, which is either full or empty.

In short, the Transport, Deliver, Make and Source functions are travelled by the information and the product streams. Figure 6.9 gives more details about the implementation of these four sets of functions.

We now detail how the simulation works.

6.3.3 Simulation mechanism

As previously noted, we assume all companies work at the same time, and that this time is discrete, i.e., every company waits for the other company to complete the five days of the QWSG, before beginning the next day. Furthermore, there is a bottleneck which is situated at the level of the Sawmill, that allows us to simulate the risks of scarcities (a cause of the bullwhip effect) leading the agents to behave in a strategic manner. This behaviour is described in the PlOrdering rule of agents: agents order more than they actually need when there are fewer incoming products than were ordered.

Now, we first present the five different types of agents (i.e., the company-agents and an additional agent called ClockAndGUI) in the simulation, and then the implementation of the company-agents. As we use the JACK™ framework, this description is a summary of JACK™ plans used to model companies.

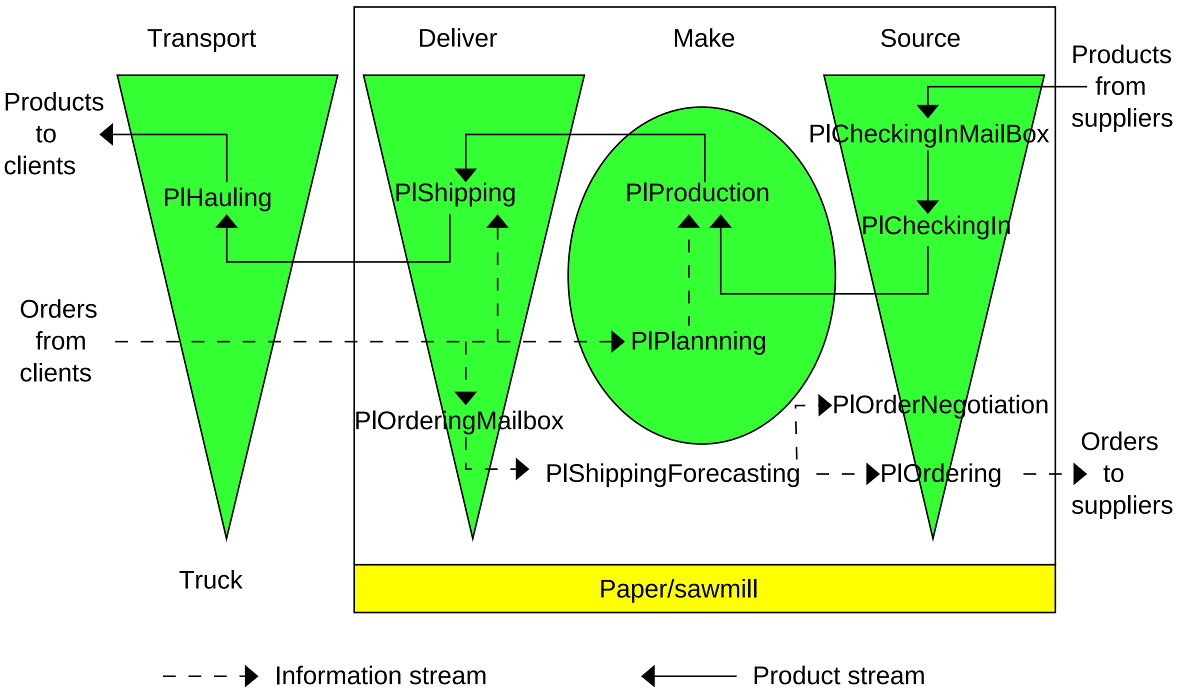


Figure 6.9: JACK™ plans in the PaperMill and the Sawmill.

Agents' Roles

To describe how to implement company-agents, we use the Russell and Norvig's PEAS (*Performance, Environment, Actuators, Sensors*) description [Russell and Norvig, 2003]. This gives a good idea of what the company-agents will perceive and do.

- P*erformance*: it is basically measured as the standard deviation of placed orders, but we can also consider the standard deviation of inventory level, inventory holding cost, stockout durations. . ..
- E*nvironment*: it encompasses other company-agents and the market. For the simulation, we also add an agent managing the clock and the graphical user interface.
- A*ctuators*: as everything is simulated, company-agents have no actuators because they are only able to send messages to other agents.
- S*ensors*: as everything is simulated, company-agents have no sensors because they are only able to receive messages from other agents.

Agents can be divided into several categories:

1. The Customers are modelled as two agents buying products from their retailers. Customer agents are very simple, because they only place orders according to a distribution law representing different kinds of consumption patterns presented in Table 7.1.

2. The Retailers and Wholesalers, i.e., the distribution network, are companies having neither production activity nor raw material inventory to manage: products coming from the supplier do not wait in the _rawMaterialInventory because these agents directly move products from _rawMaterialInventory to _productsInTruck. Moreover, these four agents have to place orders to their suppliers.

3. The PaperMill and the Sawmill are the basic company-agents in our model in Figure 6.9, because they have the Transport, Deliver, Make and Source functions. Make and Source share a common code for these two company-agents, but are initialized differently, e.g., _workInProcessInventory and _finishedProductInventory are arrays containing an integer in the case of the PaperMill, and two integers in the case of the Sawmill, and _productsInTruck are arrays containing one array of integers in the case of the PaperMill, and two arrays of integers in the case of the Sawmill.

4. The Forest is the most upstream company-agent, which has thus no supplier and does not place any orders. It is assumed to be an infinite source of wood in the QWSG, but we assume in our model that its capacity is given by a cutting plan representing companies' procurement contract with the provincial government.

5. The ClockAndGui is an agent multi-casting an event representing the time in the whole supply chain. Each tick of this clock makes other agents perform actions by triggering their JACK™ plans. This agent can also display information on the GUI (Graphical User Interface), and/or write raw simulation outcomes on the shell, and/or put simulation outcomes in an Excel file[5].

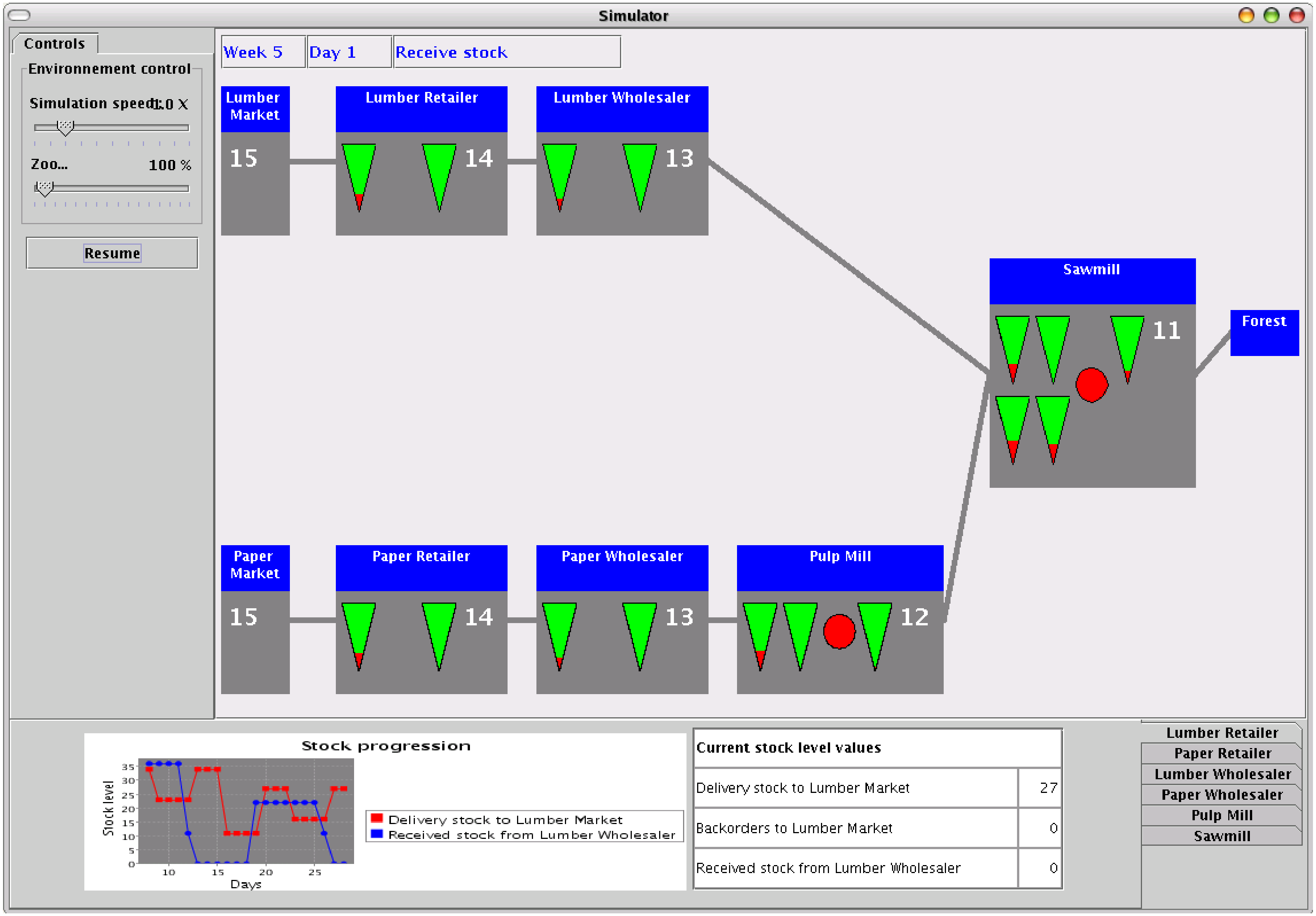
   Figure 6.10 gives an overview of the information displayed by the ClockAndGui agent[6].

We should finally note that transportation could have been modelled as truck agents instead of as a function in each company. It would have allowed us to focus on impacts of unanticipated transportation events on the bullwhip effect. We assume for the moment that transportation is a problem that can be simply viewed as a queue managed by each company.

---

[5] We use here the Java Excel API 2.3.12 from Andy Khan [2004].

[6] I would like to thank Eve Levesque who adapted the graphical user interface from the Nereus project [Soucy, 2004; Plamondon, 2003; Plamondon et al., 2003; Paquet, 2001] in the DAMAS [2004] research group.



Week 5 Day 1 Receive stock

Simulation speed: 1.0 X

Zoo... 100 %

Current stock level values	
Delivery stock to Lumber Market	27
Backorders to Lumber Market	0
Received stock from Lumber Wholesaler	0

Figure 6.10: Graphical user interface of the second simulator.

Further detail on the implementation of the JACK™ agents is in Section A.4 of Appendix A.

6.3.4 Addressing known causes of the bullwhip effect

Everything we have just described replicates a more realistic supply chain than the QWSG. Similar to our first simulation, the only cause of the bullwhip effect that appears in this simulation is the consequence of ordering and shipping delays. We now focus on taking into account the four causes and solutions of the bullwhip effect proposed by Lee et al. [1997a,b], because these are the most cited ones.

To this end, the behaviour of company-agents, i.e., their ordering scheme, has to be adapted to address these four causes of the bullwhip effect. In fact, adding all causes is easier than adapting each company's ordering rule according to these causes. In other words, making the simulator more realistic is easier than making decision in such a realistic environment. For example, it is easy to create a price fluctuation: the price of an item may be, for instance, twice the price at which the company has bought it (to make a profit) minus a tenth of the inventory level, that is, the more products the company has, the lower the charged price is, in order to empty the inventory and thus save holding costs. But we must then adapt how companies order when they want to take advantage of these promotions in the most efficient way. In other words, ordering patterns represent how companies behave to take advantage of available opportunities. These four causes are [Lee et al., 1997a,b]:

1. *Updating of demand forecasting*:

   Proposed solution: make demand data from each company available to its supplier. This allows the supplier to make better forecasts instead of making forecasts on its client's forecasts. This is information sharing.

   Consequence for the agents: Client's plan PlOrdering has to be adapted to implement our Ordering Schemes B and D, client's plan PlShippingForecasting has to be implemented to use the information shared with B and D, and supplier's plan PlPlanning may also be implemented to plan future production of the company.

   Example of ordering pattern: Ordering Schemes B and D, and PlShippingForecasting is a moving average, and PlPlanning is a FIFO (First In, First Out).

2. *Order batching*:

   Proposed solution: break up order batches by making transaction costs lower using an electronic commerce system.

Consequence for agents: production has to be made by batches (that is _workInProcessInventory is either full or empty) and transportation costs depend on the number of trucks used instead of the number of products shipped (so companies will prefer to ship by full truckloads instead of less than full truckloads).

Example of ordering pattern: placed orders by PlOrdering or PlOrderingSS can only take discrete values and these values correspond to full truckloads.

3. *Price fluctuation*:

   Proposed solution: stabilize prices (e.g. EDLP strategy - Every Day Low Price) to avoid customer and company overordering and stockpiling during promotions.

   Consequence for agents: quantity ordered depends on the price of product (client) and this price depends on inventory levels (supplier).

   Example of ordering pattern: placed orders by PlOrdering and PlOrderingSS are equal to the needed quantity plus the difference between the nominal price and the current price, where the current price is calculated by the supplier (e.g., the current price is the nominal price plus the difference between the current inventory level and a nominal inventory level).

4. *Rationing and shortage gaming*:

   Proposed solution: eliminate gaming in shortage situations by allocating products to customers in proportion to their past sales instead of in proportion to current orders. Therefore, when a company has several clients but is not able to ship them what they have ordered, the client will not overorder (i.e., gamble) in the hope of receiving its actual needs.

   Consequence for agents: agent overorders if it does not receive what it ordered four weeks before (because products ordered in week w are received in week $w + 4$).

   Example of ordering pattern: order the current requirement plus the difference between incoming shipment and what was ordered four weeks earlier.

6.4 Conclusion

We have presented the two simulators, which were implemented to study the bullwhip effect. The first simulator is the strict implementation of the QWSG (Québec Wood Supply Game), and the second one is based on the first one, on which four improvements have been added. The first improvement is the possibility of simulating several causes of the bullwhip effect, instead of only one in the QWSG in our fist simulator, where this cause is "ordering and shipping delays". The second improvement allows companies to process several types of products, instead of only one. The third enhancement concerns the ability for agents to negotiate in order to find agreements on transactions, instead of working on a mode "order implies shipping". The last improvement could have been added to our first simulator, and is the addition of a manufacturing activity by some companies. Finally, some tests have to be achieved to validate that this second simulator works as designed.

As the first simulator is less sophisticated, and has been tested, it has allowed us to understand the two principles to reduce the bullwhip effect, that are proposed in Chapter 5 and applied in our two ordering schemes. Even though this model runs in a spreadsheet program, companies are modelled as *autonomous business entities* that have to place orders by themselves. Precisely, companies are reflex agents applying their ordering scheme in a "mechanical way". Nevertheless, we can study the selfishness of companies even though they work with a mechanical way. In particular, we see in the next two chapters that this first simulator allows studying the three following points:

- How does the supply chain behave when all companies apply the same ordering scheme?
- What is the overall cost of the supply chain and the cost distribution among companies in this case when all companies apply the same ordering scheme?
- What happens when companies may choose their ordering scheme independently?

The next chapter addresses the two first of these questions.

Chapter 7

First Series of Experiments: Homogeneous Supply Chains

The two ordering schemes proposed in Chapter 5 are studied in this chapter with the first simulator introduced in Chapter 6. Let us recall, that this simulator implements the company and supply chain models in the Québec Wood Supply Games (QWSG). Let us also recall, that we have proposed the ordering schemes B and D as illustrations of our two principles, i.e., "the lot-for-lot ordering policy eliminates the bullwhip effect but does not manage inventory", and "companies should react only once per market consumption change, by over- or underordering". In order to show the efficiency of our two principles for a supply chain, we carry out the following comparisons between the seven schemes A, A′, A″, B, B′, C and D, where none, one or two of our principles are followed:

- *Comparison of* A *and* A′*:* This shows that demand-based orders are efficient in reducing the bullwhip effect;
- *Comparison of* A′ *and* B*:* This shows that our two principles must be followed, because A′ loosely follows the second one, and is thus inefficient;
- *Comparison of* B *and* B′*:* This shows that information sharing with (O, Θ) orders does not reduce the bullwhip effect alone, but our two principles are also required;
- *Comparison of* A *and* C*:* This is a confirmation that information centralization reduces the bullwhip effect;
- *Comparison of* B *and* D*:* This is also a confirmation that information centralization reduces the bullwhip effect;

- *Comparison of* B′ *with* B *and* D*:* This shows that our first principle is necessary;
- *Comparison of* A′ *with* B *and* D*:* This shows that our second principle is necessary.

Notice that in this dissertation, two series of experiments are carried out. The first series is presented in this chapter while the second series will be described in Chapter 8. For the moment, this first series studies a homogeneous supply chain in order to check that the two ordering schemes B and D, based on our two principles, are efficient for the whole supply chain. During this first series of experiments we assume that the supply chain is homogeneous, that is, all companies use the same ordering scheme. For each of the seven ordering schemes, we use one of nineteen market consumption patterns, i.e., the lumber and market customers buy during each fifty week simulation by following one of the nineteen series of end-customer demand. As a consequence, $7 * 19 = 133$ simulations are carried out. This methodology is presented in Section 7.1.

Then, we present how logistics costs are evaluated in the first series of experiments. The method to evaluate costs is based on the method in the QWSG, except that a little modification improves its realism. This modification is presented in Section 7.2.

Finally, we describe the core of this chapter, that is, the experimental outputs of the first series of experiments. To this end, we first show the orders placed and the inventory level for each company under one particular market consumption pattern. We confirm here that our two principles reduce the bullwhip effect, and that there is a change in the distribution of the backorder durations in the supply chain. Then, we simulate the supply chain under the eighteen other market consumption patterns. These simulation outcomes are presented with four metrics, because giving additional details (i.e., presenting in detail, like under the first market consumption pattern) represents a large bulk of data. We state here that D is most of the time the most efficient ordering scheme. Finally, a discussion concludes Section 7.3.

7.1 Description of our Methodology

In the first series of experiments that is described in this chapter, we use the first simulation model to simulate the Québec Wood Supply Game (QWSG) under the seven ordering schemes A, A′, A″, B, B′, C and D, and under the nineteen demand market patterns indicated in Tables 7.1 (detailed in Tables B.1, B.2 and B.3), as illustrated in Algorithm 7.1. More specifically, these three tables present the nineteen end-customer demands D_w^{lumber} and D_w^{paper} for each of the fifty weeks of a simulation.

Algorithm 7.1 methodology in the first series of experiments

```
returns orders and inventory level for each company and each week
for each of the nineteen market consumption patterns $m$ in Figure 7.1 do
    for each seven of the ordering schemes $r$ in Figure 6.1.2 do
        simulate a supply chain using $r$ under consumption $m$
```

Most of these nineteen market consumption patterns are between eleven and seventeen. On the contrary, the QWSG bounds its demand pattern between four and eight. We do not use this pattern, because there are a lot of “twos” here, that is, the two market (lumber and paper) demands are $4 = 2 * 2$ and $8 = 4 * 2$, and ordering and shipping delays are also equal to two weeks. When we study the parameter setup in the considered ordering schemes in Section 8.2, some other “twos” appear in the parameters in Schemes B and D (recall that B and D were proposed in Chaper 5 as implementations of our two principles to reduce the bullwhip effect), and we are not sure if they are due either to the market consumption pattern, or to the ordering and shipping delays, which are also equal to “two”. In order to avoid this question, we have changed four and eight in the QWSG for the two prime numbers eleven and seventeen. Therefore, when we have to set an ordering scheme parameter to “two”, this is not due to the market consumption. As a consequence of this choice for eleven and seventeen, the nineteen market consumption patterns used to study the seven ordering schemes are indicated in Table 7.1, and described now:

1. Step: The Step demand pattern indicates that the two market demands for eleven products per week in the four first weeks, then for seventeen products until the end of the simulation. We note this demand $D_w^{\text{lumber}} = D_w^{\text{paper}} = 11$ for $w \in \{1, 2, 3, 4\}$, and $D_w^{\text{lumber}} = D_w^{\text{paper}} = 17$ for $w = 5, 6, 7, \ldots 50$.
2. Inversed step: This demand corresponds to a demand for seventeen products in the four first weeks, and eleven products until the end, that is, the inverse of the previous market consumption pattern.
3. Dirac: It indicates a demand for eleven products per week throughout the simulation, except in Week five, where the demand is for seventeen products.
4. Inversed Dirac: It indicates a demand of seventeen products per week, except in Week five, where eleven products are consumed by both markets, which is the inverse of the previous pattern.
5. Increase: It corresponds to the following demand per week: 11, 11, 11, 11, 12, 13, 14. . ., 56, 57.

6. **Decrease**: It is a demand for 57, 57, 57, 57, 56, 55, 54..., 12, 11, that is, the inverse of the previous pattern.

In the three seasonalities, market consumption is eleven on average:

7. **Weak seasonality**: It has a demand for 11, 11, 11, 11, 12, 11, 10, 11, 12, 11, 10, 11, 12... products, that is, eleven products on average, with a variation of -1, 0 or +1.
8. **Medium seasonality**: It corresponds to the following demand: 11, 11, 11, 11, 12, 13, 12, 11, 10, 9, 10, 11, 12, 13, 12, 11, 10, 11, 10... products, that is, eleven products on average, with a variation of -2, -1, 0, +1 or +2.
9. **Strong seasonality**: It indicates a demand of 11, 11, 11, 11, 12, 13, 14, 13, 12, 11, 10, 9, 8, 9, 10, 11, 12, 13, 14, 13, 12, 11, 10, 9, 10, 11... products, that is, eleven products on average, with a variation of -3, -2, -1, 0, +1, +2 or +3.
10. **Uniform randoms**: They are represented by 10A to 10J, which are ten uniform distributions of demand, detailed in Tables B.2 and B.3 because of the required place to describe them. These are uniform distributions on the interval of integers [11; 17]. These ten distributions are generated once and are next memorized to use them under the seven ordering schemes. We record these ten random demands, because simulations outcomes may be specific to certain demand patterns. Finally, the patterns 10A to 10J give the same distribution of demand to the two markets, but the exact values are different, i.e., $D_w^{\text{lumber}} \neq D_w^{\text{paper}}$ for several Weeks w. On the contrary, the nine first market consumption patterns are exactly the same for both markets, i.e., $D_w^{\text{lumber}} = D_w^{\text{paper}}$ for every Week w.

Before the description of the first series of experiments in Subsection 7.3, we make the QWSG method more realistic for cost calculation. In fact, the main metric to measure the efficiency of an ordering scheme in this chapter is the standard deviation of placed orders, because it is the measure of the bullwhip effect that we would like to reduce. But we also consider the costs incurred by the consequences of the bullwhip effect, i.e., the costs incurred by higher inventory levels and customer service reductions (backorder). This is the reason why we now present the method to calculate these costs.

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…	49	50
1 Step	11	11	11	11	17	17	17	17	17	17	17	17	17	17	17	17	17	…	17	17
2 Inversed step	17	17	17	17	11	11	11	11	11	11	11	11	11	11	11	11	11	…	11	11
3 Dirac	11	11	11	11	17	11	11	11	11	11	11	11	11	11	11	11	11	…	11	11
4 Inversed Dirac	17	17	17	17	11	17	17	17	17	17	17	17	17	17	17	17	17	…	17	17
5 Increase	11	11	11	11	12	13	14	15	16	17	18	19	20	21	22	23	24	…	56	57
6 Decrease	57	57	57	57	56	55	54	53	52	51	50	49	48	47	46	45	44	…	12	11
7 Weak seasonality	11	11	11	11	12	11	10	11	12	11	10	11	12	11	10	11	12	…	12	11
8 Medium seasonality	11	11	11	11	12	13	12	11	10	9	10	11	12	13	12	11	10	…	10	9
9 Strong seasonality	11	11	11	11	12	13	14	13	12	11	10	9	8	9	10	11	12	…	8	9
10 A→J Uniform random on [11, 17]	see Tables B.2 and B.3.																			

Table 7.1: The nineteen consumption patterns, where $D_w^{\text{lumber}} = D_w^{\text{paper}}$ (detailed in Tables B.1, B.2 and B.3).

7.2 QWSG costs

In this disseration, we consider three methods to calculate the costs induced by the bullwhip effect. Basically, these costs depend on inventory holding and backorder costs. The first method is referred as QWSG, and considers that a backordered product costs twice the cost of an item in inventory. Precisely, the calcul of company i's cost C^i is the sum of company i's inventory plus two times the sum of its backorders during the whole simulation (fifty weeks): the cost of company i is $C^i_{\text{QWSG}} = \sum_{w=1}^{50}\{I^i_w + 2 * (Ob^i_w + \Theta b^i_w)\}$. This is exactly the method adopted in the QWSG, and copied from the Beer Game. In our previous papers [Moyaux et al., 2004c, 2003a], costs were calculated with this method.

In this chapter, we add a ratio of $0.37/50$ to represent inventory holding costs, according to the value of the total logistics cost given by Nahmias [1997]. In fact, products owned by the Sawmill are cheaper than products owned by the LumberRetailer, because the value of products increases along the supply chain due to logistics operations[1]. In fact, when the Sawmill sells an item to the PaperMill, some money is charged, only because this product was carried:

- 28% of the value of the product corresponds to the value of the interest rate that corresponds to the opportunity cost of alternative investment related to a number of standard accounting measures (internal rate of return, return on assets...);
- 2% for taxes and insurance;
- 6% for holding this product in inventory (loan, supervision, heating, lighting... of the warehouse);
- 1% for breakage and spoilage.

Such costs are called "improved" costs, since they are made of the addition of $(28 + 2 + 6 + 1) = 37\%$ of value of each product by each company. This 37% represents annual costs; we divide this ratio by the number of week in a year to get the cost of a week. That is, the companies have the following costs:

- The Sawmill pays the inventory holding and backorders like in the QWSG, that is, the QWSG cost is multiplied by a rate $(1 + 0.37/50)^0 = 1$;

---

[1] Logistics costs also increase due to manufacturing operations by the PaperMill and Sawmill, but we will only take these operation costs into account in Chapter 8 in "realistic" costs.

- The PaperMill and the LumberWholesaler pay products 0.37/50 more expensive than the Sawmill, that is, QWSG cost is multiplied by a rate $(1 + 0.37/50)^1 = 1.0074$;
- And so on.

As a consequence, the "improved" costs are $C^i_{\text{improved}} = (1 + 0.37/50)^k * C^i_{\text{QWSG}}$, where k depends on the considered company. Note that k is a power in the expression "$(1 + 0.37/50)^k$", and not the indice of an agent. Therefore, we use the following equations to calculate the individual costs:

1. $C^1_{\text{improved}} = (1 + \frac{0.37}{50})^2 * \sum_{w=1}^{50}\{I^1_w + 2 * (Ob^1_w + \Theta b^1_w)\}$ for the LumberRetailer,
2. $C^2_{\text{improved}} = (1 + \frac{0.37}{50})^3 * \sum_{w=1}^{50}\{I^2_w + 2 * (Ob^2_w + \Theta b^2_w)\}$ for the PaperRetailer,
3. $C^3_{\text{improved}} = (1 + \frac{0.37}{50})^1 * \sum_{w=1}^{50}\{I^3_w + 2 * (Ob^3_w + \Theta b^3_w)\}$ for the LumberWholesaler,
4. $C^4_{\text{improved}} = (1 + \frac{0.37}{50})^2 * \sum_{w=1}^{50}\{I^4_w + 2 * (Ob^4_w + \Theta b^4_w)\}$ for the PaperWholesaler,
5. $C^5_{\text{improved}} = (1 + \frac{0.37}{50})^1 * \sum_{w=1}^{50}\{I^5_w + 2 * (Ob^5_w + \Theta b^5_w)\}$ for the PaperMill, and
6. $C^6_{\text{improved}} = (1 + \frac{0.37}{50})^0 * \sum_{w=1}^{50}\{I^{6\text{-lumber}}_w + 2 * (Ob^{6\text{-lumber}}_w + \Theta b^{6\text{-lumber}}_w) + I^{6\text{-paper}}_w + 2 * (Ob^{6\text{-paper}}_w + \Theta b^{6\text{-paper}}_w\}$ for the Sawmill.

Finally, we will introduce costs adapted to the Québec wood industry in Subsection 8.2.1. This third method will be referred as "realistic". But for the moment, we present the simulation outcomes obtained with C_{improved}.

7.3 Simulation Outcomes

We can now study the bullwhip effect in the QWSG, that is, the fluctuations in orders placed by each player. We have used the first implementation of the QWSG that was presented in Section 6.2. This is the exact simulation of the QWSG, rather than its enhancement. This simulation was run with Microsoft Excel 2000 with a PC with a processor AMD Athlon XP 2200+ from Advanced Micro Devices Inc. [2004] under Microsoft Windows 2000 Service Pack 4.

7.3.1 Simulation Outputs

Format of figures

Figures 7.1, 7.3, 7.5, 7.7, 7.9, 7.11, and 7.13 present demands in the supply chain. In these figures, the first curve from the bottom represents the market consumption under the Step demand: the market demand is eleven products during the first four weeks, followed by seventeen products until the end of the simulation. Next, the second curve shows LumberRetailer's and PaperRetailer's placed orders under a particular ordering scheme. The third curve indicates LumberWholesaler's and PaperWholesaler's. Finally, next-to-last curves represents PaperMill's and last curves are Sawmill's orders.

Similarly, Figures 7.2, 7.4, 7.6, 7.8, 7.10, 7.12, and 7.14 present inventory level (and backorders when inventory level is negative) in the supply chain. In these figures, the first curve from the bottom shows LumberRetailer's and PaperRetailer's inventory level. The second curves indicate LumberWholesaler's and PaperWholesaler's inventory. Finally, next-to-last curves represent PaperMill's inventory, and last curves are Sawmill's lumber and paper inventories.

The ranges in Figures 7.1 to 7.14 are adapted to the simulation outcomes. For this reason, small fluctuations may appear large, because of the scale. Orders and inventories are measured in simulated units, like in the QWSG, and not in real units. Moreover, when (O, Θ) orders are used, Figures 7.1, 7.3, 7.5, 7.7, 7.9, 7.11, and 7.13 represent the sum of the two types of orders $(Op^i_w + \Theta p^i_w)$, and Figures 7.2, 7.4, 7.6, 7.8, 7.10, 7.12, and 7.14 the sum of inventory and backordered orders $(I^i_w + Ob^i_w + \Theta b^i_w)$. Finally, note that these seven figures are obtained with empty initial inventories, except for the Sawmill with Schemes B and D ($\forall i \in \{1, 2, 3, 4, 5, 6\text{-lumber}\}, I^i_1 = 0$ with all schemes, $I^{6-paper}_1 = 0$ with A, A', A", B' and C, and $I^{6-paper}_1 = 6$ with B and D), because of the problem addressed in Subsection C.4 (how to aggregate lumber and paper requirements by the Sawmill?) leads to less "pretty" curves due to backorders induced by this problem. Nevertheless, in the next subsection, all Tables 7.2, 7.3, 7.4 and 7.5 will be obtained with empty initial inventories ($\forall i, I^i_1 = 0$) for all companies. Let us also recall that Figures 7.1 to 7.14 are all obtained under the Step market consumption pattern.

Simulation Outputs with Ordering Scheme A

The ordering rule used in Scheme A gives very poor results. For example, the Sawmill orders more than one hundred products in some weeks in Figure 7.1, while the market

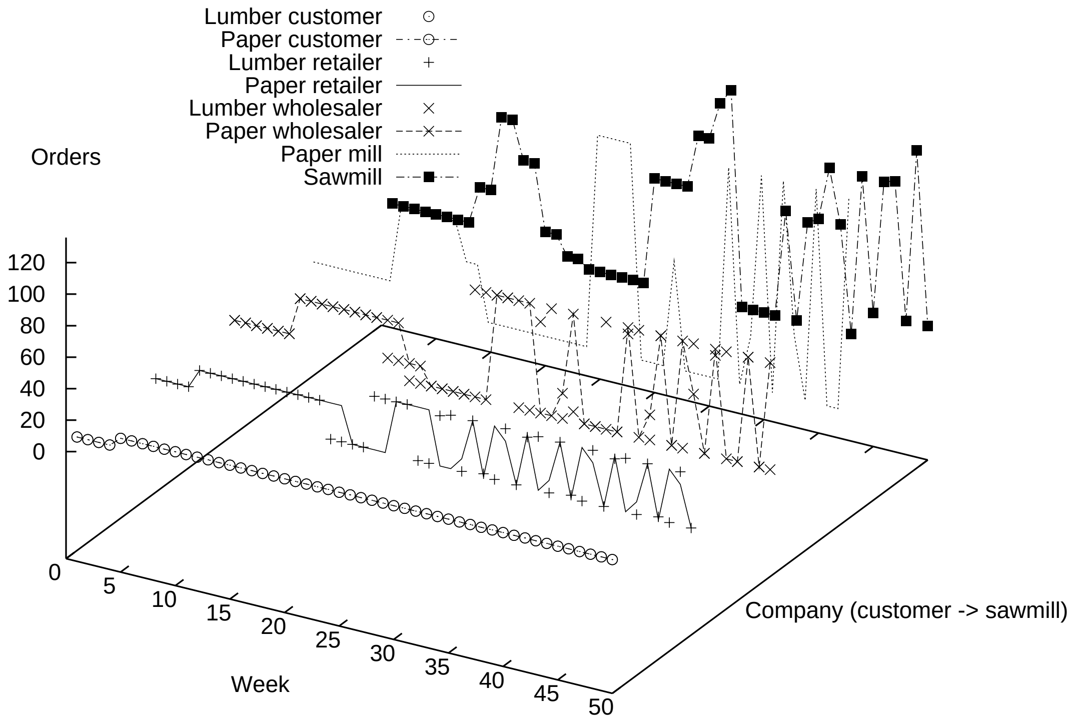


Figure 7.1: Orders in Scheme A (with $O+\Theta$ orders, without information centralization).

demand is only 17, according to the Step demand in Table 7.1, and Sawmill's inventories are over seven hundred units and are sometimes negative, that is, there are backorders in Figure 7.2.

Simulation Outputs with Ordering Scheme A′

Next, Figure 7.3 shows demand in the supply chain when every company uses Scheme A′. There are almost no bullwhip effect in this demand, because only some peaks of the demand (i.e., some overorders) appear in Figure 7.3, while curves representing companies' orders are similar to the market consumption (first curves) during the rest of the time. For example, the Sawmill orders as the market consumes, except when the Sawmill places the three huge orders represented by the three peaks of placed orders in the last curve in Figure 7.3. The problem is every company's peaks are too large, which leads to huge inventory levels in Figure 7.4.

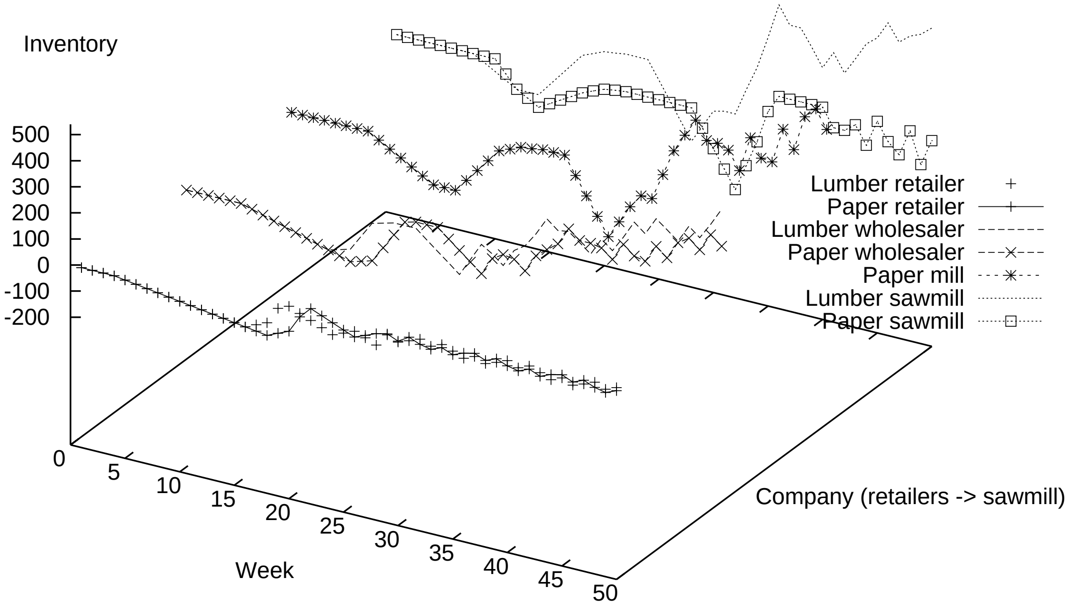


Figure 7.2: Inventories in Scheme A (with $O + \Theta$ orders, without information centralization).

Comparison of Schemes A and A′

Ordering schemes used in Schemes A and A′ do not use information sharing, i.e., our first principle, but Scheme A′ has very stable orders, conversely to Scheme A. The difference between these two schemes is that A′ almost satisfies our second principle, because companies react only once, twice or three times to each market consumption change, which is reflected by the maximum of three peaks on demand curves in Figure 7.3.

Since Scheme A′ does not use (O, Θ) orders, this reduction of the bullwhip effect cannot be explained by information sharing. Another way to explain this reduction is to focus on the basis upon which orders are placed. On the one hand, Scheme A′ bases its orders on demand, and stabilizes orders very quickly according to the market demand. On the other hand, we have seen that Scheme A bases its orders both on demand and on inventory variations, and thus, its orders fluctuate as many times as inventory. Since there are several inventory variations for each market change (this is true for all the tested Schemes A, A′, A″, B, B′, C or D), orders in Scheme A never stabilize.

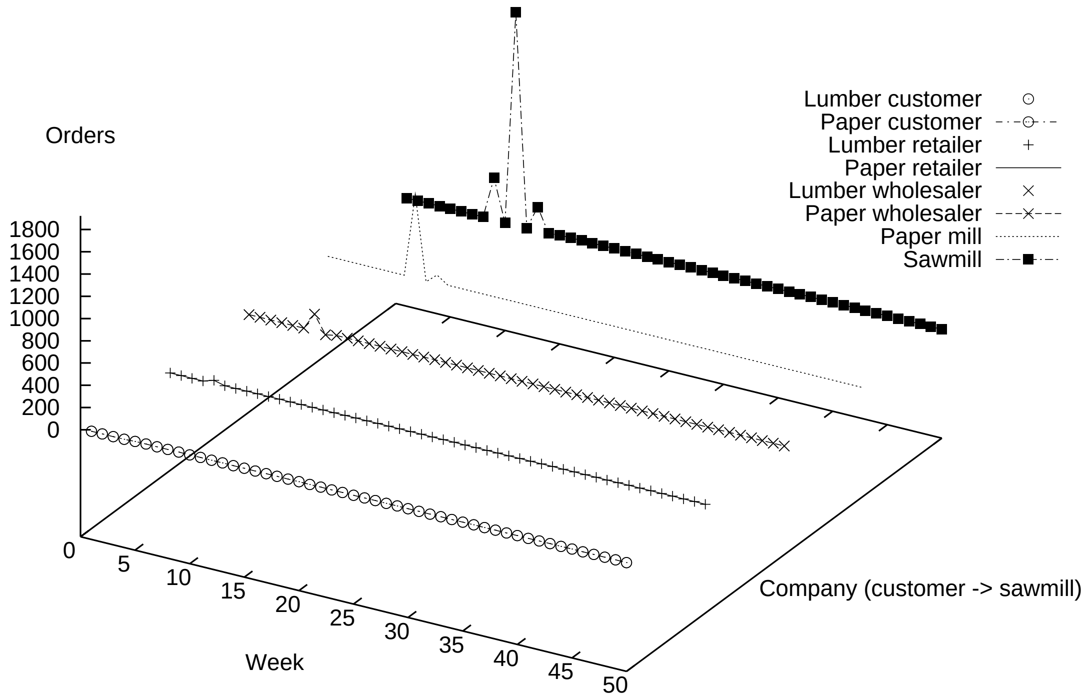


Figure 7.3: Orders in Scheme A′ (with $O{+}\Theta$ orders, without information centralization).

The fact that A′ only bases its orders on the incoming demand, while A considers both the incoming demand and the inventory level, explains why there is a great bullwhip effect in Scheme A, but not in Scheme A′. These two experiments show that orders based on demand instead of on inventory reduces the bullwhip effect in the QWSG. However, orders based on demand may be inefficient in practice, even if they are the best ones in the QWSG, because they ignore that actual inventories are affected by breakage, spoilages, thefts.... In short, the bullwhip effect is almost eliminated with A′ in our simulations, while it was really important with A. This shows that:

> Demand-based orders with A′ are more efficient in reducing the bullwhip effect than orders considering both demand and inventory level in A. But, as we have just stated, such a demand-driven scheme is not applicable in practice.

Another drawback of A′ is the fact that each company should react only once to each market consumption change according to our second principle, while the SawMill has three peaks of orders in Figure 7.3. From our point of view, this is a problem

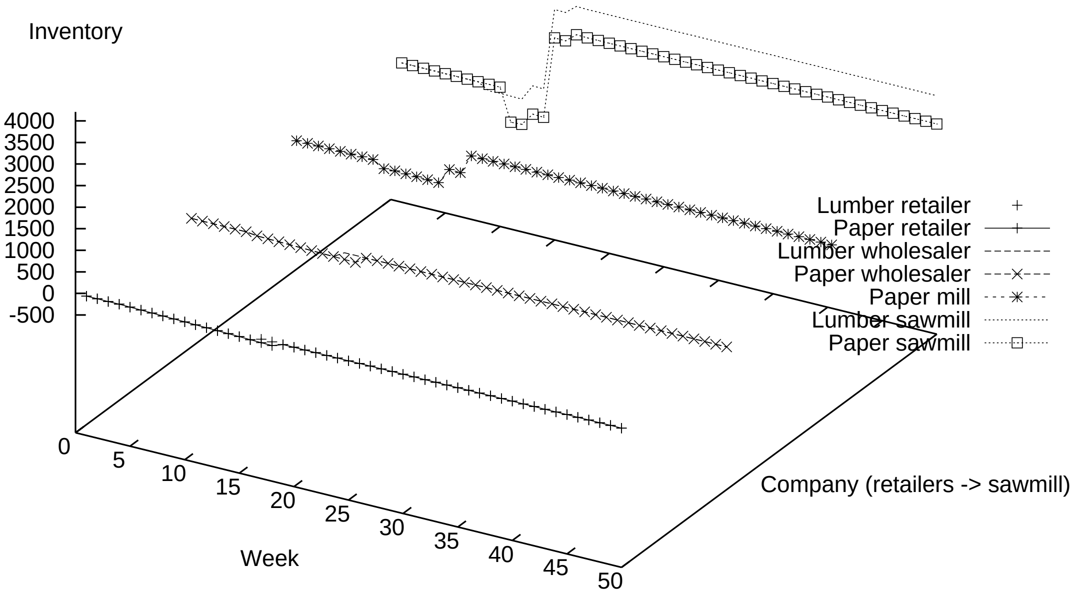


Figure 7.4: Inventories in Scheme A' (with $O + \Theta$ orders, without information centralization).

because the number and the amplitude of these peaks are not easy to predict, and thus, managing inventories with A' is not easy to effect. In other words, overorders are much too great, which lead to huge inventory levels in Figure 7.4, and the way to regulate this is not clear at a first glance.

Although the goal of this thesis is not the study of this scheme, it would be interesting to study how to obtain this regulation in Scheme A' as a future work, in order to keep the short bullwhip effect, but with a lower amplitude, so that inventory is managed in a more efficient way. Maybe Scheme A' could both manage inventories and reduce the bullwhip effect without the information sharing with (O, Θ) orders required by our two principles.

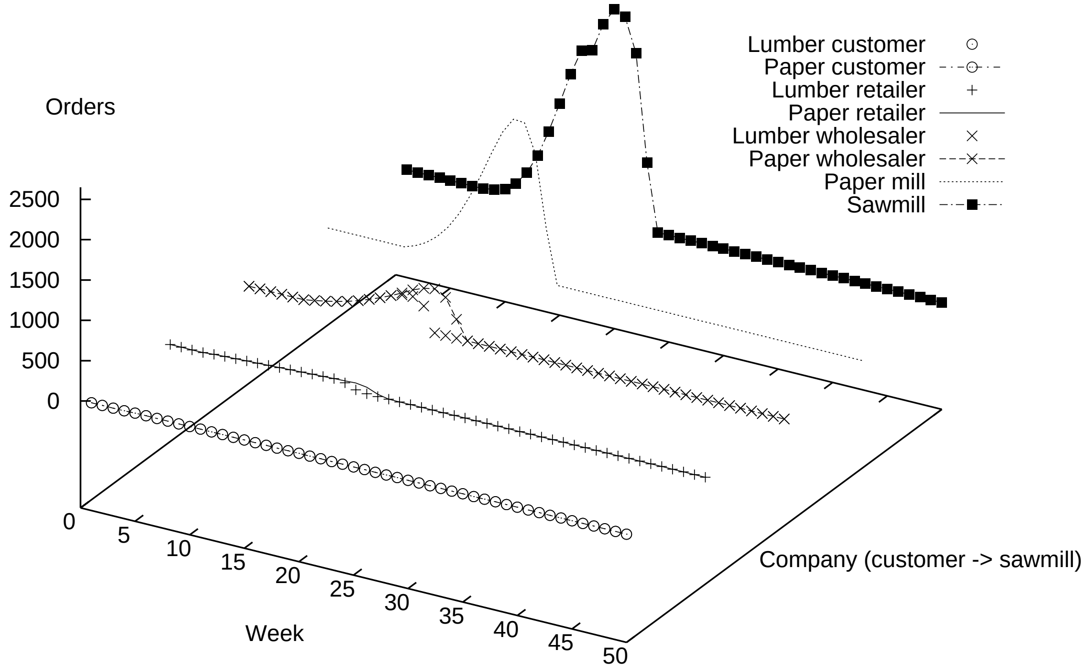


Figure 7.5: Orders in Scheme A″ (with $O + \Theta$ orders, without information centralization).

Simulation Outputs with Ordering Scheme A″

Figures 7.5 and 7.6 show the behaviour induced by Scheme A″, i.e., by an (s, S) ordering policy. The (s, S) policy is based on the model of the Economic Order Quantity (EOQ) from the inventory management literature, in which the company orders for $(S - I)$ products when its inventory I is lower than s. This model aims at optimizing the cost of the inventory system of a company that is assumed to be isolated, and not in a supply chain. We have adapted this optimization model to our simulation in Subsection B.2. The results of this optimization shows that, for every company i, $C^i_{\text{realistic}}$ is minimum when $s = 0$ and $S = Oi^i_w$, where Oi^i_w is the company i's incoming order in Week w.

These parameters are optimal under the assumptions in the EOQ. The problem is that these hypothesis are not met in our simulation. The unfulfilled assumptions from EOQ are as follows:

- the considered company has infinite sources of products as supplier;

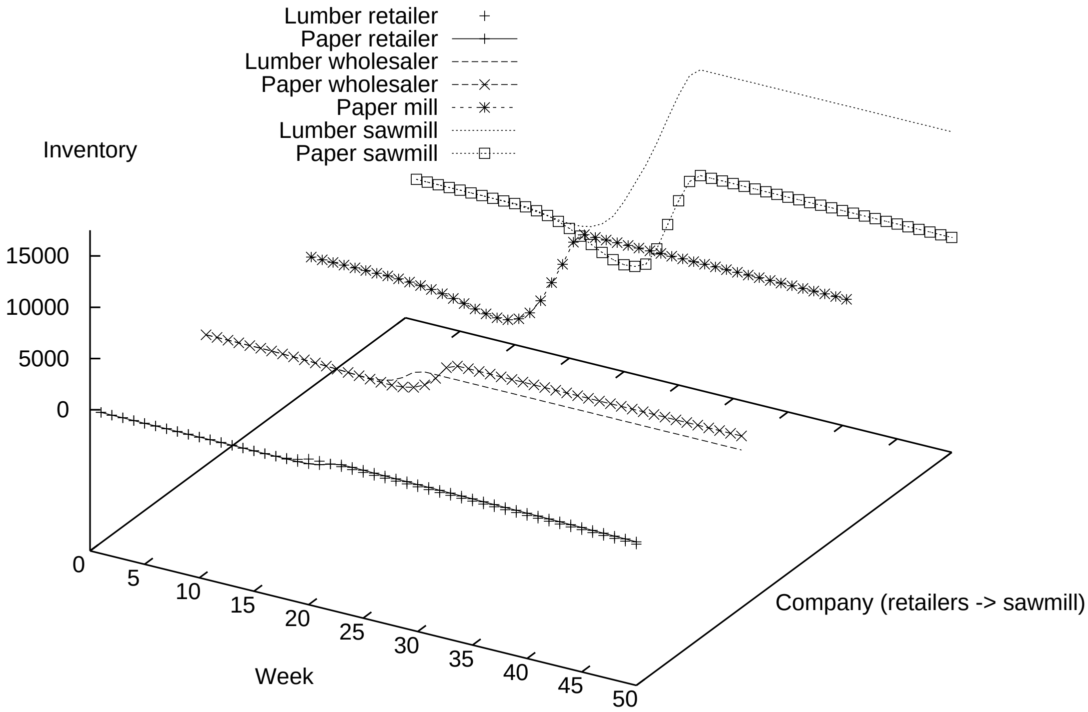


Figure 7.6: Inventories in Scheme A'' (with $O + \Theta$ orders, without information centralization).

- the demand is steady.

The first hypothesis from EOQ is not satisfied in our simulation, because backorders may occur with all suppliers, and thus, these suppliers cannot be viewed as an infinite source of products for their clients. Therefore, the assumption in EOQ of cycling operations in each inventory system, presented later in Figure B.1, does not hold. The second hypothesis, i.e., demand is steady, also does not hold, since we use the Step demand market pattern for the moment, and next, the nineteen patterns indicated in Table 7.1, and also since no companies address a steady demand to its supplier. The fact that these two assumptions are not met generate two problems, incurring at the same time:

1. *First problem:* Since the *first assumption* (suppliers may have backorders, while they are modelled in EOQ as infinite sources of products) does not hold, makes the supply chain work in a very inefficient way, or, in more general terms, *when every company optimizes its decisions locally, the overall supply chain behaviour*

is very poor.

The typical supply chain behaviour in our simulator is as follows. When the market consumption increases, each retailer has a backorder, and orders each week such as to eliminate it (they order $(S - I)$ products, where $I < 0$ is the inventory level representing their backorder). But there are not enough products by the wholesaler to fulfill this demand. Therefore, the wholesaler orders more products than the quantity ordered by the Retailer, which leads to a much bigger backorder by the next supplier. Finally, all companies overorder too much product in comparison to the market consumption. In fact, the market consumption is 17 products/week, while the Sawmill orders up to 2,632 products in Week 20!

This huge number is due to the "domino effect" that has just been outlined, and visible in the hump in orders in Figure 7.5 that are higher with the Sawmill than with the retailers: retailers increase their orders to remove their backorder, which makes wholesalers have a greater stockout than retailers. Therefore, wholesalers increase their orders more than retailers, which makes their supplier have a much greater stockout than retailers and wholesalers.

2. *Second problem:* The *second assumption* (each company's demand is steady) causes the cycling operations of inventory system to be disturbed. In fact, when the demand is steady, orders are all equal in each week: order in Week w is (i) equal to order in Week $w + 1$, and (ii) received in Week $w + 4$. On the contrary, when the demand is no longer steady, order in Week w is different from order in Week $w + 1$.

   When a stockout occurs in week w, the company orders $(S - I)$ products in weeks w, $w + 1$, $w + 2$ and $w + 3$, and also probably $w + 4$, $w + 5$, $\ldots$ to eliminate this stockout. As a consequence, the company orders too much product, because it should only order the quantity corresponding to the *stockout aggravation* between the previous and the current week, while the (s, S) policy tries to eliminate the *entire stockout* in each week.

   For example, when the inventory level is -3 in week w, and -7 in week $w + 1$, the company should order in Week w in order to eliminate the backorder of 3, and in week $w + 1$ to eliminate the aggravation of this backorder, that is, the company should order for 4 products. On the contrary, the (s, S) policy causes the company to try to eliminate a backorder of 3 in week w (which is right), and a backorder of 7 in week $w + 1$ (which is wrong). Therefore, when a stockout occurs in week w, the quantity ordered in week w is correct, but too much products are ordered in weeks $w + 1$, $w + 2$ and $w + 3$, and perhaps $w + 4$, $w + 5$, $\ldots$.

   As a consequence, companies order more and more when they have backorders, which makes their inventory level explode, when ordered products eventually arrive from the Forest. For instance, Sawmill's paper inventory fluctuates from

-3,336, i.e., a backorder of 3,336 units, to 7,198 [2] in Figure 7.6, because this company orders more than 2,000 units during six weeks in Figure 7.5, while the market consumption is only 17!

We do not compare A" with any other scheme, since it is related to no other scheme. The interest of A" is only to implement the well-known (s, S) rule, and to have a mathematical description allowing optimizing its parameters, as shown in Subsection B.2.

Simulation Outputs with Ordering Scheme B

Scheme B is the implementation of our two principles without information centralization, which thus uses (O, Θ) orders. Orders with Scheme B are presented in Figure 7.7. Peaks correspond to two concurrent events:

- Θ coming from the client are transmitted to the supplier;
- non-zero Θ are emitted, because the company becomes aware of the market consumption change at the same time as Θ are received.

This explains why PaperMill's peak is higher than LumberWholesaler's and PaperWholesaler's, which are themselves higher than LumberRetailer's and PaperRetailer's in Figure 7.7. Indeed, the following scenario occurs:

- Both retailers have a peak, because they emit Θ when the market consumption jump from eleven to seventeen.
- Two weeks later, both wholesalers receive both the new value of the market consumption in O, and non-zero Θ. The wholesalers transmit these Θ, and add to them their own Θ, because O has changed. As a consequence, the peak by the wholesalers is twice as high as the peaks by the retailers.
- Six weeks later, the Sawmill receives Θ from the LumberWholesaler, and eight weeks later from the PaperMill. Each time, the Sawmill transmits the incoming Θ, and reacts to the change in O by emitting new Θ.

---

[2] In fact, this is a decimal number. In general, decimal numbers may appear in inventory level and placed orders with all of our seven ordering schemes, because of the aggregation method of lumber and paper orders by the Sawmill: this company orders half its lumber requirements plus half its paper requirements.

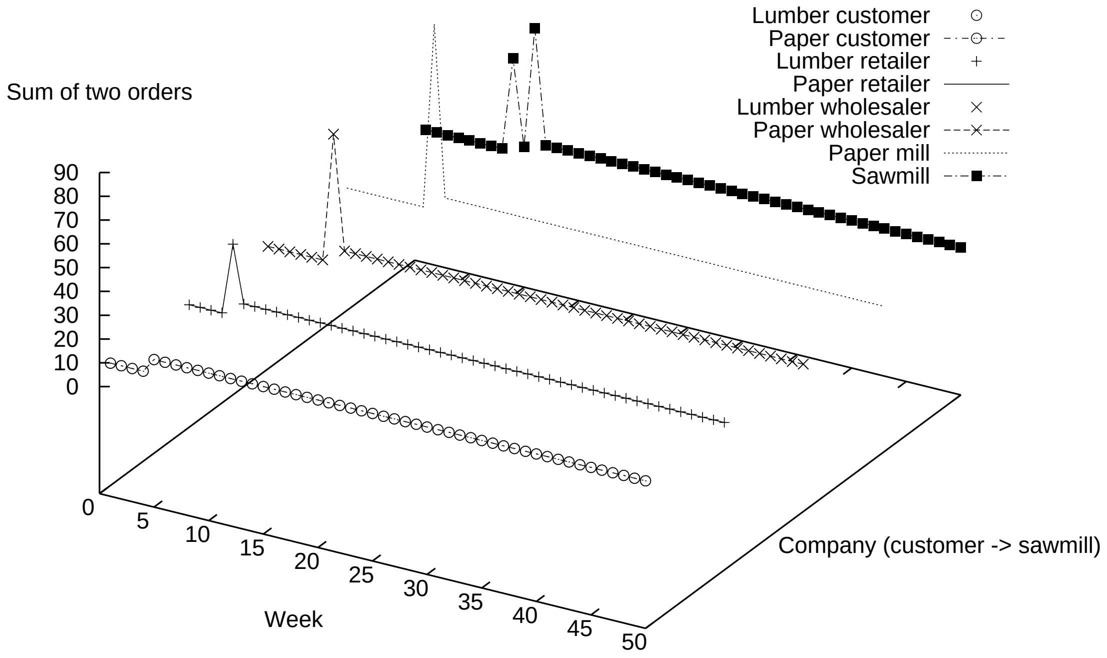


Figure 7.7: Orders in Scheme B (with (O, Θ) orders, without information centralization).

Figure 7.8 illustrates the inventory behaviour presented in Section 5.4:

- Inventory levels start decreasing when the considered company becomes aware of the market consumption increase (this increases is reflected in O). At this time, companies emit Θ to order more products, so that their inventory level increases up to its initial level.

- In general, this increase of O and the reception of non-zero Θ causes the supplier of the considered company to be in backorder. A a result, the inventory continues to decrease. However, the inventory increases sometimes with some companies, because their supplier does not incur this backorder (in particular, the inventories by both retailers in Figure 7.8 increase up to their initial level six weeks after the market consumption increases, then remains at this level for a week, before decreasing again).

- These two previous points show that the change occurs in the distribution of backorder duration in the supply chain. The backorder is very important for the

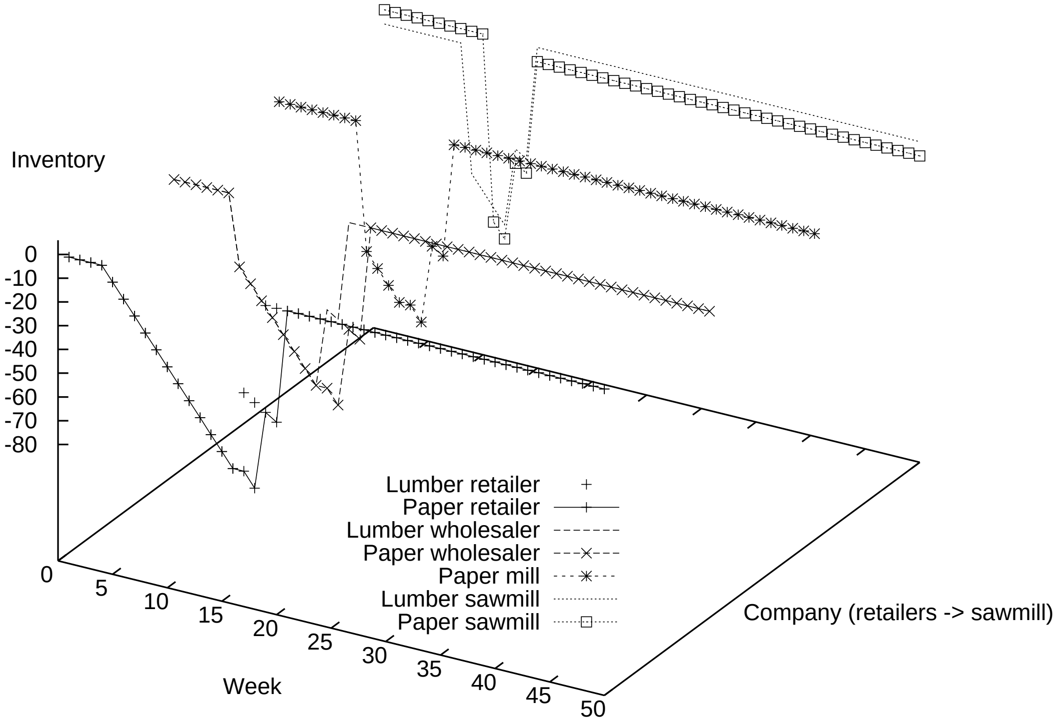


Figure 7.8: Inventories in Scheme B (with (O, Θ) orders, without information centralization).

Sawmill, but lasts a shorter duration than for the wholesalers[3] (i.e., the second curve from the bottom in Figure 7.8).

This point is one of the contributions of this thesis: *our two principles make that the inventory variations may disturb more the retailers, while the classic situation of the bullwhip effect disturbs the* Sawmill *more.*

Of course, this disturbance depends on how backorders are measured: we can refer either to the importance of the backorder, that is, how negative the inventory level is, or to the duration. In fact, a company may prefer to have a lot of unfulfilled orders for a short period (case of the Sawmill in Figure 7.8), than a few unfulfilled orders for a longer period (case of the wholesalers in Figure 7.8).

---

[3] We do not refer to the retailers here, because this is not very clear in Figure 7.8, but our presentation in Section 5.4 shows that, in theory, inventory fluctuations (and thus, possible backorders) are also longer by retailers than by the Sawmill.

Comparison of Schemes A' and B

Orders placed with Scheme A' in Figure 7.3 have a pattern very similar to orders placed with Scheme B in Figure 7.7. In fact, except for some peaks, orders placed with these two schemes are similar to the market consumption. Therefore, both schemes reduce the bullwhip effect very much. We can note that A' and B are similar in two respects:

1. Schemes A' and B both place orders only based on demand. This allows reducing the bullwhip effect, but they need some adaptations to manage inventory in real life, because they only consider incoming demand, and ignore the inventory level, while we have seen that, this level may fluctuate for any reason, such as breakage, spoilage, theft. . . .
2. Scheme B respects our second principle (only one over- or underorder as a reaction to a market consumption change), while Scheme A' is very near that, only three overorders occur (i.e., three peaks of demand) in Figure 7.3 per market consumption change.

Simulation Outputs with Ordering Scheme B'

Figures 7.9 and 7.10 presents supply chain behaviour with Scheme B'. Scheme B' has very variable orders and inventories, but less than Scheme A. In other words, there is a large bullwhip effect in Figure 7.9, which, in return induces very variable inventories in Figure 7.10.

Comparison of Schemes B and B'

Both schemes B and B' use information sharing with (O, Θ) orders, and both satisfy our first principle, i.e., the lot-for-lot ordering policy eliminates the bullwhip effect, but does not manage inventories. On the other hand, B' does not respect our second principle (only one over- or underorder as a reaction to a market consumption change), while B does.

However, orders never stabilize with Scheme B', while they do with Scheme B. This shows an important point:

> Order stabilization, i.e., the reduction of the bullwhip effect, is not only due

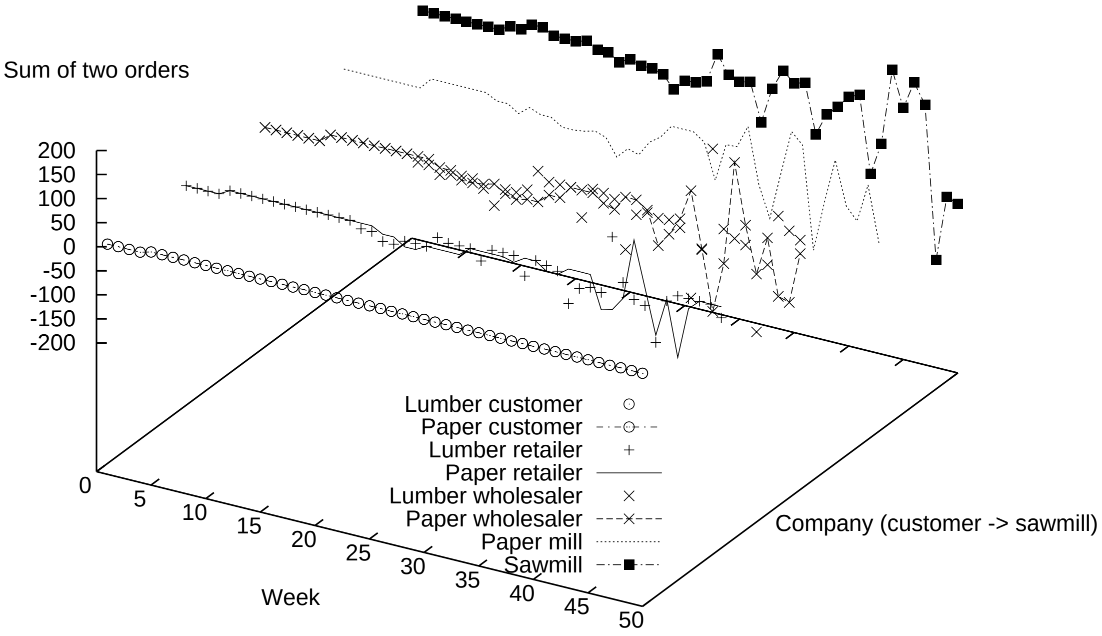


Figure 7.9: Orders in Scheme B' (with (O, Θ) orders, without information centralization).

> to information sharing with (O, Θ) orders, but also due to the two proposed principles, that should necessarily be satisfied together.

From a more general viewpoint, this means that information sharing is only a tool required to reduce the bullwhip effect, but *the main issue is how to use the shared information.*

Simulation Outputs with Ordering Scheme C

Scheme C is similar to A, except that information centralization is applied. As a consequence, companies using C base their order on the actual market consumption rather than on incoming orders. Figure 7.11 shows that Scheme C incurs the bullwhip effect, which leads in return to variability in inventory levels in Figure 7.12.

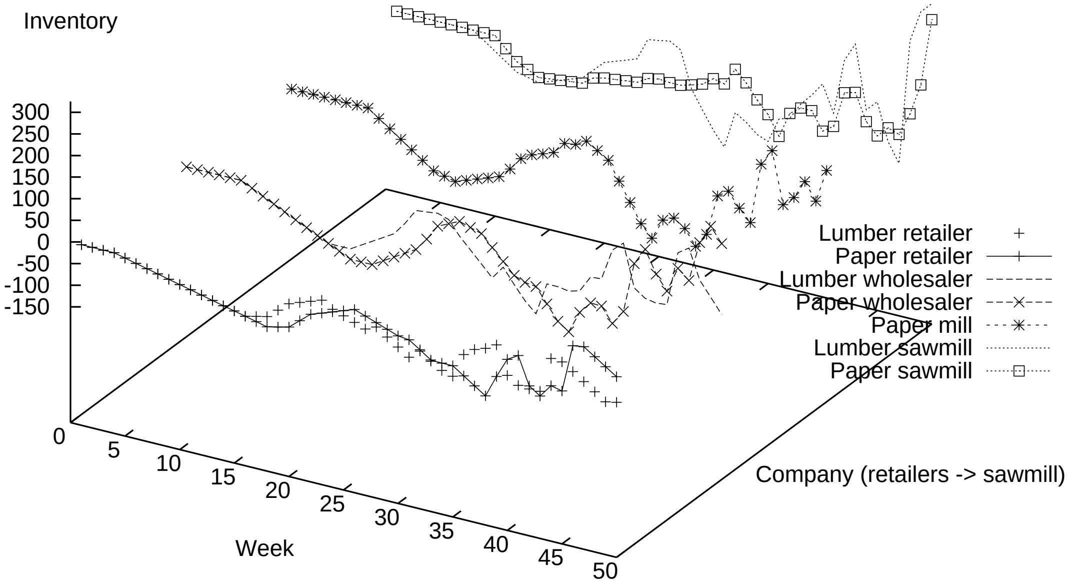


Figure 7.10: Inventories in Scheme B′ (with (O, Θ) orders, without information centralization).

Comparison of Schemes A and C

When we compare the orders placed with Scheme A in Figures 7.1, and orders placed with Scheme C in Figure 7.11, we have a confirmation that information centralization reduces the amplification of demand variability, which next leads to lower inventory in Figure 7.12. In fact, Schemes A and C only differ with the use or not of information centralization and none of them use (O, Θ) orders. In conclusion:

> Information centralization is a good strategy to reduce the bullwhip effect, as formally proven by others [Simchi-Levi et al., 2000].

Next, we obtain Scheme D from B by adding (O, Θ) orders to Scheme C.

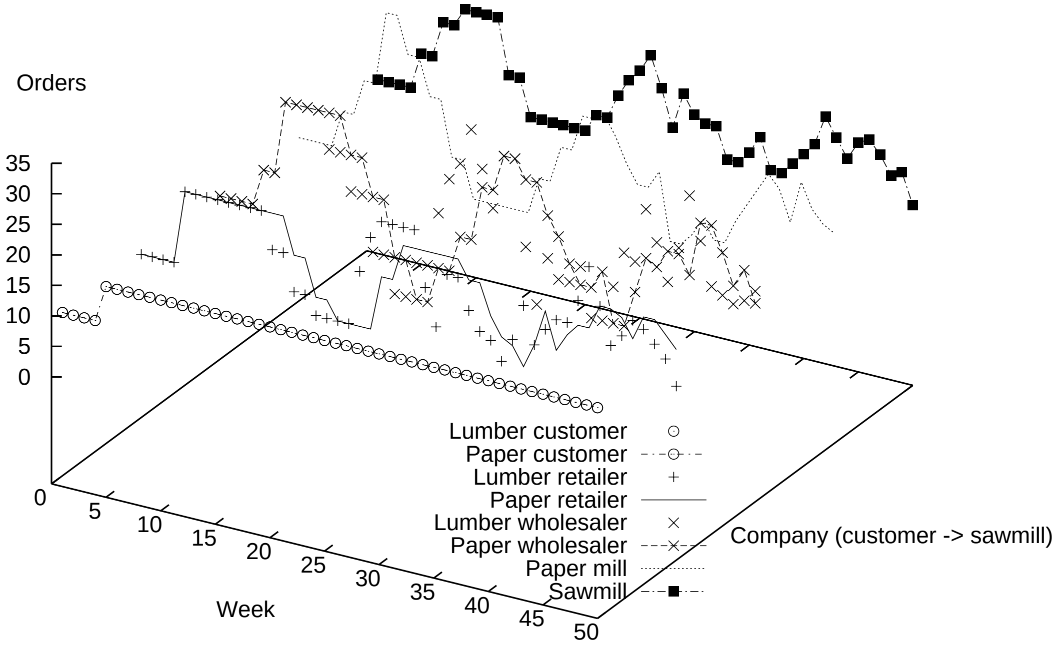


Figure 7.11: Orders in Scheme C (with $O + \Theta$ orders, with information centralization).

Simulation Outputs with Ordering Scheme D

Scheme D in Figures 7.13 and 7.14 gives results very similar to Scheme B in Figure 7.7 and 7.8: the bullwhip effect is significantly reduced, because company ordering patterns look like the market consumption except for a few peaks, as reflected in Figure 7.13. In short, the following scenario occurs in the supply chain:

- When market consumption changes, every company knows it, because of information centralization. As a consequence, every company orders both in O the new value of the market consumption, and in Θ in order to receive more products, because they know that ordering and shipping delays will cause their inventory level to decrease. These Θ correspond to the first peak by every company in Figure 7.7.
- Two weeks later, all companies, except both retailers, receives the Θ issued by their client. These companies only transmit these Θ to their supplier, without emitting new Θ, since they have already reacted to the change in the market consumption.

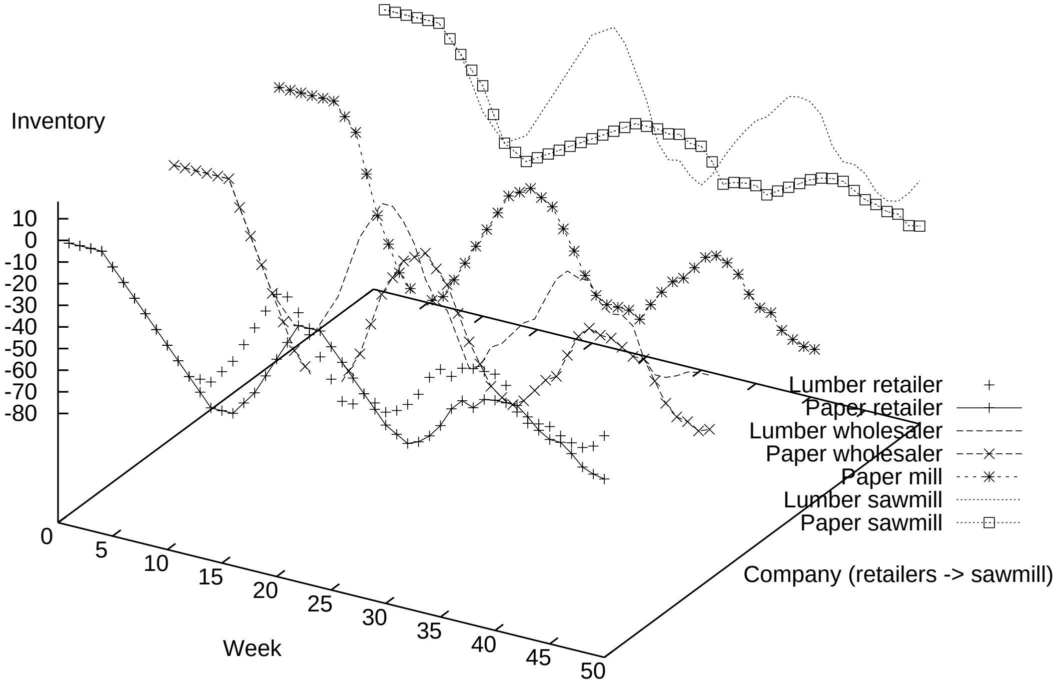


Figure 7.12: Inventories in Scheme C (with $O + \Theta$ orders, with information centralization).

- Four and six weeks later, the same events are repeated, but by fewer companies.

We can note that both retailers emit more Θ than the other companies. This is due to the fact that they sell directly to end-customers. In fact, there are two different types of position in the supply chain:

- *Retailers:* when the market consumption changes, both retailers (i) ship more product to fulfill end-customer demand, (ii) order in O the new value of the market consumption, and (iii) order in Θ additional products to stabilize their inventory on its initial level.
- *All companies, except retailers:* when the market consumption changes, every company (ii) orders in O the new value of the market consumption (which is multi-casted by the retailers), and (iii) orders in Θ additional products to stabilize their inventory on its initial level.

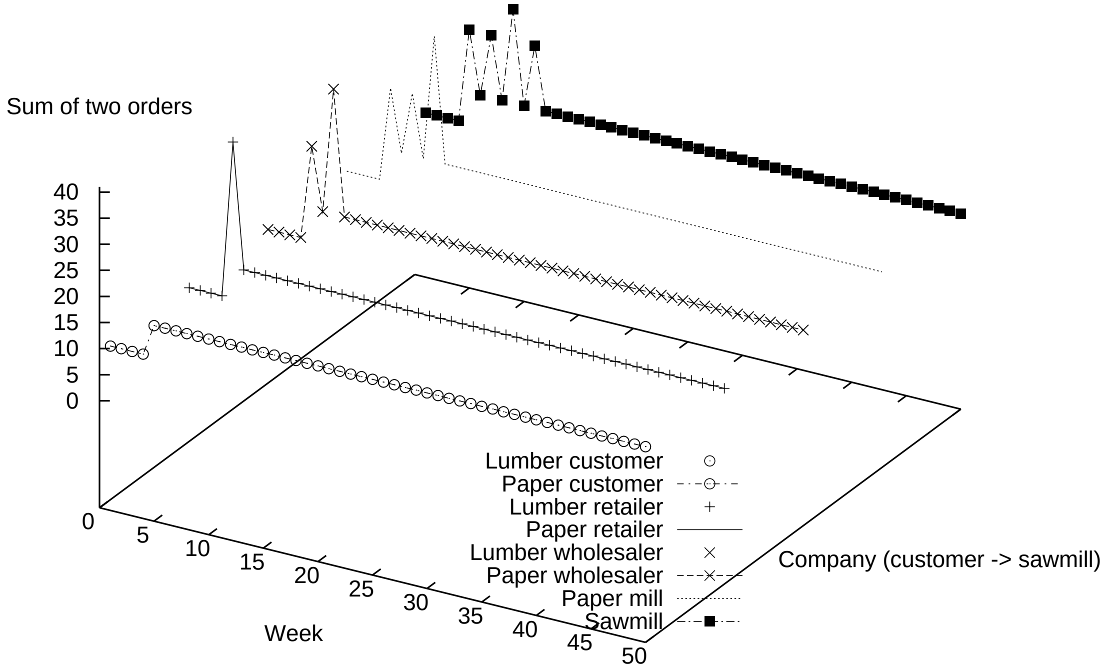


Figure 7.13: Orders in Scheme D (with (O, Θ) orders, with information centralization).

As we can see, only the retailers have the action (i) to carry out. This action causes their inventory level to decrease more than other companies' inventory. As a consequence, the parameter λ, which rules the emission of Θ, is greater by retailers than by other companies.

In conclusion, D has the best results among Schemes A, A′, A″, B, B′, C and D, because inventories fluctuate only a little in Figure 7.14, while the bullwhip effect has almost disappeared in Figure 7.13.

Comparison of Schemes B and D

Schemes B and D both implement our two principles, but D uses information centralization in addition. Inventory levels are very stable with these two schemes (cf. Figures 7.8 and 7.14). The difference between Schemes B and D is, that D applies information centralization, which allows companies to react more quickly to the change in market consumption. This is the reason why inventories decrease less with D in

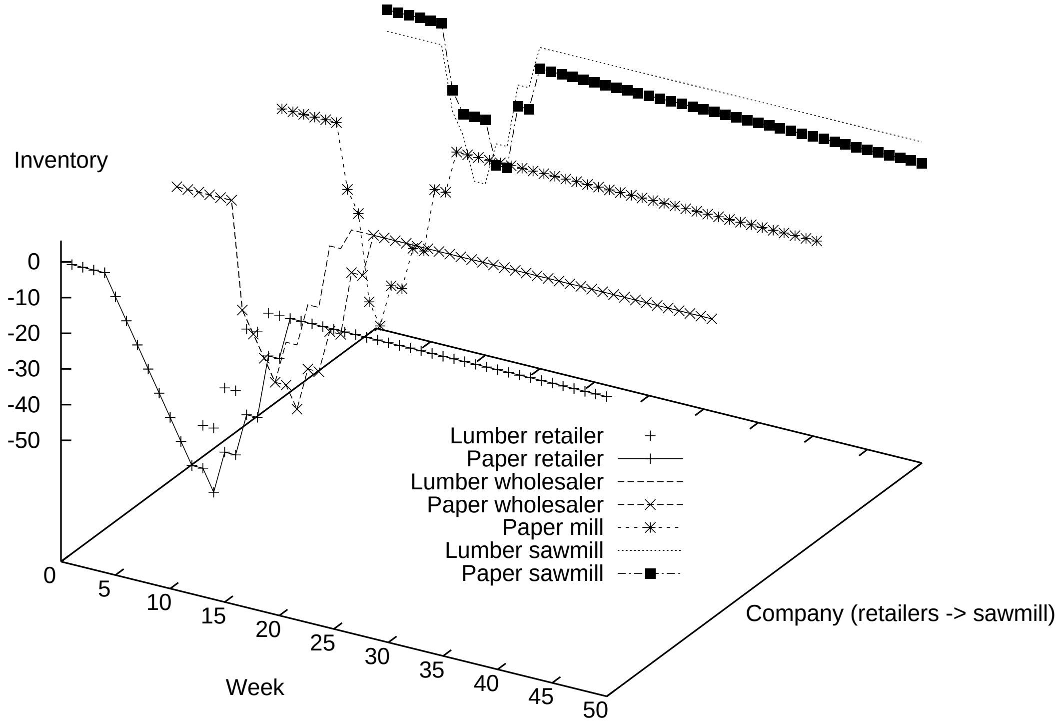


Figure 7.14: Inventories in Scheme D (with (O, Θ) orders, with information centralization).

Figure 7.14, than with B in Figure 7.8, which avoids some stockouts, and therefore improves customer service levels.

The use of information centralization is also the reason why there are more peaks but with a lower amplitude, with Scheme D in Figure 7.13, than with B in Figure 7.7: for each company using D, the first peak corresponds to the emission of Θ, and later peaks are the transmission of Θ from client to supplier.

These peaks correspond to the emission of Θ. We can note here that companies using B and the retailers using D emit the same quantity of Θ. This quantity is set by the parameter λ. In other words, retailers use the same value of λ in B and D, while the other companies only use this value when they use B. In fact, the other companies apply a lower value of λ when they use D, that is, overorders have a lower amplitude by these companies when they use D. As a consequence, the bullwhip effect is lower with D than with B, *which confirms that information centralization reduces the bullwhip effect.*

After that, when B is applied and, as announced in Section 5.4, *inventory variations* *last a longer time* for retailers than for the most upstream supplier (Sawmill), while the situation is inversed with the five other ordering schemes: inventory fluctuations are greater by the Sawmill than by Retailers in Figures 7.2, 7.4, 7.6, 7.10, and 7.12. Nevertheless, the amplitude of inventory variation remains the greatest by the Sawmill with all ordering schemes. As a consequence, depending on their position in the supply chain, and depending on the way that backorders are considered (e.g., backorders are ignored, or conversely backorders cost too much money), companies may thus prefer B than other schemes, or the contrary.

Nevertheless, when information centralization is applied, i.e., with D, backorders last four weeks for both retailers and both wholesalers, six weeks for the PaperMill, and eight weeks for the Sawmill (we can count the weeks in Figure 7.14, because each week is represented by a point). Therefore, only B has the change in the distribution of backorders that was previously stated.

Discussion

Only Schemes B and D implement our two principles. These two schemes also induce the lowest bullwhip effect, and the fewest inventory fluctuations among the seven tested ordering schemes. This shows that adopting our two principles incurs excellent results for the entire supply chain.

Next, we can question whether we can drop one of the proposed principles:

- *Principle 1:* Only B, D and B$'$ satisfy our first principle, but B$'$ incurs much worse results than B and D, which shows the necessity for our first principle.
- *Principle 2:* Only B, D and A$'$ satisfy our first principle, but A$'$ incurs much worse results than B and D, which shows the necessity for our second principle.

We recall that we have noted that only A$'$ incurs some peaks in orders, that is, the bullwhip effect lasts a very short period with A$'$. As a consequence, A$'$ should be studied further as a future work, in particular because it does not require information sharing, conversely to our two principles.

All these results were obtained with the Step market consumption pattern. We now execute our seven ordering schemes with the eighteen other market consumption patterns of Table 7.1. The following Tables 7.2, 7.3, 7.4 and 7.5 present the value of the four measures of each ordering rule efficiency for the same set of $7 * 19$ experiments (seven ordering rules × nineteen market consumption patterns). The four metrics are (i) the standard deviation of orders, (ii) the induced costs, (iii) the sum of backorders and (iv) the standard deviation of inventory levels (with backorders measured as negative inventory levels). Tables 7.2, 7.3, 7.4 and 7.5 present these data. As previously stated, these four tables are obtained with empty initial inventories ($\forall i, I_1^i = 0$). The next subsections detail these tables.

7.3.2 Comparison of the Bullwhip Effect Under the Nineteen Market Consumption Patterns

The standard deviation of orders, which is the first considered metric, is the direct quantification of the bullwhip effect. Table 7.2 summarizes this data for PaperMill. For the sake of simplicity, further details for all other companies are in Tables B.6, B.7, B.8 and B.9 in Subsection B.5.1. Each column in Table 7.2 presents the standard deviation of orders placed by the PaperMill under the nineteen considered consumption patterns.

For example, when all companies place orders with Scheme A and under the Step demand pattern, the standard deviation of PaperMill's orders is 51, while the second column in Table 7.2 says that the standard deviation of the market consumption is only 1.6. Therefore, orders placed by the PaperMill in this experiment are much more variable than the market consumption. These data have to be compared line by line. For each line, i.e., for each demand pattern, the lowest standard deviation is underlined. Since the values in Table 7.2 measure the bullwhip effect, Scheme D generally incurs the lowest bullwhip effect among the seven tested ordering schemes, except for the ten random distributions of demand. Indeed, when demand is random, there is no distinction among C and D: under this demand, the reduction of the bullwhip effect is thus due to information centralization, because information centralization is the only characteristic shared by C and D. When information centralization is not used, B incurs the lowest bullwhip effect (C is better than B, but C uses information centralization). Shortly, our two principles on which B and D are based are not contradicted, except when demand is random and information centralization is used.

Notice here that the goal of our two ordering schemes is to have a standard deviation of orders that is the nearest possible to the standard deviation of the market demand for all companies, but companies can only have a standard deviation equal or superior to the market's. Other approaches for reducing the bullwhip effect use smoothing techniques

	Market consumption	Scheme						
		A	A'	A''	B	B'	C	D
1 Step	1.6	51	105.7	547.7	10.6	42.7	7.1	4.5
2 Inversed Step	1.6	33.2	37.4	345.6	10.6	28.8	13.1	4.5
3 Dirac	0.8	18.7	111.8	90.8	15.2	17	4.4	6.2
4 Inversed Dirac	0.8	25.6	142.8	778.2	15.2	43.5	8.1	6.2
5 Increase	14.4	108.1	22.2	590.8	16.7	16.6	16.3	16
6 Decrease	14.4	27.9	20.3	1,928.2	16.7	41.7	20.6	16
7 Weak seasonality	0.7	3.6	98.4	205.5	11.6	10.6	0.9	4.2
8 Medium seasonality	1.2	25.3	73.7	38.7	11.7	15.2	4.7	3.2
9 Strong seasonality	1.8	27.1	59.3	100.7	11.7	19.8	12	6.5
10A Uniform random	1.8	30.3	275.4	576.1	32.8	37.7	10.1	13.6
10B Uniform random	1.7	39.1	206.4	644.8	25.5	38.8	11.2	10.6
10C Uniform random	1.7	36.1	215.1	43.3	28.1	41.4	9.4	11.7
10D Uniform random	1.7	35	181.8	349.2	26.6	50.1	13.5	10.2
10E Uniform random	1.9	38.9	244.3	64	31.5	51.9	9.4	13
10F Uniform random	1.8	35.3	214.4	125.9	26.8	37.6	7.1	11.9
10G Uniform random	1.9	37.9	246.1	281.8	31.5	23.1	6.7	14.7
10H Uniform random	1.9	39.5	243.6	378.6	30.5	68.9	10	11.7
10I Uniform random	1.8	36.1	263	540.1	31.3	46.8	7.3	13
10J Uniform random	1.8	28.9	261.4	754.5	28.4	33.5	8.8	12.9

Table 7.2: Standard-deviation of PaperMill's orders, i.e., $\sigma_{Op^5_w+\Theta p^5_w}$ (detailed in Tables B.6, B.7, B.8 and B.9).

which allow companies to have a standard deviation inferior to the market's. In other words, the market consumption fluctuates, but companies smooth this fluctuation and order in a more steady way. We discuss the use of such smoothing techniques to extend our work in the concluding chapter.

7.3.3 Comparison of the costs incurred under the nineteen market consumption patterns

Our second metric, the overall supply chain cost calculated with the "improved" method, is an indirect measure of the bullwhip effect reduction. Nevertheless, it is more important for companies than the standard deviation of orders, because their goal is to maximize their profit rather than reduce the bullwhip effect. In this context, Table 7.3

presents the supply chain cost C_{improved} for the the same experiments as Table 7.2.

Numbers in Table 7.3 are the costs for the entire supply chain, like in the QWSG: each unit in inventory costs each week half the cost of a backordered unit, and client's costs are (37/50)% higher than supplier's costs (cf. the definition of C^i_{improved} in Section 7.2). For the sake of simplicity, individual costs C^i_{improved} are detailed for every company in Tables B.10, B.11, B.12 and B.13 in Subsection B.5.2; we only focus on overall supply chain cost for the moment. Each line presents the overall supply chain cost under a particular demand pattern. For example, when all companies use Scheme A under the Step demand pattern, the supply chain cost is \$40,055, which is not the lowest value in this line. The lowest value is underlined. If the companies had used D, the overall cost would have been lowered to \$6,626.

We now draw some conclusions in two successive parts: we first deal with Schemes C and D, next with A, A′, A″, B and B′. First, we can note that costs with Scheme D are underlined for half of the considered demand patterns, and that costs with C are also often underlined. These two schemes use information centralization. Similarly to the first metric in the previous subsection, D is generally more efficient than C with the nine first demand patterns, while D is as efficient as C with the ten last demand patterns. This shows that results are quite obvious with demand patterns without random process (in this case, our two principles often allow incurring the lowest supply chain cost), while the use of random demand incurs fuzzy conclusions (in this case, only the information centralization explains why C and D incur the better results).

When we compare A, A′, A″, B and B′, that is, when we compare the ordering schemes that do not apply information centralization, we can see that B always incurs the lowest supply chain cost. In particular, B should be compared with B′, because both use information sharing with (O, Θ) orders: B′ incurs much higher costs that B. As a consequence, when information centralization is not applied, our two principles always incur the lowest supply chain cost.

The QWSG costs are an aggregated view of the backorders and inventory levels next measured in Tables 7.4 and 7.5. We now focus on these two metrics.

	Scheme						
	A	A'	A"	B	B'	C	D
1 Step	40,055 $	181,641 $	1,152,612 $	9,035 $	29,556 $	21,833 $	6,626 $
2 Inversed Step	37,034 $	65,593 $	576,093 $	3,237 $	17,437 $	8,426 $	2,417 $
3 Dirac	9,206 $	226,310 $	192,925 $	1,319 $	6,798 $	2,930 $	989 $
4 Inversed Dirac	14,080 $	237,074 $	1,406,648 $	1,939 $	21,373 $	3,887 $	1,369 $
5 Increase	51,681 $	31,071 $	1,130,332 $	46,754 $	95,308 $	80,770 $	32,620 $
6 Decrease	25,022 $	24,111 $	3,542,566 $	8,754 $	17,888 $	17,436 $	6,207 $
7 Weak seasonality	678 $	244,467 $	182,966 $	2,163 $	2,409 $	279 $	1,194 $
8 Medium season.	12,619 $	140,492 $	76,903 $	7,938 $	7,255 $	3,237 $	3,699 $
9 Strong season.	20,774 $	104,918 $	213,305 $	11,970 $	14,742 $	9,031 $	7,527 $
10A Uniform random	19,525 $	666,181 $	1,107,530 $	9,234 $	12,930 $	5,017 $	5,561 $
10B Uniform random	27,959 $	622,179 $	1,254,093 $	12,140 $	17,929 $	5,711 $	4,684 $
10C Uniform random	26,502 $	751,719 $	158,395 $	12,768 $	26,019 $	8,826 $	6,529 $
10D Uniform random	40,100 $	618,905 $	640,874 $	10,175 $	20,865 $	5,763 $	6,447 $
10E Uniform random	23,916 $	720,748 $	145,021 $	12,144 $	13,937 $	5,017 $	5,574 $
10F Uniform random	20,946 $	727,832 $	280,938 $	8,076 $	12,767 $	5,892 $	4,240 $
10G Uniform random	22,385 $	789,886 $	679,511 $	10,677 $	10,180 $	6,866 $	7,392 $
10H Uniform random	40,087 $	821,050 $	820,017 $	12,009 $	24,549 $	7,973 $	9,286 $
10I Uniform random	27,251 $	825,016 $	1,164,409 $	12,721 $	19,540 $	7,971 $	7,184 $
10J Uniform random	20,203 $	701,809 $	1,568,480 $	10,018 $	13,027 $	6,484 $	5,911 $

Table 7.3: Supply chain costs C_{improved} (detailed in Tables B.10, B.11, B.12 and B.13).

	Scheme						
	A	A'	A"	B	B'	C	D
1 Step	1,418	905	8,943	135	747	2,081	84
2 Inversed step	157	65	5,755	16	215	157	0
3 Dirac	246	1,021	1,461	39	186	206	45
4 Inversed Dirac	300	967	12,942	113	473	233	91
5 Increase	3,259	417	9,617	126	664	7,956	96
6 Decrease	48	0	32,178	0	70	0	0
7 Weak seasonality	43	203	3,129	143	154	22	48
8 Medium seasonality	247	107	608	374	333	206	81
9 Strong seasonality	255	189	1,643	422	556	651	164
10A Uniform random	221	233	9,257	366	581	297	262
10B Uniform random	447	247	10,616	604	542	345	115
10C Uniform random	388	473	531	343	511	632	158
10D Uniform random	118	605	5,433	360	593	179	104
10E Uniform random	623	714	6,127	393	505	374	175
10F Uniform random	437	779	1,996	321	259	474	132
10G Uniform random	549	1,323	4,999	600	551	630	369
10H Uniform random	352	772	6,451	367	1,033	393	88
10I Uniform random	807	1,638	10,221	922	966	933	287
10J Uniform random	361	217	13,596	209	229	300	104

Table 7.4: Sum of PaperMill's backorders, i.e., $\sum_w (Ob^5_w + \Theta b^5_w)$ (detailed in Tables B.14, B.15, B.16 and B.17).

7.3.4 Comparison of the Customer Service Level Under the Nineteen Market Consumption Patterns

Our third metric is the sum of backorders, which is a metric for the customer service level presented in Table 7.4 (as for the two previous metrics, further details can be found in Tables B.14, B.15, B.16 and B.17 in Subsection B.5.3). This sum has to be minimized, because when it is zero, clients have the products they want, or else they have to wait for their availability. This measure is included in costs, but we separate it for now. In fact, backorders can be avoided by overstocking, which increases costs but reduces this measure. Therefore, costs are more important than the sum of backorders, but the way of pricing a backorder depends on the manager's choice. In particular, a manager could choose that backorders cost nothing while we choose that they cost twice the inventory holding price.

Next, backordered O and Θ (Ob and Θb) are taken into account in the same way and added up on the fifty weeks of a simulation. This sum is only presented for PaperMill in Table 7.4. Other companies are detailed in Tables B.14, B.15, B.16, and B.17 in Appendix B. The data in Table 7.4 are obtained with empty initial inventories. For example, for Scheme A under the Step demand pattern, PaperMill has 1,418 delays for shipping products. This number does not mean 1,418 orders were lately fulfilled. In fact, the sum of backorders increases by one unit if an order cannot be fulfilled the first week, the sum of backorders increases by two units if an order cannot be fulfilled the first and the second weeks, etc. Again, the lowest value for each consumption pattern is underlined. Scheme D almost always has the best results, followed by B. This shows that our two principles incur good customer service levels. Scheme C still incurs good results, even if they are not as good as with the two first metrics.

7.3.5 Comparison of the Inventory Levels Variations Under the Nineteen Market Consumption Patterns

The last metric is the standard deviation of inventories (with backorders measured as negative inventory levels), which is used to choose the target inventory level. In fact, when the standard deviation of inventories increases, the target inventory level has to increase to avoid stockouts. In other words, safety inventory has to be increased. Therefore, when this measure decreases, inventory levels also decreases which reduces companies' costs (i.e., the second metrics) without increasing their sum of backorders (i.e., the third metrics). Data in Table 7.5 presents the fluctuation of PaperMill's inventory, when backorders are seen as negative inventory levels: the more these values are, the more PaperMill's inventory fluctuates. For example, under the Step demand with Scheme A, the PaperMill's standard deviation of inventory is 180.2, which is much higher than the underlined minimum, 14.4 with Scheme D. The best possible value is zero, i.e., always steady inventory and never backorders, but this is only attainable when the whole demand is perfectly known in the future, which is not possible in practice because there are always forecasting errors. In general, Scheme D has the best results followed by B, but there are several exceptions. In particular, C has sometimes better results than B.

Data for the other companies are presented in Tables B.18, B.19, B.20 and B.21 in Subsection B.5.4. In this appendix, notice that the standard deviation of inventories is greater for upstream suppliers than for retailers for all ordering schemes. This is induced by the bullwhip effect. For schemes B and D, this fact conceals the fluctuation of inventory levels for a shorter period by upstream suppliers than by retailers, as

	Scheme						
	A	A'	A"	B	B'	C	D
1 Step	180.2	324.7	4,495.5	22.8	65.8	23.9	14.4
2 Inversed step	153.6	108.5	2,766.7	20.8	42.2	24.1	9.4
3 Dirac	42.6	405.7	714.4	7.8	19.5	6.9	6
4 Inversed Dirac	54.1	441.6	6,370	11.1	69.2	12.5	6
5 Increase	185.2	30.8	4,764.8	42.2	131.3	114.3	30.5
6 Decrease	97.5	30.2	15,538.9	16	42.8	50.6	9.6
7 Weak seasonality	2	608.4	1,235.8	3.1	6.4	0.5	1.4
8 Medium seasonality	69.9	321.6	289	13.8	13.8	6.7	8.7
9 Strong seasonality	98	238.8	840.3	27.2	44.4	27.8	17.4
10A Uniform random	72.4	1,714	4,524.3	21.8	31.6	15.2	12.9
10B Uniform random	114.2	1,671	5,310.2	22.4	57.1	16.3	8.3
10C Uniform random	110.2	1,979.7	194	20.9	95.5	17.4	9.7
10D Uniform random	137	1,303.9	2,713.4	23.9	63	23.2	10.7
10E Uniform random	96	1,894.2	436.1	23.2	43.9	11.2	10.4
10F Uniform random	101.6	1,672.7	994.8	17.4	51.6	15.1	10.2
10G Uniform random	107.6	1,771.9	2,478.1	22.5	25.6	8	11.2
10H Uniform random	124.7	2,045.9	3,346.4	23.1	69.6	16.5	11.1
10I Uniform random	134.1	1,787.3	4,318.6	22.3	50.2	10.6	11.3
10J Uniform random	99	1,902.5	6,753.4	20.9	42.2	11	10.4

Table 7.5: Standard-deviation of PaperMill's inventories and backorders, i.e., $\sigma_{I^5+Ob^5+\Theta b^5}$ (detail in Tables B.18, B.19, B.20 and B.21).

explained in Section 5.4. In fact, the upstream suppliers' standard deviation is bigger because their fluctuation is greater, which is not true for the other ordering schemes.

7.3.6 Synthesis of the Results with the Four Metrics

In conclusion, Tables 7.2, 7.3, 7.4 and 7.5 show that Scheme D is almost always the best choice for all companies to reduce the bullwhip effect when the demand is not random. When the demand is random, D has no obvious advantage over C, which shows that information centralization is the main reason of good results under random demand. On the contrary, when information centralization is not allowed, B has the better results, which shows the strengh of our two principles.

7.4 Comments

Our experimental results conform to our predictions described in Chapter 5: the bullwhip effect is greatly reduced and inventory variations are longer for upstream suppliers than for retailers. Moreover, our two ordering schemes B and D actually follow the two proposed principles presented in Subsection 5.2, i.e., lot-for-lot orders eliminate the bullwhip effect, and companies should react only once to each market consumption change. Since this reaction has to be correctly interpreted by its supplier, information sharing based on (O, Θ) orders permits this supplier to distinguish between the market change (visible in O) and inventory fluctuation (i.e., the consequence of the change of O which is visible in Θ). This stabilizes the whole supply chain at the same level, i.e., at the actual market consumption. If we aggregate O and Θ under the form of $X = O + \Theta$, the level O at which companies have to stabilize their orders is lost for upstream companies because they do not know which part is required by the market and which part reflects inventory variations induced by delays, i.e., a cause of the bullwhip effect. Sharing information about market consumption is a way to align companies behaviour to the same goal: to deliver products to clients, and Θ are only required by supply chain dynamics.

From a more general point of view, companies may use Θ in different ways: to reduce or increase their inventory, companies may emit non-zero Θ (*not anytime, but according to our two principles*) that will not be interpreted by suppliers as market consumption changes, to balance scrappings due to production contingency, etc.

(O, Θ) orders with information centralization is a special (and improved) case of (O, Θ) orders, because both allow every company to know the actual market consumption. In fact, when (O, Θ) orders are used without information centralization, order O is equal to market consumption, which is the piece of information broadcasted by information centralization. The first difference between these two systems lies in market consumption propagation speed, information O is as slow as orders while information centralization supposes each company knows in real-time the market consumption demand (retailers multicast the market consumption to the whole supply chain). The second difference between (O, Θ) orders and information centralization is the fact that *information centralization requires a stable supply chain structure to allow retailers to multicast* market consumption to the whole supply chain, while (O, Θ) orders use a *more decentralized* approach.

As stated several times, we could try to reduce the bullwhip effect with smoothed orders by placing orders based on forecasts. In our experiments, this would make the stabilization period longer. For example, with the classic QWSG market consumption (i.e., a steady market, except in Week five when there is a change that remains), companies would overorder less than in Scheme B, but this overordering would be longer and during this period, companies would not know what the market consumption is. In other words, instead of having only one peak by company for each market change, there would be a less high plateau than the peak but during a longer period. However, this does not contradict our two principles: there is only one reaction per market change, but this reaction is only different.

For industrial practitioners, the lesson of this discussion is that they have to transmit consumption by retailers when they place orders to their suppliers. That is, $X(= O+\Theta)$ represents the order they would normally place (e.g., this value is given by their (s, S) policy, but care has to be taken to satisfy our two principles), but also the market consumption O seen by retailers. This information allows their suppliers to have a better understanding of the dynamics of the supply chain. *Collaboration is therefore very important to reduce the bullwhip effect.*

Finally, one crucial result is the change in the distribution of inventory variation between the current bullwhip effect and the supply chain behaviour induced by Ordering Scheme B and D. In fact, the bullwhip effect disturbs upstream suppliers (Forest) more than retailers because orders are more stable at the market end and order restabilization comes from the market. On the contrary, with our Ordering Schemes B and D, there is a fewer number of order fluctuations and what disturbs companies now are inventory fluctuations. The companies most disturbed by the bullwhip effect are the ones disturbed for the shortest period with Ordering Scheme B and D. In fact, we stated at the end of Subsection 5.4 that upstream suppliers are less disturbed with inventory variations than retailers, because these variations last longer with retailers than by upstream suppliers, and inventory restabilization comes from the most upstream company. As Scheme D has better results than Scheme B, information centralization improves Ordering Scheme B efficiency. Fortunately, retailers also have the greatest incentive to have this information centralization because of the change in the distribution of inventory variation, and fortunately, retailers are the companies that provide information centralization by multicasting in real-time the actual market consumption. More precisely, *when* (O, Θ) *orders are used, retailers should insist for multicasting the market consumption.* We check these kinds of propositions in Section 8.4. But before doing so, we adapt our simulation to the Québec forest industry. This concludes this first series of experiments.

7.5 Conclusion

This chapter has presented the first series of experiments, in which we verify that the two principles, that we proposed in Chapter 5, actually reduce the bullwhip effect. To this end, we have also proposed two ordering schemes based on these two principles. In this chapter, we compared their efficiency with five other ordering schemes. Such a comparison was repeated for each of the nineteen market consumption patterns, that we consider in this dissertation. These comparisons were made following four metrics, which are the direct measure of the bullwhip effect, the logistics costs incurred by the bullwhip effect, the backorders incurred, and the average inventory level measured as the variation of inventory levels.

Three main points appear in these comparisons. First, when our two principles are used at the same time, the bullwhip effect is much reduced, and inventories are managed efficiently by every company, which validates our two principles. We have also noted, that both principles are required, and not only one. Second, information sharing is required to reduce the bullwhip effect, but companies also have to know how to use the shared information. A part of this knowledge is provided by our two principles, even if these principles do not explain the efficiency of Scheme C. Third, demand-based ordering schemes perform better than schemes that take into account both demand and inventory level. In particular, we have noted that one demand-based scheme (i.e., A′) could be improved as a future work, because it induces the bullwhip effect over a very short period. However, demand-based schemes are not efficient in practice, because they ignore inventory levels. One may object that our two schemes are also only based on the demand. This is true for these two schemes, but our two principles give companies the freedom to order according to their inventory level. In fact, our second principle allows over- or underordering, when there is a market consumption change, so that inventories are well managed. Of course, defining what is such a change in practice should also be investigated.

We can extend this study by stating that our two principles, and thus Schemes B and D, change the dynamics in the supply chain. Some companies may prefer one dynamic of the supply chain, while others may prefer the dynamics incurred by another scheme. As a consequence, we can wonder which ordering schemes are preferred by each company. Moreover, there may be some incompatibilities between these individual preferences. For example, if the Sawmill likes information centralization, while the retailer does not like it, the Sawmill cannot force the retailer to achieve it. This is the kind of question addressed in the next chapter.

Chapter 8

Second Series of Experiments: Heterogeneous Supply Chains

The first series of experiments, presented in Chapter 7, has assumed that the supply chain was homogeneous, that is, every company used the same ordering scheme, and the question was to determine if our two ordering schemes B and D reduce the bullwhip effect more than the five other schemes. The second series of experiments now checks if each company has individual incentives to use one of our two ordering schemes B and D. In fact, it does not because an ordering scheme incurs the minimum of the overall supply chain cost that this reduction is equally reflected in each company, because a company may have a higher cost to reduce other company's costs. Moreover, each company could prefer the whole supply chain cooperates by using B or D, except itself. In this case, the company would profit from others' effort, without participating in this effort. To address such questions of incentives, we no longer focus on the reduction of the bullwhip effect, as we did in Chapter 7, but we are interested in the money saved by this reduction. The methodology to achieve this, is described in Section 8.1.

Since we consider costs of companies in this chapter, it is important to make them meaningful. As a consequence, we adapt the method of cost calculation, introduced in Section 7.2, to realistic costs resembling the Québec forest industry. Then, we optimize ordering parameters, in order to minimize the overall supply chain cost calculated with this method. These settings of costs and ordering parameters are explained in Section 8.2.

Next, the concepts from game theory, that were presented in Section 4.2, are adapted to our model, and illustrated in Section 8.3.

Since the context of this second series of experiments is clarified, we finally present this series in an iterative way. We successively assume that:

1. the supply chain behaves like a single company: it tries to minimize its costs. In other words, every company belongs to the same player-coalition, who forces every company to use the same ordering scheme, which is exactly the case of the homogeneous supply chain studied in Chapter 7;
2. the PaperRetailer no longer belongs to this coalition. Therefore, we have two players: the PaperRetailer decides for itself, and the RestOfTheSupplyChain forces the five other companies to use the same ordering scheme;
3. three players control the supply chain: the LumberRetailer decides alone, the PaperRetailer also decides for itself, and the RestOfTheSupplyChain forces the four other companies to use the same ordering scheme;
4. each of the six companies in the Québec Wood Supply Game (QWSG) is controlled by a different player.

In these four cases (where each subsection is presented in a subsection), simulation outputs are analyzed according to concepts from game theory. These experiments, their analysis and some comments are presented in Section 8.4.

8.1 Description and Computational Complexity of our Methodology

The basic idea of our methodology is to simulate all combinations of the three ordering schemes A″, B and D among the six companies, and to analyze these simulations with concepts from game theory. In comparison with the previous chapter, we keep A″, because it is an (s, S) ordering policy, which is classic in inventory management, and B and D, because they follow our two principles to reduce the bullwhip effect, while we drop A, A′, B′ and C, because they were only used to understand the two principles on which B and D have been designed. For each of the first nine market consumption patterns[1] in Table 7.1, i.e., Step, Inversed step, Dirac, Inversed Dirac, Increase, Decrease,

---

[1] We drop the ten Uniform random market patterns, because we are not able to determine the optimal ordering schemes' parameters in a clean way in Section 8.2, i.e., each pattern 10A to 10J would have its own optimized parameters, instead of the same parameters for all random demand patterns.

Algorithm 8.1 methodology in the second series of experiments

```
returns orders and inventory level for each company and each week
for each of the nine first market consumption patterns $m$ in Table 7.1 do
    optimizeInventories($m$)
    simulateSupplyChain($m$)
    analyzeSimulation($m$)
```

and Weak, Medium and Strong seasonalities, we carry out the three steps outlined in Algorithm 8.1, and described now:

optimizeInventories(): This first step is outlined in Algorithm 8.2, presented in Section 8.2, and detailed in Sections B.3 and B.4 in Appendix C. This step aims at obtaining the best value of initial inventory levels $\{I_1^i\}_{i=1}^N$, when the supply chain is homogeneous. In fact, initial conditions are very important for Schemes B and D, because they are designed so that inventory levels eventually stabilize on their initial level after a demand change. With such optimization, inventory levels with D are lower than with B, and thus, this optimization allows benefiting from the information centralization in D.

Algorithm 8.2 describes how we have every company use Scheme B, and we let the Solver find the value of every I_1^i that minimizes the overall supply chain cost C. We call $\{I_1^{i-B}\}_{i=1}^N$ these optimal parameters. Next, we repeat these operations when every company applies Scheme D instead of B in order to get $\{I_1^{i-D}\}_{i=1}^N$. Note that we do not consider Scheme A", because it was set up in Chapter 7, according to the mathematical model called Economic Order Quantity, and we have found that the optimal initial inventories are $\{I_1^{i-A''}\}_{i=1}^N = 0$ for every company i.

As we have just mentioned, we have used the Solver included in Microsoft Excel 2000 to find the sets $\{I_0^{i-B}\}_{i=1}^N$ and $\{I_0^{i-D}\}_{i=1}^N$ minimizing C. According to Microsoft Corp. [2004b], the Solver uses the Generalized Reduced Gradient (GRG2) algorithm developed by Leon Lasdon and Allan Waren [Lasdon et al., 1978]. We have not found the complexity of this algorithm, but only some comparisons of the computation time with some other algorithms.

However, we now give an estimation of the time required by the step **optimizeInventories()**. In fact, $2*9 = 18$ of such optimizations are carried out in this chapter, because we optimize inventories, when the supply chain is homogeneous and uses either B or D, and under 9 market consumption patterns. Each optimization takes a total time of around 40 seconds on our 2 Ghz PC, that is $9 * 40 = 360$ seconds, or 6 minutes for all runs of **optimizeInventories()**.

simulateSupplyChain(): This step is oulined in Algorithm 8.3. During $3^6 = 729$

Algorithm 8.2 optimizeInventories(m)

```
returns $\{I_0^{i-B}\}_{i=1}^{6}$ and $\{I_0^{i-D}\}_{i=1}^{6}$
set LumberRetailer to $r^1$ =B
set PaperRetailer to $r^2$ =B
set LumberWholesaler to $r^3$ =B
set PaperWholesaler to $r^4$ =B
set PaperMill to $r^5$ =B
set Sawmill to $r^6$ =B
optimize $\{I_0^i\}_{i=1}^{6}$ with the Solver, such as $C_{\text{realistic}}$ is minimized under the consumption $m$
save the optimal values $\{I_0^{i-B}\}_{i=1}^{6}$
set LumberRetailer to $r^1$ =D
set PaperRetailer to $r^2$ =D
set LumberWholesaler to $r^3$ =D
set PaperWholesaler to $r^4$ =D
set PaperMill to $r^5$ =D
set Sawmill to $r^6$ =D
optimize $\{I_0^i\}_{i=1}^{6}$ with the Solver, such as $C_{\text{realistic}}$ is minimized under the consumption $m$
save $\{I_0^{i-D}\}_{i=1}^{6}$
```

simulations, Excel computes the values of the six individual costs $\{C^i\}_{i=1}^{N}$ for all combinations of the three ordering schemes among the six companies, where each simulation starts with the initial inventories $\{I_0^{i-A''}\}_{i=1}^{N}$, $\{I_0^{i-B}\}_{i=1}^{N}$ or $\{I_0^{i-D}\}_{i=1}^{N}$ found by optimizeInventories(). Each simulation produces a set of costs $\{C^i\}_{i=1}^{N}$.

We can note here, that the sets $\{I_1^{i-A''}\}_{i=1}^{N}$, $\{I_1^{i-B}\}_{i=1}^{N}$ and $\{I_1^{i-D}\}_{i=1}^{N}$, found in **optimizeInventories()**, minimize C, when every company uses the same ordering scheme, while this is not the case in our $3^6 = 729$ simulations, because companies are allowed to use different ordering schemes (heterogeneous supply chain). As a result, homogeneous supply chains are favored in comparison with heterogeneous supply chains, because the optimization carried out in **optimizeInventories()** were made with homogeneous supply chains. However, we will see in our results in Subsection 8.4.2, that heterogeneous supply chains may also incur the lowest value of C.

As future work, we should correct this point by moving **optimizeInventories()** into **simulateSupplyChain()**, as illustrated by **simulateSupplyChain'()** in Algorithm 8.4. Algorithm 8.4 shows how to optimize before each simulation, and thus, also before the simulation of heterogeneous supply chains. Since more optimizations are carried out, the computation time increases a lot!

Algorithm 8.3 simulateSupplyChain(m)

```
returns sets of simulated costs $\{C^i\}_{i=1}^6$ and $C$
for each $r^1 \in$ {A'', B, D} do
  set LumberRetailer to use $r^1$ with $I_0^1 =$ either 0, $I_0^{1-B}$ or $I_0^{1-D}$
  for each $r^2 \in$ {A'', B, D} do
    set PaperRetailer to use $r^2$ with $I_0^2 =$ either 0, $I_0^{2-B}$ or $I_0^{2-D}$
    for each $r^3 \in$ {A'', B, D} do
      set LumberWholesaler to use $r^3$ with $I_0^3 =$ either 0, $I_0^{3-B}$ or $I_0^{3-D}$
      for each $r^4 \in$ {A'', B, D} do
        set PaperWholesaler to use $r^4$ with $I_0^4 =$ either 0, $I_0^{4-B}$ or $I_0^{4-D}$
        for each $r^5 \in$ {A'', B, D} do
          set PaperMill to use $r^5$ with $I_0^5 =$ either 0, $I_0^{5-B}$ or $I_0^{5-D}$
          for each $r^6 \in$ {A'', B, D} do
            set Sawmill to use $r^6$ with $I_0^6 =$ either 0, $I_0^{6-B}$ or $I_0^{6-D}$
            simulate the supply chain under the consumption $m$
            save the simulated value of $\{C^i\}_{i=1}^6$ and $C$
```

Let us evaluate the computation time for **simulateSupplyChain()** in Algorithm 8.3. Similar to the rest of this dissertation, we call N the number of agents, and $|R^i|$ the number of ordering rules available to agent i. $|R^1| \times |R^2| \times \ldots \times |R^N|$ simulations are carried out to find the values of functions C^i. Since the duration of each simulation depends linearly on the simulation duration Ds (this is the number of simulation weeks) and on the number of simulated companies N, the complexity order of **simulateSupplyChain()** is $O(N \times Ds \times \{\max_i(|R^i|)\}^N)$. For example, in this section, $N = 6$, $Ds = 50$ and $\max_i(|R^i|) = 3$, because $|R^i| = 3$ for all agent i, and thus, the complexity of our simulations is $O(6 \times 50 \times 3^6) = O(218, 700)$. If we consider a supply chain model with additional companies, e.g., when there are several suppliers for each wholesaler, N may take greater values, and the problem thus becomes intractable.

In practice, a macro written in the language Visual Basic for Applications (VBA), included in Excel, makes the $3^6 = 729$ changes of the 3 ordering schemes among the 6 company-agents. Each run of this macro generates the $6 * 3^6$ individual costs needed to build the game corresponding to a particular market consumption pattern. Each run of this macro takes around 6 minutes. Since 9 consumption patterns are considered in this chapter, a total of around $9 * 6 = 54$ minutes is required to run the simulations for the nine consumption patterns.

analyzeSimulation(): This step is outlined in Algorithm 8.5, and its results are presented in Section 8.4. After each run of **simulateSupplyChain()**, the obtained

Algorithm 8.4 simulateSupplyChain'(m) (one possible improvement on Algorithms 8.2 and 8.3).

```
returns sets of simulated costs $\{C^i\}_{i=1}^6$ and $C$
for each $r^1 \in \{A'', B, D\}$ do
  for each $r^2 \in \{A'', B, D\}$ do
    for each $r^3 \in \{A'', B, D\}$ do
      for each $r^4 \in \{A'', B, D\}$ do
        for each $r^5 \in \{A'', B, D\}$ do
          for each $r^6 \in \{A'', B, D\}$ do
            set LumberRetailer to use $r^1$
            set PaperRetailer to use $r^2$
            set LumberWholesaler to use $r^3$
            set PaperWholesaler to use $r^4$
            set PaperMill to use $r^5$
            set Sawmill to use $r^6$
            optimize $\{I_0^i\}_{i=1}^6$ with the Solver, such as $C_{\text{realistic}}$ is minimized under the consumption $m$
            save the values of $\{C^i\}_{i=1}^6$ and $C$
```

costs $\{C^i_{i=1}\}^N$ and C are copied in a new sheet of the spreadsheet (i.e., not in the sheet of the simulator), which collects simulation outputs. Therefore, we only have to ask the minimum of all C to Excel to get the minimum supply chain cost.

Then, $\{C^i_{i=1}\}^N$ are multiplied by -1, and copied into a Gambit file. Gambit opens this file, and looks therein iteratively for strictly dominated strategies, which are saved in any document (e.g., the draft of an article). Finally, Gambit looks for every pure Nash equilibria in the remaining strategies, and we also save them.

Let us now consider the time required by **analyzeSimulation()**. As pointed out in Section 4.3, we do not know the complexity of finding Nash equilibria, but we noted that eliminating first strictly dominated strategies, much reduces the computation time without losing any Nash equilibria [McKelvey et al., 2004]. In fact, this elimination is almost instantaneous on our computer, while the computation of Nash equilibria in a complete game takes some minutes (this time is quite variable: between 40 seconds to 3 minutes). In total, the computation of Nash equilibria with the nine consumption patterns, takes around 25 minutes.

Finally, the methodology in Algorithm 8.1 takes around 6 minutes for **optimizeInventories()**, plus 54 minutes for **simulateSupplyChain()**, plus 25 minutes for **analyzeSimulation()**, that is around 1 hour and a half of pure computation time. A much

Algorithm 8.5 analyzeSimulation(m)

```
returns sets of dominated strategies, Nash equilibria and minima of $C$
copy the $3^6$ values of $C$ in an Excel sheet
apply the function min() to these $C$
save the minima of $C$
copy the $3^6$ sets $\{C^i\}_{i=1}^6$ in a Gambit file
add a minus sign in front of these $\{C^i\}_{i=1}^6$ in the Gambit file
compute iteratively all strongly dominated strategies with Gambit
save dominated strategies
apply Gambit method EnumPureSolve to find all Nash equilibria
save Nash equilibria
```

longer time is ignored here, which is the time to write macros in Excel that switches each company from an ordering scheme to another, to convert data from Excel to Gambit, to copy Nash equilibria and dominated strategies from Gambit to a file, and so on.

8.2 Realistic Costs and Optimization of Initial Inventory Levels

In this section, we adapt to the Québec wood industry the "improved" costs introduced in Section 7.2, and we optimize parameters in the three ordering schemes, so that the supply chain cost is minimum. This optimization outlines **optimizeInventories()** in Algorithm 8.2. Next, we look for optimal values of parameters in Schemes B and D. Indeed, we look for the value of λ stabilizing the inventory level on its initial level, next we seek every company's initial inventory, that reduces the overall supply chain cost.

8.2.1 Realistic costs

Conversely to Chapter 7 and to our previous papers [Moyaux et al., 2004c, 2003a,b], we adapt cost parameters from the Québec wood industry to the cost function in the QWSG. More precisely, the calculation of company i's cost C^i is adapted from the method in the QWSG, in which it is the sum of company i's inventory plus twice the sum of its backorders during the whole simulation, that is, $C^i_{\text{QWSG}} = \sum_{w=1}^{50}\{I^i_w + 2 * (Ob^i_w + \Theta b^i_w)\}$. An annual factor of 37% was added to this cost in Subsection 7.2, such as $C^i_{\text{improved}} = (0.37/50+1)^\alpha * C^i_{\text{QWSG}}$, where $\alpha = 0, 1$ or 2, depending on the considered company i. This "improved" cost is the basis that is next translated into "realistic" cost in the following way.

Firstly, one of two conversion factors is applied:

- CF^{lumber} translates lumber units in the QWSG into quantity of lumber measured in Mpmp, that is, thousand pmp -in French "pied mesure planche"- where 1 pmp is a 1 inch × 1 foot × 1 foot piece of wood,
- CF^{paper} translates paper units in the QWSG into quantity of paper measured in tma, that is, anhydrous metric ton, which represents between 2.5 and 2.8 m^3, depending on the type of wood.

Secondly, we consider a Sawmill that processes 70,000 Mpmp/year of lumber, which also represents 63,000 tma/year of paper, because there is a transformation ratio of 0.9 tma/Mpmp[2]. This corresponds to a classic Sawmill in Québec [Kruger, 2004].

Next, we calibrate the two market demands, so that they represent 70,000 Mpmp/year of lumber and 63,000 tma/year of paper, which corresponds to the capacity of the Sawmill that we consider, and thus, of our supply chain, but may not reflect the demand on a real market. In order to transform every consumption pattern in Table 7.1 into these two quantities, we use the two conversion factors $CF^{\text{lumber}} = 70,000/\sum_{w=1}^{50} D_w^{\text{lumber}}$ and $CF^{\text{paper}} = 63,000/\sum_{w=1}^{50} D_w^{\text{paper}}$. For example, with the Step market consumption pattern, $\sum_{w=1}^{50} D_w^{\text{lumber}} = \sum_{w=1}^{50} D_w^{\text{paper}} = 11+11+11+11+17+17+17+\ldots+17+17 = 11*4+17*46 = 826$ units, and thus $CF^{\text{lumber}} = 70,000/826 = 85$ Mpmp/units and $CF^{\text{paper}} = 63,000/826 = 76$ tma/units, where units are the products simulated in the QWSG.

Thirdly, we use real sale prices to convert these quantities into Canadian dollars (CAD), by applying the prices $P^{\text{lumber}} = 430$ CAD/Mpmp, $P^{\text{6-paper}} = 125$ CAD/tma and $P^{\text{5-paper}} = 690$ CAD/tma (paper has two prices, because the PaperMill transforms it, and thus increases its value). We have taken these parameters from Pribec [Conseil de l'Industrie Forestière du Québec, 2004][3]. Finally, we apply a ratio of 0.37/50 that represents logistic costs (37% per year, (37/50)% per week because our year counts fifty weeks) according to Nahmias [1997], like in improved costs in Chaper 7. This leads

---

[2] This also represents an input of 315,000 m^3/year of wood coming from the Forest, because there is a transformation ratio of 4.5 m^3/Mpmp, but we do not use this information in our simulation.

[3] I thank Martin Cloutier (CRIQ -Québec Industrial Research Center- and Master's student in For@c), who kindly gave me these parameters.

us to use the costs in Equations 8.1, 8.2, 8.3, 8.4, 8.5 and 8.6 in this chapter.

$$C^1_{\text{realistic}} = CF^{\text{lumber}} * P^{\text{lumber}} * \frac{0.37}{50} * (1 + \frac{0.37}{50})^2 * \sum_{w=1}^{50} \{I^1_w + 2 * (Ob^1_w + \Theta b^1_w)\} \quad (8.1)$$

$$C^2_{\text{realistic}} = CF^{\text{paper}} * P^{\text{5-paper}} * \frac{0.37}{50} * (1 + \frac{0.37}{50})^2 * \sum_{w=1}^{50} \{I^2_w + 2 * (Ob^2_w + \Theta b^2_w)\} \quad (8.2)$$

$$C^3_{\text{realistic}} = CF^{\text{lumber}} * P^{\text{lumber}} * \frac{0.37}{50} * (1 + \frac{0.37}{50}) * \sum_{w=1}^{50} \{I^3_w + 2 * (Ob^3_w + \Theta b^3_w)\} \quad (8.3)$$

$$C^4_{\text{realistic}} = CF^{\text{paper}} * P^{\text{5-paper}} * \frac{0.37}{50} * (1 + \frac{0.37}{50}) * \sum_{w=1}^{50} \{I^4_w + 2 * (Ob^4_w + \Theta b^4_w)\} \quad (8.4)$$

$$C^5_{\text{realistic}} = CF^{\text{paper}} * P^{\text{5-paper}} * \frac{0.37}{50} * \sum_{w=1}^{50} \{I^5_w + 2 * (Ob^5_w + \Theta b^5_w)\} \quad (8.5)$$

$$\begin{aligned} C^6_{\text{realistic}} &= CF^{\text{lumber}} * P^{\text{lumber}} * \tfrac{0.37}{50} * \textstyle\sum_{w=1}^{50} \{I^{\text{6-lumber}}_w + 2 * (Ob^{\text{6-lumber}}_w + \Theta b^{\text{6-lumber}}_w)\} \\ &+ CF^{\text{paper}} * P^{\text{6-paper}} * \tfrac{0.37}{50} * \textstyle\sum_{w=1}^{50} \{I^{\text{6-paper}}_w + 2 * (Ob^{\text{6-paper}}_w + \Theta b^{\text{6-paper}}_w)\} \end{aligned} \quad (8.6)$$

For example, under the Step demand, $C^1_{\text{realistic}} = 85 * 430 * \frac{0.37}{50} * (1 + \frac{0.37}{50})^2 * \sum_{w=1}^{50} \{I^1_w + 2 * (Ob^1_w + \Theta b^1_w)\} = 274.49 * \sum_{w=1}^{50} \{I^1_w + 2 * (Ob^1_w + \Theta b^1_w)\}$. We recall that all costs in this chapter are in Canadian dollars for a market consumption of 70,000 Mpmp/year of lumber and 63,000 tma/year of paper, and that the demand patterns in Table 7.1 represent the distribution of these lumber and paper demands during a year. In the rest of the dissertation, all costs are "realistic", and thus, we drop this subscript, i.e., $\forall i, C^i = C^i_{\text{realistic}}$. Note that realistic costs incur greater numbers than improved costs, and thus, the unit is now k\$ (= 10^3\$), instead of \$.

We now optimize A", B and D according to these prices. In order to understand how these optimizations are obtained, we recall that:

- A" is a classic (s, S) ordering policy, in which the company orders for $I - S$ products, when inventory I is lower than s. In this case, s and S are the two parameters to optimize, so that the inventory system of the company has the lowest cost, Since this scheme does not assume, that companies collaborate, the parameters s and S are optimized, so that each company cost is minimum.

- B and D are the two ordering schemes that follow our two principles, in which companies use (O, Θ) orders. In these two schemes, O is an order, that follows the lot-for-lot policy. The emission of orders Θ is ruled by a parameter λ, so that the inventory level eventually stabilizes on its initial level. λ only depends on the ordering and shipping delays, but the initial inventory level is also important, because it is chosen to balance between inventory holding and backorder costs. Since B and D assume that companies collaborate, we choose λ so, that the total supply chain cost is minimum.

8.2.2 Optimization of Scheme A''

Scheme A'' requires the same optimization as in Chapter 7. This optimization is presented in Subsection B.2, and was also used in Chapter 7. In this chapter, we keep the same parameters as in Chapter 7, that are:

- $s = 0$;
- $S = Oi_w^i$, where Oi_w^i is the company i's incoming order in week w;
- $\forall i, I_1^{i-A''} = 0$, that is, all initial inventories are empty.

8.2.3 Optimization of Scheme B

We now set the two parameters λ and I_1^i (i.e., company i's inventory in the first week) in Scheme B. Parameter λ is easier to choose than the initial inventory level, because λ only depends on the delays between each company and its direct supplier(s), while the initial inventory level depends both on the demand and on the pricing function. In particular, if backorders were considered as free, the optimal value of the initial inventory level would be zero, because companies would only want to avoid holding products in inventory.

The first parameter, λ, represents the quantity of Θ to send to eventually stabilize the inventory at its initial level, when the market consumption becomes steady. This parameter only depends on the ordering and shipping delays between the considered company and its direct supplier. We can note here, that λ *is not set with an optimization process.*

The second parameter, I_1^i, depends on the structure of the supply chain and on the cost function. It results from an optimization process, and we call its result I_1^{i-B}. We first seek the good value of λ, because it rules the process of inventory stabilization, and next the optimal value of I_1^i, because the incurred cost depends on this parameter, which indicates the level at which inventory eventually stabilizes, when the demand is steady.

- *λ:* In our simulations, we use $\lambda = 4$ for every company, as detailed in Subsection B.3, because this value is the sum of ordering and shipping delays.
- *I_1^i:* Since inventory levels fluctuate during a longer period with the retailer than with its suppliers, and since this fluctuation is more important for the suppliers than for the retailer, the optimal initial inventory level is not the same for all companies. Since the ordering scheme B is collaborative, we define the optimal initial inventories as the inventory levels incurring the lowest cost C for the whole supply chain (which may be different from the optimal initial inventory for each company). This optimum is experimentally obtained on our simulation with the Solver in Microsoft Excel. Section B.3 further explains this optimization, whose result is presented in column "Ordering scheme B" in Table 8.1. This optimization impacts on the simulation outcomes.

The format of data in Table 8.1 is $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$. For example, with Scheme B and under the Step demand, $[0, 0, 0, 30, 0, 39, 39] \rightarrow 173 + 198 + 139 + 609 + 282 + 703 = 2,104$ k\$ means that the supply chain cost C is minimum when the two Retailers and the LumberWholesaler have empty initial inventories, the PaperWholesaler has 30 items, the PaperMill has nothing, and the SawMill 39 units of lumber and 39 units of paper at the beginning of the simulation. This initial state of the supply chain leads to a cost $C = 2,104$ k\$ for the supply chain when all companies use B. In this case, for example, the SawMill incurs a cost of $C^6 = 703$ k\$ for its two inventories.

8.2.4 Optimization of Scheme D

The rule D is very similar to B, and therefore has the same two parameters λ and I_1^n to be set up. These two setups have to be carried out in the same order: first λ, next I_1^n:

- *λ:* Since Scheme D uses information centralization, market consumption information travels instantaneously in the supply chain, and therefore, the ordering delay does not have to be taken into account to set λ.

Now, λ is only equal to the shipping delay, i.e., $\lambda = 2$. This is true for all companies, except for both retailers. In fact, retailers have to use the same λ as with Scheme B, i.e., $\lambda^{\text{retailer}} = 4$, which is equal to the sum of the shipping delay and of the ordering delay.

This difference between retailers and all other companies is due to the fact, that retailers have to overorder more (by sending $\Theta \neq 0$) than other companies, because they do two things when the market consumption changes: they send Θ and ship products corresponding to the new consumption. On the other hand, all other companies do only one thing: they send Θ. In fact, orders received by companies, except retailers, will have the new value of the market when the ordering delay has elapsed, because every company places orders with this new value in O as soon as this value appears on the market. Two weeks later, these companies will ship products corresponding to the new consumption, like retailers, but two weeks after them. As a consequence, retailer's inventory varies greater than others' inventory, and thus, they have to send more positive or negative Θ.

Subsection B.4 describes how to choose λ. We only note here that in our simulations, every company uses $\lambda = 2$, except retailers that use $\lambda = 4$, or in other terms, retailers use the same λ than with B.

- I_1^i: The difference between rules B and D is the use of information centralization, which allows improving the reactivity of the supply chain to changes on the market. More precisely, this improvement reduces inventory fluctuations, and therefore, the duration of backorders and of overstockings. Since inventories fluctuate less with Scheme B than with D, every company saves money if it chooses a lower initial inventory with Scheme D.

  The optimal initial inventory is obtained with Solver in Microsoft Excel with the same method as B. This is explained in Subsection B.4. The results are summarized in Table 8.1, with the same format as parameters for B.

8.3 Adaptation of Game Theory Concepts to our Simulation

The three ordering schemes optimized in the previous section are simulated in the next section. Before these simulations, we adapt definitions of game theory in Subsection 4.2 to our needs. After this adaptation, we illustrate their use on a classic game called the Prisonners' Dilemma.

Market cons. pattern	Ordering scheme A"	Ordering scheme B	Ordering scheme D
1. Step	[0, 0, 0, 0, 0, 0, 0] →2,658+5,988+14,764+35,574+98,680+149,945=307,609 k$	[0, 0, 0, 30, 0, 39, 39] →173+198+139+609+282+703=2,104 k$	[0, 0, 0, 0, 12, 24, 24] →109+208+75+159+265+416=1,232 k$
2. Inversed step	[0, 0, 0, 0, 0, 0, 0] →1,061+1,853+12,471+22,331+72,635+107,140=217,491 k$	[0, 0, 0, 0, 0, 6, 6] →30+48+93+140+260+242=813 k$	[0, 0, 0, 0, 0, 6, 6] →30+43+51+74+101+173=470 k$
3. Dirac	[0, 0, 0, 0, 0, 0, 0] →692+955+2,796+6,945+25,737+38,319=75,444 k$	[0, 0, 0, 6, 0, 0, 0] →61+28+61+166+100+119=535 k$	[0, 0, 0, 0, 0, 0, 0] →48+95+53+94+110+56=456 k$
4. Inversed Dirac	[0, 0, 0, 0, 0, 0, 0] →3,299+6,732+19,531+41,176+120,053+176,755=367,546 k$	[0, 0, 0, 0, 0, 0, 0] →73+114+73+114+114+80=568 k$	[0, 0, 0, 0, 0, 0, 0] →58+102+55+97+75+38=427 k$
5. Increase	[0, 0, 0, 0, 0, 0, 0] →1,327+2,722+5,553+18,267+49,087+75,991=152,947 k$	[10, 10, 26, 26, 43, 69, 69]→8+17+28+50+105+308=516 k$	[10, 9, 19, 19, 28, 58, 58]→8+32+19+37+69+364=529 k$
6. Decrease	[0, 0, 0, 0, 0, 0, 0] →3,692+7,029+23,648+48,107+143,390+215,647=441,513 k$	[0, 0, 0, 0, 0, 0, 0] →57+82+141+204+312+275=1,071 k$	[0, 0, 0, 0, 0, 0, 0] →57+82+95+150+206+144=734 k$
7. Weak seasonality	[0, 0, 0, 0, 0, 0, 0] →1,563+1,748+5,583+9,089+22,739+32,454=73,176 k$	[0, 1, 4, 6, 4, 2, 2] →83+154+55+174+79+179=724 k$	[2, 1, 2, 1, 2, 0, 0] →68+90+40+66+50+41=354 k$
8. Medium seasonality	[0, 0, 0, 0, 0, 0, 0] →509+864+1,494+1,886+10,492+14,645=29,890 k$	[5, 4, 14, 9, 7, 4, 4] →129+151+267+342+519+571=1,979 k$	[4, 4, 5, 5, 3, 0, 0] →91+135+191+271+324+230=1,242 k$
9. Strong seasonality	[0, 0, 0, 0, 0, 0, 0] →621+958+2,902+5,893+29,358+43,320=83,052 k$	[8, 8, 16, 16, 12, 16, 16]→160+220+311+480+868+887=2,926 k$	[6, 6, 11, 11, 11, 8, 8]→162+239+294+415+522+428=2,058 k$

Table 8.1: Optimal initial inventories and induced costs of homogeneous supply chains (recall of Table C.7).

8.3.1 Definitions

The 729 sets of costs $\{C^i\}_{i=1}^6$ obtained in **simulateSupplyChain** are used to build a game in the normal form, that consists [Cachon and Netessine, 2003] of:

- players, who are here the six company-agents written as i;
- strategies r^i, which are the three ordering rules A", B, and D available to each company i (we use indifferently the terms "rule", "scheme" and "strategy");
- payoffs/utilities, which are here "replaced" by all company's costs C^i.

This latter replacement is due to the fact, that we do not consider a company's utility but its inventory and backorder costs. Since we assume production cost is equal to zero, we could refer to profit by adding a minus sign to the costs, but instead of that, we only consider costs in order to facilitate the readability of data. Therefore, agents do not want to maximize their utility, but they seek to minimize their costs. The main adaptations between our notations and traditional economic definitions, come from this difference, and other adaptations are due to the use of supply chain management vocabulary. We recall that, similar to the rest of this dissertation, companies are written as power, e.g., C^i is company i's cost, and C is the overall supply chain cost.

The following definitions are adapted from Jehle and Reny [2000]'s book:

Strategic form game: A strategic form game is a tuple $G = (R^i, c^i)_{i=1}^N$, where for each agent $i = 1, \ldots, N$, R^i is the finite set of ordering rules available to agent i, and $C^i : \times_{j=1}^N R^j \to \Re$ describes agent i's cost as a function of the strategies chosen by all agents. The values of the functions C^i are obtained through simulations in **simulateSupplyChain()**. In this dissertation, there are $N = 6$ agents, each having the cost function $C^i : R^1 \times R^2 \times R^3 \times R^4 \times R^5 \times R^6 \to \Re$.

Joint strategy/rule: The set r of ordering schemes $(r^1, r^2, r^3, r^4, r^5, r^6)$ used by companies is a joint strategy.

More precisely, r is a vector of six strategies/ordering rules, where the first rule, r^1, refers to the rule used by the LumberRetailer, the second, r^2, to the rule used by the PaperRetailer,... and the sixth, r^6, to the rule used by the Sawmill (again, correspondence between i and the position in the supply chain is given in Figure 6.3, and recalled in Figure 8.1).

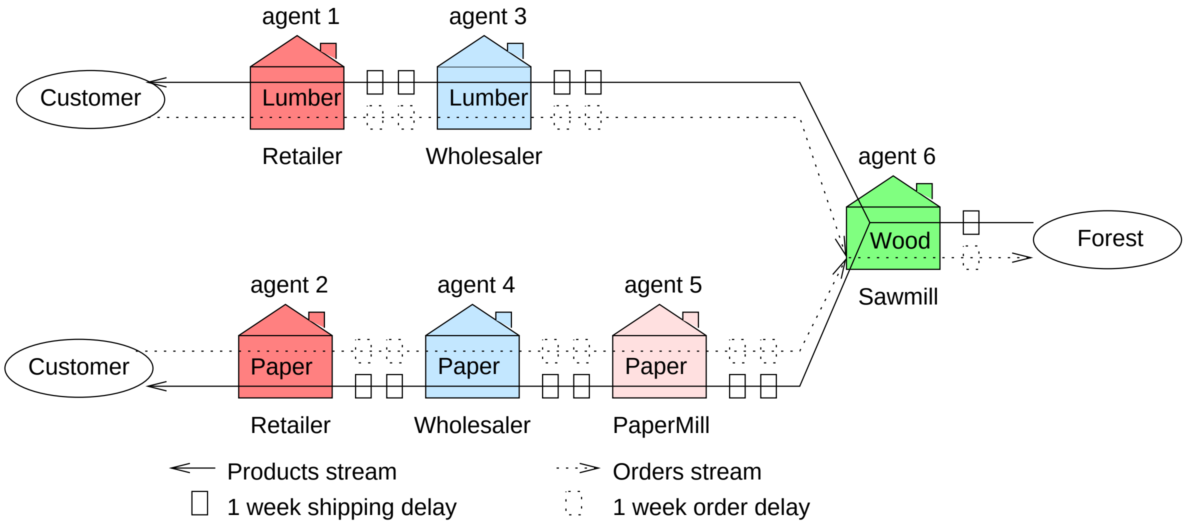


Figure 8.1: Forest supply chain in the QWSG (recall from Figure 6.3).

We write r^{-i} the joint strategy used by all companies except i, and therefore $r = (r^i, r^{-i})$ for any company i. For example, $C(r) = \sum_{i=1}^{6} C^i(r^i, r^{-i})$ is the cost incurred by the entire supply chain, when the joint strategy r is used. In this example, $C^4(r^4, r^{-4}) = C^4(r^4, (r^1, r^2, r^3, r^5, r^6))$ is the cost for the PaperWholesaler ($i = 4$) of ordering with r^4, when the rest of the supply chain uses the joint strategy/rule $r^{-4} = (r^1, r^2, r^3, r^5, r^6)$.

Strictly dominant strategy/rule: A strategy/rule $\hat{r}^i$ for company i is strictly dominant if $C^i(\hat{r}^i, r^{-i}) < C^i(r^i, r^{-i})$, for all (r^i, r^{-i}) such as $r^i \neq \hat{r}^i$.

In other words, a strictly dominant strategy/rule $\hat{r}^i$ always incurs a lower cost C^i for company i than any other ordering strategy/rule, whatever rule is used by the five other companies. More precisely, the dominant strategy $\hat{r}^i$ is always the best choice for company i, and therefore, only this company should use it. Such best strategies are very rare, which explains why we look rather for strictly dominated strategies in our experiments.

Strictly dominated strategy/rule: Company i's strategy $\hat{r}^i$ strictly dominates another of its strategies $\check{r}^i$, if $C^i(\hat{r}^i, r^{-i}) < C^i(\check{r}^i, r^{-i})$ for all r^{-i}. In this case, we also say that $\check{r}^i$ is strictly dominated.

This means that whatever joint rule r^{-i} is chosen by the five other companies, a dominated strategy $\check{r}^i$ incurs higher costs C^i than another rule $\hat{r}^i$. More precisely, the dominated strategy $\check{r}^i$ is always worse than $\hat{r}^i$ for company i. Note that we only consider strictly dominated strategies, but never weakly dominated strategies, i.e., we only consider "$<$" and never "$\leq$" in the previous definition.

Next, we use the software McKelvey et al. [2004]'s Gambit 0.97.05 to remove strictly dominated ordering schemes. Since Gambit only tries to maximize agents' utility, it finds out how to maximize costs! To solve this problem, we add a minus sign before our simulation outputs, in order to transform costs into profits when companies do not earn money. We do not show these negative values in this dissertation, in order to facilitate readability.

Pure Nash equilibrium: The joint strategy/rule $\hat{r}$ is a pure Nash equilibrium if for each company i, $C^i(\hat{r}^i, \hat{r}^{-i}) \leq C^i(r^i, \hat{r}^{-i})$ for all $r^i \neq \hat{r}^i$, where $\hat{r} = (\hat{r}^i, \hat{r}^{-n})$.

Conversely to the two above domination relations, a Nash equilibrium deals with joint strategies and not with individual strategies. A Nash equilibrium is a stable state of the supply chain: when companies choose the joint strategy $\hat{r} = (\hat{r}^1, \hat{r}^2, \hat{r}^3, \hat{r}^4, \hat{r}^5, \hat{r}^6)$, none of the companies has incentives to use another ordering scheme while fully aware of the others' behaviour. In other words, no company wants to unilaterally deviate from it, since such behaviour would lead to higher costs [Cachon and Netessine, 2003]. For example, if $\hat{r}^1$=D in the Nash equilibrium $\hat{r}$, the LumberRetailer has no incentive to change $\hat{r}^1$ for A" or B, when this retailer observes others' behaviour. This does not mean that a Nash equilibrium is the best joint strategy for the supply chain (i.e., it does not incur the minimum of costs C), it only means that the supply chain will remain in this equilibrium after it is reached: it is a "shaft state".

As previously stated, to simplify the research of Nash equilibria, we first successively eliminate all strictly dominated strategies, since no Nash equilibria are lost by such eliminations [McKelvey et al., 2004]. Then, we look for Nash equilibria in the reduced game with Gambit's method "EnumPureSolve", which finds all Nash equilibria in pure strategy.

Minimum of overall cost C: Since we consider costs instead of profits, overall cost C replaces the "social welfare" concept, where $C = \Sigma_i C^i$.

The lower the overall cost C is, the more efficient the supply chain is. This concept of solution is important, because competition has shifted from single companies to supply chains [Stadtler, 2000; Kotler, 2001]. If the supply chain was only one company, its goal would be to minimize C. The problem is that some companies may have no incentive to reach the minimum of C, i.e., some companies may have to sacrifice themselves by increasing their cost, in order to improve the welfare of the rest of the supply chain. Therefore, the fact that a joint strategy incurs the minimum of C means that this joint strategy is the best one for the supply chain as a whole, but it does not mean that it will be used, because some companies may have incentives to deviate from it, because it is not a Nash equilibrium. This point leads us to introduce the following term.

Best and good joint strategy/rule: We call "best joint strategy" (it is not a term from game theory) a joint strategy that both minimizes C and is a Nash equilibrium. In the same way, we refer to "good joint strategy" for a joint strategy that is a Nash equilibrium, but that only approaches the minimum of C.

Pareto-dominated joint strategy: the joint rule r is Pareto-dominated by $\hat{r}$, if $C^i(\hat{r}^i) \leq C^i(r^i)$ for all companies i, and $C^j(\hat{r}^j) < C^j(r^i)$ for at least one company j.

This means that $\hat{r}$ makes someone better off and no one worse off, because $\hat{r}$ strictly reduces one company's cost without increasing any other company's cost. In our experiments, we will use Pareto-domination with the minimum of C to try comparing Nash equilibria. Unfortunately, Pareto-domination cannot order all pairs of joint strategies. This is the reason for the introduction of the following definition.

Pareto-efficient joint strategy: the joint rule $\hat{r}$ is Pareto-efficient, if it is Pareto-dominated by no other joint strategy.

In general, a Pareto-efficient joint strategy does not incur the minimum of C, nor is it a Nash equilibrium, and therefore, it is not what we call "the best joint strategy".

Full collaboration: we call "full collaboration" (it is not a term from game theory) the joint strategy in which all players apply D. This corresponds to a supply chain at the higher level of collaboration.

Let us consider the Prisoners' Dilemma, in order to illustrate that a Pareto-efficient joint strategy is not a Nash equilibrium. However, the Pareto-efficient joint strategy in this example is also the minimum of C, which is not always true.

8.3.2 Example of the Prisoners' Dilemma

The game in Table 8.2 is an adaptation of the Prisoners' Dilemma [Guerrien, 1995; Sandholm, 1999; Wooldridge, 2001; Yildizoglu, 2003] to cost minimization. This game has the following scenario [Wooldridge, 2001]:

> Two men, called Player1 and Player2, are both charged with a crime and held in separate cells. They have no way of communicating with each other or making any kind of agreement. The two men are told (i) if one of them

		Player2	
		Cooperate	Defect
Player1	Cooperate	1 ; 1	6 ; 0
	Defect	0 ; 6	5 ; 5

Table 8.2: Adaptation of the Prisoners' Dilemma to cost minimization.

> confesses to the crime and the other does not, the confessor will go free, and the other will be jailed for three years, and (ii) if both confess to the crime, then each will be jailed for two years. Both prisonners know that if neither confesses, then they will each be jailed for one year.

Table 8.2 is an instance of the game having these rules. In this table, "0 ; 6" in the entry (Defect, Cooperate) represents costs incurred by both players, when Player1 chooses Defect and Player2 chooses Cooperate. In this situation, Player1 incurs a cost of \$0 and Player2 a cost of \$6. The question is to find which joint strategy, i.e., which entry in Figure 8.2, will be conjointly chosen by players. In fact, Player1 (respectively Player2) chooses a line (respectively a column) so that her/his cost is minimized, because s/he is rational, but we will see that this behaviour leads them to choose a non Pareto-efficient joint strategy, i.e., we could make one player strictly better off without the other player being worse off!

Cooperate is a dominated strategy for both players, because each time a player chooses this rule, s/he would be better to change for Defect. We check this twice:

- Let us consider Player1:
  - her/his cost is \$1 if Player2 also chooses Cooperate;
  - her/his cost is \$6 if Player2 chooses Defect.

  In this context, Player1 could respectively incur 0 \$ and 5 \$ if s/he had chosen Defect. Therefore, whatever is chosen by Player2, Player1 prefers Defect. Cooperate is thus dominated by Defect for Player1.
- We repeat the same reasoning by replacing Player1 by Player2 to show that Cooperate is a dominated strategy for Player2.

Therefore, Cooperate is a dominated strategy for both players. These dominated strategies are not considered by both players, because they always incur a higher cost. When we remove them from Figure 8.2, (Defect, Defect) is the only remaining joint strategy. No player has an incentive to leave it. It is thus a Nash equilibrium, and more precisely, we call such an equilibrium a iterated dominance equilibrium (cf. Subsection 4.2.4). The entry (Defect, Defect) is thus chosen in Figure 8.2 by our players.

The problem is that this Nash equilibrium is Pareto-dominated by (Cooperate, Cooperate), because both players have the lowest cost there. Both players should prefer (Cooperate, Cooperate), but both also have incentives to change for Defect too. Therefore, (Cooperate, Cooperate) is not a Nash equilibrium, even if both players prefer it to the Nash equilibrium (Defect, Defect). Next, (Cooperate, Cooperate) also incurs the minimum of C (=1+1), and therefore, the minimum of C is not a Nash equilibrium in this example. In short, (Cooperate, Cooperate) *both minimizes* C *and Pareto-dominates the Nash equilibrium* (Defect, Defect) *in this game.*

No player chooses Cooperate to reach this joint strategy, because *both players fear* that the other one is choosing Defect. Each player has to *trust* the other (e.g., by signing a contract), if s/he wants to reach (Cooperate, Cooperate), but this does not conform to game theory. Moreover, we can notice that all pairs of joint strategies cannot be compared with Pareto-domination, e.g., (Cooperate, Cooperate) neither Pareto-dominates nor is Pareto-dominated by (Defect, Cooperate) and (Cooperate, Defect).

We can wonder if situations resembling the Prisoners' Dilemma appear in supply chains, that is, if there is a joint strategy minimizing both individual and the overall costs, but no company chooses it, because each one fears the decision of others. We answer this question in the next section, by building a game with QWSG simulation outcomes.

8.4 Simulation of Heterogeneous Supply Chains

This section presents the second series of experiments, in which each company may use its preferred ordering scheme. Simulations use the method of realistic cost evaluation and the optimal parameters of ordering schemes presented in Section 8.2. Next, the concepts from game theory introduced in Section 8.3 are applied to analyze simulation outcomes.

This series of experiments is presented in an iterative way: we start from the homogeneous supply chain in Subsection 8.4.1, in which all companies use the same ordering scheme. Then, we assume the PaperRetailer can use another ordering scheme than the rest of the supply chain in Subsection 8.4.2. Next, we assume the two retailers may decide which ordering rule they use in Subsection 8.4.3. Finally we consider in Subsection 8.4.4 the case of the heterogeneous supply chain, where each company decides which ordering scheme it uses.

8.4.1 Analysis of Simulation Outcomes with one Player: The Homogeneous Supply Chain

In this subsection, we consider the same scenario as in Chapter 7, because all companies use the same ordering scheme, except that:

- we consider realistic costs instead of improved costs;
- initial inventories are not empty;
- the ten Uniform random market consumptions are ignored;
- only the Schemes A", B and D are simulated.

A simulation is carried out for each of the three ordering schemes and for each of the nine market consumption patterns, i.e., $3 \times 9 = 27$ simulations. For each simulation, the individual costs C^i and the global cost C are given in Table 8.3 with the format $(r^1, r^2, r^3, r^4, r^5, r^6) \to C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$. For example, for Scheme B under the Inversed step demand in Table 8.3, the entry is (B, B, B, B, B, B)$\to 30 + 48 + 93 + 140 + 260 + 242 = 813$ k\$, which means that the PaperWholesaler has a cost $C^4 = 140$ k\$ and the entire supply chain $C = 813$ k\$. In each line, the lowest cost is underlined, e.g., in line "Inversed step", $C = 470$ k\$ incurred by Scheme D is underlined, because it is lower than $C = 217,491$ k\$ incurred by Scheme A", and than $C = 813$ k\$ incurred by Scheme B.

We can notice that Scheme D always incurs a lower cost C than Scheme B, which itself incurs a much lower C than Scheme A", and thus, C incurred by D is always underlined. However, this is true for all C, but not for all C^i. In fact, some companies are sometimes better off with Scheme B than with Scheme D. For example, with the Step demand, the PaperRetailer incurs a cost $C^2 = 198$ k\$ with B, which is lower than $C^2 = 208$ k\$ with D.

In this case, the problem is that PaperRetailer may disagree to multicast the market consumption required by Scheme D, and therefore, the rest of the supply chain may not

Market cons. pattern	Ordering scheme A”	Ordering scheme B	Ordering scheme D
1. Step	(A”, A”, A”, A”, A”, A”) →2,658+5,988+14,764 +35,574+98,680+149,945 =307,609 k$	(B, B, B, B, B, B) →173+198+139 +609+282+703 =2,104 k$	(D, D, D, D, D, D) →109+208+75 +159+265+416 =1,232 k$
2. Inversed step	(A”, A”, A”, A”, A”, A”) →1,061+1,853+12,471 +22,331+72,635+107,140 =217,491 k$	(B, B, B, B, B, B) →30+48+93 +140+260+242 =813 k$	(D, D, D, D, D, D) →30+43+51 +74+101+173 =470 k$
3. Dirac	(A”, A”, A”, A”, A”, A”) →692+955+2,796 +6,945+25,737+38,319 =75,444 k$	(B, B, B, B, B, B) →61+28+61 +166+100+119 =535 k$	(D, D, D, D, D, D) →48+95+53+ 94+110+56 =456 k$
4. Inversed Dirac	(A”, A”, A”, A”, A”, A”) →3,299+6,732 +19,531+41,176 +120,053+176,755 =367,546 k$	(B, B, B, B, B, B) →73+114+73 +114+114+80 =568 k$	(D, D, D, D, D, D) →58+102+55 +97+75+38 =427 k$
5. Increase	(A”, A”, A”, A”, A”, A”) →1,327+2,722+5,553 +18,267+49,087+75,991 =152,947 k$	(B, B, B, B, B, B) →8+17+28 +50+105+308 =516 k$	(D, D, D, D, D, D) →8+32+19 +37+69+364 =529 k$
6. Decrease	(A”, A”, A”, A”, A”, A”) →3,692+7,029 +23,648+48,107 +143,390+215,647 =441,513 k$	(B, B, B, B, B, B) →57+82+141 +204+312+275 =1,071 k$	(D, D, D, D, D, D) →57+82+95 +150+206+144 =734 k$
7. Weak seasonality	(A”, A”, A”, A”, A”, A”) →1,563+1,748+5,583 +9,089+22,739+32,454 =73,176 k$	(B, B, B, B, B, B) →83+154+55 +174+79+179 =724 k$	(D, D, D, D, D, D) →68+90+40 +66+50+41 =354 k$
8. Medium seasonality	(A”, A”, A”, A”, A”, A”) →509+864+1,494 +1,886+10,492+14,645 =29,890 k$	(B, B, B, B, B, B) →129+151+267 +342+519+571 =1,979 k$	(D, D, D, D, D, D) →91+135+191 +271+324+230 =1,242 k$
9. Strong seasonality	(A”, A”, A”, A”, A”, A”) →621+958+2,902 +5,893+29,358+43,320 =83,052 k$	(B, B, B, B, B, B) →160+220+311 +480+868+887 =2,926 k$	(D, D, D, D, D, D) →162+239+294 +415+522+428 =2,058 k$

Table 8.3: Costs with one player (similar to the homogeneous supply chain presented in Table 8.1, but with another data format).

be able to use D and to have the cost $C = 1,232$ k\$. In order to avoid having the cost $C = 2,104$ k\$ incurred when all companies use B, the LumberWholesaler may multi-cast its incoming orders to the rest of the supply chain. We can expect the overall supply chain cost to be between the two previous values of C, but we have to simulate this scenario in order to know its exact value. In the next subsection we study a scenario including this case, where all companies use D except the LumberRetailer who uses B. In this case, the LumberRetailer is a player who can choose her/his ordering rule and the RestOfTheSupplyChain is another player[4].

8.4.2 Analysis of Simulation Outcomes with two players

As a first simple example of individual behaviour, we assume there are now two players: the PaperRetailer vs. the RestOfTheSupplyChain. This is interesting, because there are at least three reasons to study the behaviour of a retailer:

1. the bullwhip effect begins with retailers;
2. retailers are the companies that decide to use information centralization or not;
3. we noticed in Subsection 8.4.1, that retailers may have incentives not to share information, and thus not to multi-cast the information required by information centralization, because they may be worse off doing so.

When two players are considered, a little heterogeneity is introduced in the supply chain. In this context, simulations are carried out with the nine joint strategies in Table 8.4. These joint strategies are given with the format $(r^2; r^{-2})$, i.e., the PaperRetailer's strategy is first given, then the RestOfTheSupplyChain's:

1. *Simulation* (A" ; A"): Both players use Rule A", which is exactly the experiment carried out in Subsection 8.4.1 with a homogeneous supply chain using A".
2. *Simulation* (A" ; B): The PaperRetailer uses Scheme A", and thus only gives one piece of information to the RestOfTheSupplyChain, or more precisely, to the PaperWholesaler. The PaperWholesaler is therefore the first company to use (O, Θ)

---

[4] We recall that (O, Θ) orders in Schemes B and D are vectors of orders, that are used to share the demand information in Schemes B and D, so that our two principles proposed in Section 5.1 are followed. We also recall that information centralization is an improvement on information sharing, because the sharing is quicker, due to the fact that retailers multi-cast their sales in real-time to the rest of the supply chain.

orders in the paper sub-supply chain, and for example, O received by the PaperMill are not the market consumption, but the output of the PaperRetailer's rule A".

3. *Simulation* (A" ; D): The PaperRetailer gives only one piece of information to the PaperWholesaler, and does not multi-cast the market consumption required by its suppliers. As in Simulation (B ; D), the PaperWholesaler is the first company to use (O, Θ) orders. Moreover, the PaperWholesaler multi-casts its incoming orders to the RestOfTheSupplyChain, i.e., the PaperMill and the Sawmill have to take PaperWholesaler's incoming orders as the basis for information centralization, which is taken into account by conditions in Equations A.25 and A.26.

4. *Simulation* (B ; A"): The PaperWholesaler gets (O, Θ) orders from PaperRetailer, that are fulfilled by shipping products without any adaptations, because the basic modelization of all companies manages incoming Θ, even when the considered company does not place (O, Θ) order (cf. Equations A.2 and A.3 that describe Too^i_w and Tob^i_w). PaperWholesaler's and other companies in RestOfTheSupplyChain's orders are next based on their inventory level with Scheme A".

5. *Simulation* (B ; B): Both players use Scheme B, which is exactly the experiment carried out in Subsection 8.4.1 with a homogeneous supply chain using B.

6. *Simulation* (B ; D): The PaperRetailer does not multi-cast the market consumption, but the RestOfTheSupplyChain requires this information. Therefore, like in (A" ; D), information centralization is achieved by the PaperWholesaler. In other words, the PaperMill and the Sawmill have to take PaperWholesaler's incoming orders as the nearest value of market consumption, rather than PaperRetailer's incoming orders.

7. *Simulation* (D ; A"): PaperWholesaler gets (O, Θ) orders from PaperRetailer that are fulfilled by shipping products without any adaptations, because the basic modelization of all companies manages incoming Θ, even when the considered company does not place (O, Θ) order (cf. Equations A.2 and A.3 describing Too^i_w and Tob^i_w). PaperWholesaler's and other companies in RestOfTheSupplyChain's orders are next based on their inventory level with Scheme A", without taking into account the market consumption multicasted by the PaperRetailer. Therefore, this simulation is the same as (B, A").

8. *Simulation* (D ; B): The PaperWholesaler multi-casts the market consumption, but none of its suppliers use it. That is, the simulation is the same as (B, B), because the difference between B and D (information centralization) is not exploited by neither the PaperWholesaler, nor the PaperMill, nor the Sawmill.

9. *Simulation* (D ; D): Both players use Rule D. This is exactly the experiment carried out in Subsection 8.4.1 with a homogeneous supply chain using D.

		RestOfTheSupplyChain		
		A”	B	D
PaperRetailer	A”	(A” ; A”)	(A” ; B)	(A” ; D)
	B	(B ; A”)	(B ; B)	(B ; D)
	D	(D ; A”)	(D ; B)	(D ; D)

Table 8.4: The nine simulations required to study the PaperRetailer vs. the RestOfTheSupplyChain (two player game).

		RestOfTheSupplyChain		
		A”	B	D
PaperRetailer	A”	5,988 ; 301,621	1,468 ; 14,419	1,899 ; 8,266
	B	1,286 ; 54,275	198 ; 1,906	619 ; 1,507
	D	1,286 ; 54,275	198; 1,906	208 ; 1,024

Table 8.5: Outcomes when the PaperRetailer and the RestOfTheSupplyChain are selfish (two player game).

We first simulate these nine cases under a Step market consumption, that is, both lumber and paper markets consume eleven products per week in the four first weeks, then seventeen products per week from Week five to Week fifty (cf. Table 7.1). The companies' outcomes in these nine simulations are given in Table 8.5. For example, when the PaperRetailer uses A” and the RestOfTheSupplyChain uses D, the outcome is (1,899 ; 8,266) which means the PaperRetailer has a cost of 1,899 k$ and the RestOfTheSupplyChain 8,266 k$, without any distinctions between the company belonging to this group of companies. Since we assume all companies have no income, the traditional utility/profit maximization is achieved by minimizing costs: every company wants to reach the entry in Table 8.5 in which its cost is the lowest.

In this context, we can note that ordering with A” is strictly dominated for the PaperRetailer, because it always incurs a higher cost when it uses this rule than when it uses B or D, whatever is chosen by the RestOfTheSupplyChain. Therefore, the PaperRetailer never chooses A”. In the same way, A” and B are dominated for the RestOfTheSupplyChain. We assume that both players know these facts (or they infer this information based on history). This first turn of elimination of dominated strategies leads to Table 8.6.

The same reasoning can be applied in a second turn of elimination of stricly dominated strategies: Scheme B is dominated for the PaperRetailer. B was not dominated for the PaperRetailer in Table 8.5, but becomes so in Table 8.6. When B is eliminated, we obtain Table 8.7, in which only D is available to both players. The reduced game in

RestOfTheSupplyChain

PaperRetailer

	D
B	619 ; 1,507
D	208 ; 1,024

Table 8.6: New Table 8.5 after the first turn of elimination of dominated strategies.

RestOfTheSupplyChain

PaperRetailer

	D
D	208 ; 1,024

Table 8.7: New Table 8.5 after the second turn of elimination of dominated strategies.

Table 8.7 has the same Nash equilibria as the original one in Table 8.5 [McKelvey et al., 2004].

These dominances are summarized in the first line, called "Step", in Table 8.9, in which "1" means A″ is dominated for both players and B for the RestOfTheSupplyChain in the original game, and "2" that B becomes dominated for the PaperRetailer if "1" rules are not taken into account.

After the iterative elimination of strictly dominated strategies, we look for Nash equilibria in the remaining strategies. Since (D, D) is the unique survivor of the elimination of dominated strategies, it is a Nash equilibrium. If several joint strategies had survived the elimination of strictly dominated strategies, we would have tested each of them to check which ones are Nash equilibria, and there may be several or no equilibria.

We could have looked for Nash equilibria in the complete game in Table 8.5 instead of Table 8.7, but it would have been a longer job. In fact, we must check if each of the nine entries in this matrix is a "shaft state", that is, if each player has no incentive to change its ordering rule when it sees what is made by the other player. More precisely, (D, D) is a Nash equilibrium in Table 8.5, because:

- if the PaperRetailer changes for A″, it increases its costs from 208 k$ to 1,899 k$, and if this player changes for B, its cost increases to 619 k$;
- if the RestOfTheSupplyChain changes for A″, it increases its cost from 1,024 k$ to 1,906 k$, and to 54,275 k$ if it changes for B.

We can note here that a player does not wish to change its ordering rule if it does not *strictly* reduces its cost. For example, if (D, D) incurred (619 ; 1,024) instead of

		RestOfTheSupplyChain		
		A"	B	D
PaperRetailer	A"	5,988+301,621=307,609	1,468+14,419=15,887	1,899+8,266=10,165
	B	1,286+54,275=55,561	198+1,906=2,104	619+1,507=2,126
	D	1,286+54,275=55,561	198+1,906=2,104	208+1,024=$\underline{1{,}232}$

Table 8.8: Overall supply chain's costs when the PaperRetailer and the RestOfTheSupplyChain are selfish.

(208 ; 1,024), the PaperRetailer would neither increase nor decrease its cost by changing for B: (D, D) would thus remain a Nash equilibrium. On the other hand, (B, B) is not an equilibrium, because when both players use B, the RestOfTheSupplyChain prefers to change for D in order to decrease its cost from 1,906 k$ to 1,507 k$.

Once the Nash equilibria are determined, we wonder whether the Nash equilibrium (D, D) is a "good" solution for the whole supply chain. Until now, we have just checked that no players would like to leave this joint strategy, but we do not know if it is a good joint strategy, meaning as inducing the minimal of C while being a Nash equilibrium. Practically, we add the PaperRetailer's cost C^2 to the RestOfTheSupplyChain's cost $C^{-2}(= C^1 + C^3 + C^4 + C^5 + C^6)$ for the nine entries in Table 8.5, which results in $C(= C^1 + C^2 + C^3 + C^4 + C^5 + C^6)$. This calculation is detailed in Table 8.8. We can see that the minimum of costs for the supply chain is 1,232 k$ (this lowest C is underlined in Table 8.8), which is reached for the joint strategy (D, D). Therefore, this joint strategy is not only stable, but it is also the best joint strategy for the global supply chain.

We do not consider Pareto-dominations in this example, because we have here only one Nash equilibrium, while we use this relation for comparing Nash equilibria. Moreover, we can also note that a situation resembling the Prisonners' Dilemma does not occur here.

We did the same nine simulations for the eight other market consumption patterns Inversed step, Dirac, Inversed Dirac, Increase, Decrease, and Weak, Medium and Strong seasonalities detailed in Table 7.1. Analysis results of simulation outcomes are summarized in Tables 8.9, 8.10, 8.11 and 8.12. For example, in Table 8.12, for the Strong seasonality pattern, (B, D)→210+2,023=2,233 k$ indicates that the PaperRetailer uses B and the RestOfTheSupplyChain uses D, and 2,233 k$ is the total cost for the supply chain when these two strategies are used together, 210 k$ is the PaperRetailer's cost and 2,023 k$ is the RestOfTheSupplyChain's cost. (B, D) is a Nash equilibrium that does not incur the lowest C among the nine entries of this game, because 2,233 k$ is not

Market consumption patterns	Paper Retailer			RestOfThe SupplyChain		
	A"	B	D	A"	B	D
1. Step	1	2	-	1	1	-
2. Inversed step	1	-	-	1	-	-
3. Dirac	1	-	2	1	1	-
4. Inversed Dirac	1	-	2	1	1	-
5. Increase	1	-	-	1	-	-
6. Decrease	1	-	-	1	-	-
7. Weak seasonality	1	2	-	1	1	-
8. Medium seasonality	1	2	-	1	1	-
9. Strong seasonality	-	2	1	1	1	-

Table 8.9: Strictly dominated strategies when the PaperRetailer and the RestOfTheSupplyChain are selfish.

underlined. The lowest C, 2,060 k\$, is underlined, and appears in the upper entry in each of Tables 8.10, 8.11 and 8.12.

In short, Tables 8.10, 8.11 and 8.12 show all Nash equilibria in pure strategy (second line), if these equilibria incur the minimum of C (first line), and if they correspond to the full collaboration of the supply chain (last line). We focus on full collaboration because it must always incur the minimum of C: we check if this is true. The dream case would be that full collaboration both incurs the minimum of C and is a Nash equilibrium (i.e., the three lines in Tables 8.10, 8.11 and 8.12 indicate the same information), because this best joint strategy would be easy to identify. In fact, it is intuitive that full collaboration is the best solution for the supply chain, but we hope no company has an incentive to deviate from it. It is interesting to note the following conclusion:

> Fortunately, full collaboration incurs the minimum of C and is a Nash equilibrium under half market consumption patterns (Step, Inversed step, Decrease, Weak seasonality and Medium seasonality).

In these patterns, full collaboration is thus both the best joint strategy (i.e., minimum of C and Nash equilibrium) and easy to find (i.e., all companies fully collaborates). When it is not the case, there is at least one Nash equilibrium (Nash equilibria in pure strategies do not always occur), and one of the equilibria incur a C close to the minimum. Finally, when there are two Nash equilibria (Inversed Dirac), the full collaboration (D, D) is Pareto-dominant over the other equilibrium (B, B), because both players have a lower (or equal) cost with (D, D).

	Step	Inversed step	Dirac	Inversed Dirac
Minimum of C	(D, D) →208+1,024 =1,232 k$	(D, D) →43+429 =472k$	(B, D) →90+332 =422 k$	(B, D)→89 +336=425 k$ (D, D)→102 +323=425 k$
Nash equilibria	(D, D) →208+1,024 =1,232 k$	(D, D)→43 +429=472 k$ (B, B)→48 +765=813 k$	(B, D) →90+332 =422 k$	(B, D) →89+336 =425 k$
Full collaboration	(D, D) →208+1,024 =1,232 k$	(D, D) →43+429 =472 k$	(D, D) →95+361 =456 k$	(D, D) →102+323 =425 k$

Table 8.10: Results of analysis when the PaperRetailer and the RestOfTheSupplyChain are selfish (1/3).

	Increase	Decrease
Minimum of C	(B, B)→17+499=516 k$	(D, D)→82+652=734 k$
Nash equilibria	(B, B)→17+499=516 k$ (D, D)→32+497=529 k$	(D, D)→82+652=734 k$ (B, B)→82+989=1,071 k$
Full collaboration	(D, D)→32+497=529 k$	(D, D)→82+652=734 k$

Table 8.11: Results of analysis when the PaperRetailer and the RestOfTheSupplyChain are selfish (2/3).

We now study if all these results still hold when the PaperRetailer no longer belongs to the player called RestOfTheSupplyChain.

8.4.3 Analysis of Simulation Outcomes with three players

We assume now that the two retailers can make their own decisions as two independent players, while the RestOfTheSupplyChain is a third player. We have $3^3 = 27$ different combinations of possible ordering, because there are three players (the PaperRetailer, the LumberRetailer and a coalition called RestOfTheSupplyChain regrouping the four other companies), and each of them has three different choices.

Figure 8.2 presents the outcomes of these 27 simulations for the Step market consumption. These outcomes could be represented in a $3 \times 3 \times 3$ matrix, but this is not

	Weak seasonality	Medium seasonality	Strong seasonality
Minimum of C	(D, D)→90 +265=<u>355 k$</u>	(D, D)→135 +1,107=<u>1,242 k$</u>	(D, D)→239 +1,821=<u>2,060 k$</u>
Nash equilibria	(D, D)→90 +265=<u>355 k$</u>	(D, D)→135 +1,107=<u>1,242 k$</u>	(B, D)→210 +2,023=2,233 k$
Full collaboration	(D, D)→90 +265=<u>355 k$</u>	(D, D)→135 +1,107=<u>1,242 k$</u>	(D, D)→239 +1,821=<u>2,060 k$</u>

Table 8.12: Results of analysis when the PaperRetailer and the RestOfTheSupplyChain are selfish (3/3).

so easy to represent on a paper. Instead, they are represented by three 3 × 3 matrices:

- the player RestOfTheSupplyChain selects which matrix will be used by choosing the strategy A", B or D;
- the LumberRetailer and the PaperRetailer both choose a row and a column in the selected matrix.

These three choices are made at the same time. For example, when the RestOfTheSupplyChain chooses D, the LumberRetailer B and the PaperRetailer A", the outcome is (421 ; 2,059 ; 6,986), which means that the LumberRetailer incurs a cost of 421 k$, the PaperRetailer 2,059 k$ and the RestOfTheSupplyChain 6,986 k$.

We can look for dominances in Table 8.2. In the first round of elimination of strictly dominated strategies, A" incurs a higher cost for each player, whatever is chosen by the two other players. Therefore, A" is dominated in the first round for every player. When A" are no longer taken into account, no other ordering rules become dominated. This single round of elimination of dominated schemes is summarized in the first line (step) in Table 8.13 with dominances for the eight other demand patterns. When there are several survivors, each has to be tested to check if it is a Nash equilibrium, and it is yet possible that the game has no equilibria. Tables 8.14, 8.15 and 8.16 summarizes this equilibrium and those obtained with the other demand patterns.

Finally, we can notice in Table 8.13 that non-collaborating is always dominated for all players, while collaborating is rarely dominated. Next, Tables 8.14, 8.15 and 8.16 show there exists at least one Nash equilibrium for all demand patterns, and that this equilibrium is unique under half of the considered demand patterns. Moreover, these tables also show that full collaboration (D, D, D) is both a Nash equilibrium and incurs the minimum of C with several demand patterns, i.e., with Step, Inversed

The RestOfTheSupplyChain chooses A”:

		PaperRetailer		
		A”	B	D
LumberRetailer	A”	2,658 ; 5,988 ; 298,963	2,664 ; 1,286 ; 51,611	2,664 ; 1,286 ; 51,611
	B	867 ; 6,491 ; 397,135	867 ; 1,193 ; 43,547	867 ; 1,193 ; 43,547
	D	867 ; 6,491 ; 397,135	867 ; 1,193 ; 43,547	867 ; 1,193 ; 43,547

The RestOfTheSupplyChain chooses B:

		PaperRetailer		
		A”	B	D
LumberRetailer	A”	1,075 ; 1,355 ; 12,077	1,093 ; 256 ; 5,312	1,093 ; 256 ; 5,312
	B	442 ; 1,468 ; 13,977	173 ; 198; 1,733	173 ; 198 ; 1,733
	D	442 ; 1,468 ; 13,977	173 ; 198 ; 1,733	173 ; 198 ; 1,733

The RestOfTheSupplyChain chooses D

		PaperRetailer		
		A”	B	D
LumberRetailer	A”	1,084 ; 2,011 ; 10,444	1,179 ; 696 ; 5,679	1,241 ; 332 ; 5,222
	B	421 ; 2,059 ; 6,986	405 ; 638 ; 1,752	380 ; 243 ; 1,330
	D	245 ; 1,899 ; 8,021	133 ; 619 ; 1,374	109 ; 208 ; 915

Figure 8.2: Outcomes when the LumberRetailer, the PaperRetailer and the RestOfTheSupplyChain are selfish (Step demand).

step, Decrease, Weak seasonality, Medium seasonality and Strong seasonality. This is a good result because it means that *no companies would like the entire supply chain to collaborate without itself*. Similarly to the previous subsection, when full collaboration is a Nash equilibrium, i.e., (D, D) in Subsection 8.4.2 and (D, D, D) now, it Pareto-dominates other equilibria, except under the Increase demand.

We check in the next subsection if these conclusions still hold in the general case, i.e., when each company chooses its ordering scheme.

8.4.4 Analysis of Simulation Outcomes with six players: The heterogeneous supply chain

We now study the most realistic scenario, which is also the most complex, in which each company acts alone. More precisely, we now consider six different selfish company-players, whose goal is to minimize their own costs, without directly considering the

Market consumption patterns	Lumber Retailer			Paper Retailer			RestOfThe SupplyChain		
	A″	B	D	A″	B	D	A″	B	D
1. Step	1	-	-	1	-	-	1	-	-
2. Inversed step	1	-	-	1	-	-	1	-	-
3. Dirac	1	-	-	1	-	-	1	1	-
4. Inversed Dirac	1	-	2	1	-	2	1	1	-
5. Increase	1	-	-	1	-	-	1	-	-
6. Decrease	1	-	-	1	-	-	1	-	-
7. Weak seasonality	1	2	-	1	2	-	1	1	-
8. Medium seasonality	1	3	-	2	2	-	1	1	-
9. Strong seasonality	2	-	3	2	-	1	1	1	-

Table 8.13: Strictly dominated strategies when the LumberRetailer, the PaperRetailer and the RestOfTheSupplyChain are selfish.

impact of their behaviour on the costs of the rest of the supply chain. However, each company indirectly takes into account its impact on the supply chain. In fact, if a company amplifies the bullwhip effect, stockouts will occur more often in the whole supply chain, and also in the company itself, which will cost more money for the considered company.

This study is similar to what was done in the two previous subsections, except that many more combinations of the three ordering schemes are considered. In fact, we simulate $3^6 = 729$ combinations of A″, B and D among the six companies/players, and we put the outcomes of these simulations in a $3 \times 3 \times 3 \times 3 \times 3 \times 3$ matrix that represents the game in the normal form. Since it is very difficult to study such big games by hand, we use the software Gambit to remove dominated ordering schemes and then to look for Nash Equilibria. With this game, we address the following questions:

1. Which ordering schemes are dominated, and thus, never used?
2. Which joint strategies minimize the total supply chain costs?
3. Which joint strategies are Nash equilibria, i.e., are composed of individual strategies that each company wants to keep when it sees the others' behaviour?
4. Is it possible to order the Nash equilibria with the relation of Pareto-domination?

The most interesting joint strategies are those found at the same time in points 2. and 3., which we call "best joint strategies". Let us insist that they are both (a) the overall

	Step	Inversed Step	Dirac
Minimum of C	(D, D, D)→109 +208+915=1,232 k$	(D, D, D)→30 +43+399=472 k$	(D, B, D)→44 +90+288=422 k$
Nash equilibria	(D, D, D) →109+208 +915=1,232 k$ (B, B, B) →173+198 +1,733=2,104 k$	(D, D, D)→30 +43+399=472 k$ (B, D, D)→30 +43+706=779 k$ (B, B, B)→30 +48+735=813 k$ (D, B, B)→30 +48+735=813 k$	(D, B, D) →44+90 +288=422 k$ (B, D, D) →61+28 +446=535 k$
Full collaboration	(D, D, D)→109 +208+915=1,232 k$	(D, D, D)→30 +43+399=472 k$	(D, D, D)→48 +95+313=456 k$

Table 8.14: Results of analysis when the LumberRetailer, the PaperRetailer and the RestOfTheSupplyChain are selfish (1/3).

best choices because they minimize the total supply chain costs, and (b) no company wants to change for another ordering scheme because they are Nash equilibria.

In comparison with the previous three subsections, the situation is made more complex by the increase of the number of players. Therefore, conclusions of experiments are less clear than in the three previous subsections. We now present the results performed by Gambit.

First, iteratively dominated strategies are represented in Table 8.17, in which, numbers represent the turn where the corresponding rule is eliminated because it is dominated by another one. This figure shows the following conclusion:

> Companies sometimes have individual incentives to collaborate (because the non collaborating rule A″ is sometimes dominated), but they do not have an incentive not to collaborate (because collaborating schemes B are dominated only under the Step demand).

After the iterative elimination of strictly dominated strategies, emphasis is put on Nash equilibria. Tables 8.18, 8.19 and 8.20 present these equilibria. The format of data is $(r^1, r^2, r^3, r^4, r^5, r^6) \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$. For example, in the column Step in Table 8.18, the first line, called “Minimum of C”, shows that the joint strategy (D, D, D, D, D, D) minimizes the total supply chain costs C when the market consumption is Step. This minimal overall cost is 1,232 k$. Since it is the minimum

	Inversed Dirac	Increase	Decrease
Minimum of C	(B, D, D)→49 +100+262=411 k$	(B, B, B)→8 +17+491=516 k$ (D, B, B)→8 +17+491=516 k$	(D, D, D)→57 +82+595=734 k$
Nash equilibria	(B, B, D)→53 +89+272=414 k$	(B, B, B)→8 +17+491=516 k$ (D, B, B)→8 +17+491=516 k$ (D, D, D)→8 +32+489=529 k$	(D, D, D)→57 +82+595=734 k$ (B, D, D)→57 +82+915=1,054 k$ (D, B, B)→57 +82+932=1,071 k$ (B, B, B)→57 +82+932=1,071 k$
Full collaboration	(D, D, D)→58 +102+265=425 k$	(D, D, D)→8 +32+489=529 k$	(D, D, D)→57 +82+595=734 k$

Table 8.15: Results of analysis when the LumberRetailer, the PaperRetailer and the RestOfTheSupplyChain are opponents (2/3).

	Weak seasonality	Medium seasonality	Strong seasonality
Minimum of C	(D, D, D)→68 +90+197=355 k$	(D, D, D)→91 +135+1,016=1,242 k$	(D, D, D)→162 +239+1,659=2,060 k$
Nash equilibria	(D, D, D)→68 +90+197=355 k$	(D, D, D)→91 +135+1,016=1,242 k$	(B, B, D)→142 +210+1,990=2,342 k$
Full collaboration	(D, D, D)→68 +90+197=355 k$	(D, D, D)→91 +135+1,016=1,242 k$	(D, D, D)→162 +239+1,659=2,060 k$

Table 8.16: Results of analysis when the LumberRetailer, the PaperRetailer and the RestOfTheSupplyChain are opponents (3/3).

of C (of course, because this line only gives minima of C), this cost is underlined. We can also read that the LumberRetailer has a cost C^1 =109 k$, the PaperRetailer C^2 =208 k$… This minimum of C only occurs with (D, D, D, D, D, D) because there is only this joint strategy in this entry of Table 8.18.

In the second line of Tables 8.18, 8.19 and 8.20, all Nash equilibria are listed. The two equilibria for the Step pattern are (D, D, D, D, B, D), which only is a good joint strategy because it both incurs a C close to the minimum of C and is a Nash equilibrium, and (D, D, D, A”, D, D), which is not a good strategy because it incurs a C much greater than the lowest. If the first equilibrium (D, D, D, D, B, D) had incurred $C = 1,232$ k$ instead of $C = 1,491$ k$, it would have been a best joint strategy.

	LumberRet.			PaperRet.			LumberWhl.			PaperWhl.			PaperMill			Sawmill		
	A''	B	D	A''	B	D	A''	B	D	A''	B	D	A''	B	D	A''	B	D
1. Step	1	5	-	1	-	-	3	-	-	-	-	-	3	-	-	2	4	-
2. Inversed step	1	-	-	1	-	-	-	-	-	1	-	-	-	-	-	1	-	-
3. Dirac	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4. Inversed Dirac	1	-	-	1	-	-	1	-	-	-	-	-	-	-	-	-	-	-
5. Increase	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6. Decrease	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-	1	-	-
7. Weak seasonality	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8. Medium seasonality	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9. Strong seasonality	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 8.17: Strictly dominated strategies when all companies are selfish (six player game).

	Step	Inversed step	Dirac
Minimum of C	(D, D, D, D, D, D) →109+208+75 +159+265+416 =1,232 k$	(D, D, D, D, D, D) →30+43+51 +74+101+173 =472 k$	(D, D, B, D, D, D) →48+75+53 +74+85+61 =396 k$
Nash equilibria	(D, D, D, D, B, D) →115+326+82 +277+186+505 = 1,491 k$ (D, D, D, A'', D, D) →109+888+75 +47+1,900+289 =3,308 k$	(D, D, D, D, D, D) →30+43+51 +74+101+173 =472 k$ (B, D, B, D, D, B) →30+43+93 +74+101+199 =540 k$ (D, B, D, B, B, B) →30+48+51 +140+260+236 =765 k$ (B, B, B, B, B, B) →30+48+93 +140+260+242 =813 k$	(B, D, D, D, B, D) →44+83+45+83 +87+67=409 k$ (D, B, D, D, D, D) →44+90+45+90 +84+69=422 k$ (A'', D, A'', D, D, B) →663+93+1,551+92 +102+2,788=5,289 k$ (A'', B, A'', D, B, B) →709+98+1,778+98 +104+2,675=5,462 k$ (D, A'', B, D, D, A'') →396+480+411+1,669 +1,936+1,090=5,982 k$ (B, A'', B, A'', B, B) →61+945+61+5,826 +4,584+4,452=15,929 k$
Full collaboration	(D, D, D, D, D, D) →109+208+75+159 +265+416=1,232 k$	(D, D, D, D, D, D) →30+43+51+74 +101+173=472 k$	(D, D, D, D, D, D) →48+95+53+94 +110+56=456 k$

Table 8.18: Results of simulation analysis when all companies are selfish (1/3).

Finally, the line called "Full collaboration" exhibits the costs of (D, D, D, D, D, D). In the case of the Step demand, these costs were already given in the first of the tableau, as it should often happens because full collaboration should incurred the minimum of C. But in general, the costs of (D, D, D, D, D, D) only appears in the last line.

	Inversed Dirac	Increase	Decrease
Minimum of C	(B, B, D, B, B, D) →46+73+46 +73+74+59=371 k$	(D, B, D, B, B, B) →8+12+19 +40+90+311=480 k$	(D, D, D, D, D, D) →57+82+95+150 +206+144=734 k$
Nash equilibria	(B, B, D, B, B, D) →46+73+46 +73+74+59=371 k$ (B, B, B, D, B, D) →54+85+54 +85+80+62=420 k$ (B, B, B, B, D, D) →53+85+53 +85+85+68=429 k$	(B, B, B, B, B, B) →8+17+28+50 105+308=516 k$ (D, D, D, D, D, D) →8+32+19+37 +69+364=529 k$ (B, D, B, D, B, D) →8+29+28+32 +136+336=569 k$ (D, A", D, D, D, A") →423+785+452+2,511 +2,077+1,438=7,686 k$ (B, A", B, D, B, A") →564+938+610+3,699 +3,250+1,622=10,683 k$	(D, D, D, D, D, D) →57+82+95+150 +206+144=734 k$ (B, D, B, D, D, B) →57+82+139+150 +206+189=823 k$ (D, B, D, B, B, B) →57+82+104+204 +311+266=1,024 k$ (B, B, B, B, B, B) →57+82+141+204 +312+275=1,071 k$
Full collaboration	(D, D, D, D, D, D) →58+102+55 +97+75+38=425 k$	(D, D, D, D, D, D) →8+32+19 +37+69+364=529 k$	(D, D, D, D, D, D) →57+82+95+150 +206+144=734 k$

Table 8.19: Results of simulation analysis when all companies are selfish (2/3).

Let us now draw some conclusions from the outputs of Gambit. We can first notice in Table 8.17, that non-collaborating (A") is sometimes dominated while collaborating (B and D) is almost never dominated. Tables 8.18, 8.19 and 8.20 show that there exists at least one Nash equilibrium for all demand patterns, and there are no more than six equilibria per market consumption pattern. Moreover, these figures also show, that full collaboration (D, D, D, D, D, D) is both a Nash equilibrium, and incurs the minimum of C (i.e., it is the best joint strategy) under three demand patterns, i.e., with Decrease, Weak seasonality and Medium seasonality. Moreover, when (D, D, D, D, D, D) is not a Nash equilibrium or does not incur the minimum of C, there is always an equilibrium incurring a C very near the minimum, and in which most companies use D and the others B (however, this is not true only with the Increase demand, because (B, B, B, B, B, B) incurs lower costs than (D, D, D, D, D, D)). Such an equilibrium in which most companies use D is thus a “good joint strategy”. As in the previous subsection, we can therefore say that:

> No company would like the entire supply chain to collaborate (by applying Scheme B or D) while the considered company does not collaborate (by

	Weak seasonality	Medium seasonality	Strong seasonality
Minimum of C	(D, D, D, D, D, D) →68+90+40+66 +50+41=355 k$	(D, D, D, D, D, D) →91+135+191+271 +324+230=1,242 k$	(D, D, D, D, D, D) →162+239+294+415 +522+428=2,060 k$
Nash equilibria	(D, D, D, D, D, D) →68+90+40+66 +50+41=355 k$	(D, D, D, D, D, D) →91+135+191+271 +324+230=1,242 k$ (A", D, A", D, D, D) →333+131+941+297 +367+1,192=3,261 k$ (D, A", D, A", B, B) →163+890+290+3,967 +4,530+3,688=13,528 k$	(B, B, D, D, D, D) →142+210+336+474 +624+556=2,342 k$
Full collaboration	(D, D, D, D, D, D) →68+90+40+66 +50+41=355 k$	(D, D, D, D, D, D) →91+135+191+271 +324+230=1,242 k$	(D, D, D, D, D, D) →162+239+294+415 +522+428=2,060 k$

Table 8.20: Results of simulation analysis when all companies are selfish (3/3).

using A"), even if a few companies use B instead of D: all companies prefer collaboration.

In this result, it is important to recall that Scheme A" is based on a mathematical model minimizing the cost of a single company when some assumptions are met, while Scheme B and D do not have this formal base, but are based on collaboration through information sharing to reduce the bullwhip effect. Moreover, the costs incurred by A" are higher to costs incurred by B and D by *several orders of magnitude*, as reflected, for example, in Table 8.3.

Finally, when there are several Nash equilibria, the Nash equilibrium incurring the lower C always Pareto-dominates all other equilibria, except for one player with one demand pattern: with Increase, the PaperWholesaler prefers (D, D, D, D, D, D) in which $C^4 = 37$ k$ and (B, D, B, D, B, D) in which $C^4 = 32$ k$, to (B, B, B, B, B, B) in which $C^4 = 50$ k$. We do not have any reason to explain this irregularity, but we can only note that this is an addition to the fact that (B, B, B, B, B, B) incurs a lower C than (D, D, D, D, D, D).

8.4.5 Comments

First, we only consider pure strategies in this dissertation, that is, companies use the same ordering scheme in each of the fifty weeks of simulation. If mixed strategies had been used, we would have used probabilities for all these rules, e.g., the PaperRetailer would use A″ during 75% of the time, and B during the remaining time. In fact, we cannot use mixed strategies here, because:

1. Our first simulation does not allow changing companies' ordering rule during a simulation (but our second simulation, presented in section 6.3, can take this constraint into account). Therefore, we cannot study supply chain transition, when a company switches from one ordering scheme to another one. In fact, we are not allowed to calculate the expected outcome of a simulation with mixed strategies based on two simulations with pure strategies, because the result would have no meaning in reality. For example, if the PaperRetailer uses A″ 75% of the time and B during the remaining time, the expected PaperRetailer's outcome is not 75% of its outcome in simulation in which only A″ is used, plus 25% of its outcome in a second simulation in which only B is applied. This method is used in traditional game theory, but it is not appropriate here, because of the transition time between the two ordering rules: when the PaperRetailer switches to B, products previously ordered with A″ will still arrive, which has an impact on the PaperRetailer's inventory level, and therefore on its future orders. In short, there is an inertia in the supply chain, when companies switch ordering schemes.

2. In addition to the first point, we do not know algorithms to determine Nash equilibria in mixed strategies requiring a reasonable time for any of our games. The determination of Nash equilibria in pure strategies requires the comparison of individual outcomes, while in mixed strategies, it requires the resolution of linear equations, a more complex task. For example, we have run Gambit on the games induced by our nine demand patterns, and we have stopped the computation after a week on a 2.7 GHz PC, while the computation was not finished!

   We have seen in section 4.3 that questions related to Nash equilibria are considered to be harder than P and easier than NP. In other words, they are considered to be harder than easy problems, and easier than hard problems. Nevertheless, questions related to the computation of Nash equilibria are still being researched, even though some game-theoretic softwares such as Gambit exists.

Experiments indicate that not collaborating is sometimes a dominated strategy while collaborating is never dominated, which means that, in general, companies should collaborate if they want to reduce their logistics costs. Then, full collaboration is, or is very near the minimum of the overall cost C, and is also often a Nash equilibrium, which means that no company would like to stop collaborating. Moreover, it is very easy to identify the desirable Nash equilibria: when these equilibria do not correspond to full collaboration, they are very near this, that is, only one or two companies use B instead of D. Another good point is the fact that equilibria very loosely depend on the market consumption pattern. As we analyze simulations as static games, each time the market changes the way it consumes, the supply chain has to change its joint strategy because the former joint strategy is no longer an equilibrium. Therefore, the experiments show that all or most companies in the supply chain will use D with all demand patterns, and a few may sometimes use B for any market consumption.

Moreover, there is always a Nash equilibrium, which is not always the case in game theory when pure strategies are considered. When there are several equilibria, at least one equilibrium is very near the minimum of C, and most companies use D and few use B in this equilibrium. If we focus in future work on how companies decide during a fifty week simulation which ordering scheme to use, all companies may eventually stabilize their choice on one of our Nash equilibria. During these future simulations, the PaperRetailer would, for example, use B during six weeks, then decide, with an unspecified algorithm, that it will now use A″. Even if we do not have this algorithm of ordering scheme selection, we already know that companies may eventually stop switching, because there is always at least one Nash equilibrium. However, they may also switch indefinitely, if they do not find this equilibrium, or if they do not recognize it, i.e., if their algorithm of ordering scheme selection is not well designed.

When there is only one Nash equilibrium, we know that the global cost C is the minimum, or is very near this minimum. The problem for companies is that they do not know the number of Nash equilibria. They only know they have found one because no companies change the way they place orders. If we look at Tables 8.18, 8.19 and 8.20 again, we can remark that Nash equilibria in which no companies use A″ often have a C near the minimum, while equilibria in which some companies use A″ always occur a much higher C than the minimum. In other words, the algorithm for switching between A″, B and D should only consider B and D, because this insures (i) finding an equilibrium quicker, because fewer alternatives are tried and, (ii) finding an equilibrium incurring low overall and individual costs in the same time.

Another point concerns cost evaluation: as long as backorders cost twice the price of inventory holding, the way to calculate individual costs C^i (C^i_{QWSG}, C^i_{improved} or $C^i_{\text{realistic}}$)

has no impact on dominations[5], Nash equilibria and Pareto-domination. This is due to the fact that these concepts only rely on cost comparison for the same company, thus are not sensitive to multiplication by a constant. For example, the company has the costs $C_{\text{QWSG}} = \$5$, $C_{\text{improved}} = \$5 * IM$ and $C_{\text{realistic}} = \$5 * RE$ with Scheme B and $C_{\text{QWSG}} = \$10$, $C_{\text{improved}} = \$10 * IM$ and $C_{\text{realistic}} = \$10 * RE$ with Scheme D (where IM and RE are two positive constants that characterize the method of cost calculation): whatever method of cost calculation is used, the cost incurred by B is always lower than the cost incurred by D.

However, such a multiplication impacts on the global cost C, and therefore, changing this method of calculation will change the joint strategies incurring the minimum of C. Moreover, initial inventory levels also rely on C. If the relative weights of company's costs change (e.g., costs are identical for all companies instead of having higher costs for clients than for suppliers due to logistics costs), all initial conditions also change.

Finally, we insist on the interpretation of outcomes in our experiments, and in particular Nash equilibria. Such outcomes represent costs for fifty week simulations. In other words, when companies consider them, they only base their decision on long term costs. This means that Nash equilibria are stable states for the supply chain, when all companies use the same time horizon. If a company considers a shorter term, it could have an incentive to leave such equilibria. Fortunately, collaboration is often seen as long term, because it requires companies to agree and to install some collaboration devices, such as information technologies. This long term view required by collaboration is in *conformance* with the duration of our simulations. The next section extends this discussion about our conclusions.

8.5 Practical Implications

The main conclusion of the second series of experiments presented in this chapter can be stated as follows: *companies prefer to use our collaborative ordering schemes*. More precisely, companies prefer basic information sharing than no information sharing, and they also prefer information centralization to basic information sharing. In practice, this means that retailers would like to share the market consumption, and other companies wish to get this information. We now discuss techniques and technological tools supporting such information sharing, and in particular information centralization.

---

[5] However, the method of cost evaluation changes the initial inventory levels, that is, the initial conditions of each set of 3^6 simulations, which may affect results.

As presented in Subsection 2.1.1, information centralization is already implemented in techniques such as VMI (Vendor Managed Inventory), CRP (Continuous Replenishmend Program) and CPFR (Collaborative Planning, Forecasting and Replenishment), but on a few levels of a supply chain (i.e., between some companies and their suppliers). The results of our dissertation show that these techniques should be extended to the rest of supply chains. In fact, our results show that applying information centralization only by retailers and wholesalers never induces the minimum of the overall supply chain cost C, or is a Nash equilibrium. On the contrary, applying information centralization by all companies is a Nash equilibrium incurring the minimum of the overall supply chain cost C.

Of course, these collaboration techniques are currently based on the Internet, but some of them are also based on older technologies, such as EDI (Electronic Data Interchange). On the Internet, a technology called “e-Hub” looks interesting as a support to information centralization. We have seen, that information centralization is the best method to follow our two principles to reduce the bullwhip effect by our two principles, because Scheme D gives better results than B. Basically, an e-Hub can be seen as an electronic blackboard, with which retailers share the market consumption. More precisely, Lee [2001] presents it as a node in the data network, that *instantaneously* and in *real-time* processes and forwards all relevant information to all appropriate nodes, and in particular, to all appropriate companies. An e-Hub can store data, process information, and publish information in pull and/or push mode, as illustrated in Figure 8.3.

Concerning our approach, retailers can automatically multicast market consumption in the rest of the supply chain through an e-Hub. This multicast would be instantaneous and in real-time, as required by the information centralization used in our Scheme D. From a more general viewpoint, e-Hub is designed to increase supply chain integration, i.e., coordination and collaboration approach between supply chain partners to stay competitive and enlightened. The four critical dimensions of this way to coordinate and collaborate are information integration, synchronized planning, coordinated workflow and new business models. To achieve these four points, companies’ information systems, e.g., companies’ ERP as introduced in Subsection 2.3.1, are linked together through these e-Hubs to form a “hub-and-spoke” system at the level of the supply chain [Lee, 2001].

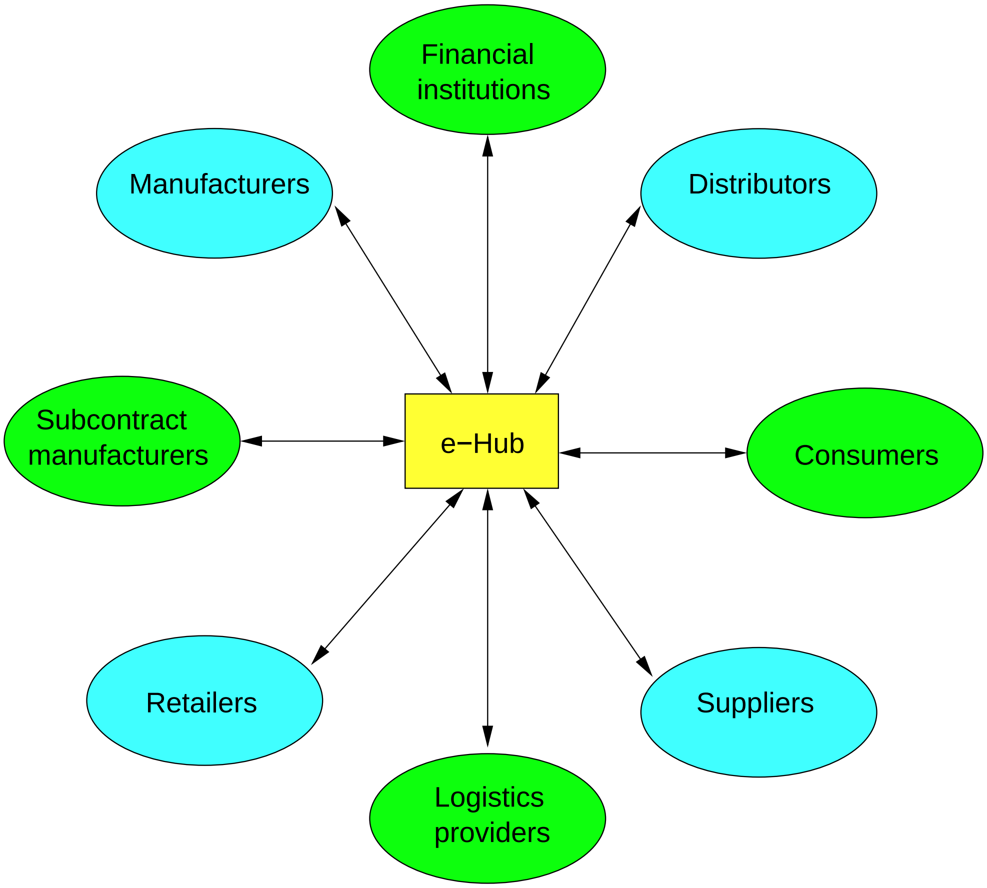


Figure 8.3: The e-Hub model [Lee, 2001].

8.6 Conclusion

The first series of experiments, as presented in the previous chapter, has assumed that all companies were using the same ordering scheme. This assumption is relaxed in the second series of experiments presented in this chapter. By doing so, we have verified if our two principles allow companies to minimize supply chain costs and individual costs. Thus, the reduction of the bullwhip effect was the metrics in the first series of experiments, while costs were the metrics in this chapter.

This is the reason why costs in the QWSG were adapted to the Québec forest context, and parameters of ordering schemes were optimized to minimize these costs. The crucial question was to check, if companies were going to cooperate to reduce the bullwhip effect, or if each company would like the whole supply chain to cooperate except itself, which would eventually lead to total absence of collaboration in the supply chain.

Fortunately, we find that all companies have incentives to collaborate, because basic collaboration (Scheme B) and improved collaboration (Scheme D) are in the Nash equilibria that are the most efficient for the overall supply chain, and such equilibria Pareto-dominate other equilibria.

Chapter 9

Conclusion

9.1 Summary

In this dissertation, we have addressed the issues related to the interactions of autonomous entities, and more precisely, how such entities can obtain and maintain coordination in order to reduce stream fluctuations in distributed systems. The particular case of distributed system on which we have focussed, is the supply chain modelled as a multi-agent system. In the context of supply chains, stream fluctuations are known as the bullwhip effect, which consists in the amplification of the demand variability. The bullwhip effect is an issue of supply chain management, in which the variability of orders placed by companies is amplified. This variability incurs costs due to higher inventory levels and agility reduction. This cost has been estimated at 40-60 million USD for a 300 kton paper mill in the north of Europe.

By considering the special case of supply chains, this reduction gives us some hints in return to reduce stream fluctuations in other types of distributed systems, such as multi-agent systems. For that purpose, this thesis has been divided into two parts. In the first part, we have proposed to see delays as a cause of stream fluctuations, and we have proposed two principles to reduce the impact of this cause. Based on these two principles, we have then designed a decentralized decision making process to reduce the bullwhip effect. The reduction of the bullwhip effect induced by this mechanism was validated in a first series of experiments.

The second part of this thesis has checked with game-theoretic concepts, if all companies have individual incentives to follow our two principles. In fact, it is not because a coordination mechanism is globally efficient that individuals agree to use it, because coordination may have a price, and some companies may not earn enough to cover this price. We have verified here that companies should follow our two principles because each company increases its costs by deviating from them.

Notice that the principles and the approaches in these two parts are specific neither to some specific supply chains, nor to some particular multi-agent systems. In fact, only some elements of our two agent-based simulations are specific to the field of supply chain management, and only the supply chain model and the method to calculate costs in the supply chain are particular to the Québec wood industry. For instance, we have shown that our solution to the bullwhip effect can often be translated into solutions for any distributed system, and in particular, in some other multi-agent systems. This is due to the fact that companies and agents share certain properties, such as their autonomy.

We now detail these two parts of our work. Let us recall that both series were carried with our first simulator and that our second simulator is still under development.

9.1.1 Decentralized Coordination Mechanism

In the first part of this thesis, two decentralized decision processses (i.e., two ordering schemes in the context of supply chains), based on information sharing, have been proposed to minimize the bullwhip effect, without neglecting the importance of inventory management and operational constraints. These ordering schemes were based on two principles:

- The lot-for-lot ordering policy eliminates the bullwhip effect, but does not manage inventory;
- Companies should react only once to each market consumption change by over- or underordering.

These two principles explain why it is important that companies share their information (i.e., companies have to know market consumption to be able to react to its changes), and how to use the shared information (i.e., by over- or under-ordering exactly once per market consumption change to stabilize inventory level).

The first series of experiments has validated these conceptual propositions. This validation used a simulator inspired by the Québec forest industry, which is based on the model of the Québec Wood Supply Game (QWSG). This game is derived from the Beer Game, which is a board game designed to teach the bullwhip effect. Each company in this simulator is seen as a reflex agent that applies a given ordering rule. We compared the reduction of the bullwhip effect incurred by our two ordering schemes with five other rules. The second of our two schemes is an improvement on the first one through the addition of information centralization. Information centralization consists in the multi-casting by retailers of their sales in real-time to the rest of the supply chain. This is an acceleration of the information sharing implied by our two principles, and implemented in our first scheme, because the market consumption is as slow as order in our first scheme while it is instantaneous in our second scheme.

Two main points can be drawn from the simulation outputs of this first series of experiments. The first point is that the bullwhip effect is reduced as much as is possible, but it cannot be totally eliminated, because a little amplification of the demand should remain in order to avoid stockouts and overstockings, due to ordering and shipping delays. This first point led us to propose that delays are also a cause of the bullwhip effect, although they were only considered as an aggravating factor for another known cause of the bullwhip effect. The second point to note is that our ordering scheme transforms the behaviour of the supply chain. In fact, our two ordering schemes make it so: retailers are disturbed for a longer period by inventory variations than upstream suppliers (e.g., the forest), while order variability has almost disappeared. On the other hand, the classic problem of the bullwhip effect is the contrary: retailers are less disturbed by order variation than upstream suppliers.

9.1.2 Individual vs. Common Interest for Coordination

In the second part of this thesis, we check whether each company-agent in the simulation has an individual incentive to fulfill our two principles by using one of our two ordering schemes. While the first part only focussed on the bullwhip effect reduction, the second part addressed financial benefits of the bullwhip effect reduction. This explains why we first apply costs from the Québec forest industry to our simulation model. Next, we set parameters in the three ordering rules considered in our second series of experiments. In fact, companies in the considered scenario had to choose among three ordering rules: (i) a classic (s, S) policy, (ii) our first ordering scheme, that requires information sharing only between each company and its supplier, and (iii) our second scheme, that requires information centralization. The parameters to set up in the first scheme are s and S, and in our two ordering schemes the initial inventory level and the quantity of products to over- or underorder at each market consumption change. Note that these three schemes were also considered in the first part, and the other four schemes have been dropped. Then, concepts from game theory concepts were adapted to our context.

Finally, the second series of experiments was achieved. A simulation was carried out for each combination of the three ordering schemes among the six companies in the QWSG, and under nine market consumption patterns, that is, $9 * 3^6 = 6,561$ simulations were carried out. Since the outcome of each of these 6,561 simulations is nine individual company's costs, a free software called Gambit was used to analyze this huge quantity of data. This software showed that using our two ordering schemes never results in dominated strategies, while using the (s, S) policy is quite often dominated. We have also looked for Nash equilibria. For half of the considered market consumption patterns, there is only one Nash equilibrium, which also incurs the lowest cost for the entire supply chain. In fact, another result could have occured, in which each company would have preferred the whole supply chain to collaborate (by using one of our two ordering schemes), except itself, or, in other words, the best solution for the group would not have matched the best individual solution. This is not the case here, because every company has an incentive to collaborate by using one of our two schemes, which is a very good result.

9.2 Discussion on our Methodologies

In this thesis, we have used the two methodologies that we have just recalled. The first methodology was used in the first part of our work, and was the proposition of two principles to design decentralized coordination mechanisms that do not induce stream fluctuations. The second methology is at a different level than the first one, because it deals with how to study that every agent has individual incentives to adopt our two principles. In other words, the second methodology is used to verify whether agents will agree to behave according to the first methodology. We now detail these two methodologies.

9.2.1 Methodology to Propose a Decentralized Coordination Mechanism

The first methodology deals with the coordination required to reduce the bullwhip effect, and is summarized in our two principles. The goal of these two principles is to guide the emergent behaviour of a supply chain towards a stable one. In summary, the first methodology, and thus our two principles, uses communication to coordinate the entities of distributed systems so that such systems are more stable. The experimental results in Chapters 7 and 8 show that incurred costs can be decreased by several orders of magnitude when company-agents agree to share market consumption information. Therefore, we have focussed on *why* to share information and on *how* to use the shared information, rather than on the optimization of the parameters used in the coordination mechanisms that instantiate our two principles. Only after we have finished designing this methodology, we were able to perform the optimization of its parameters, as we have done with the Solver in Chapter 8 (detail in Sections C.1 and C.2).

From a higher viewpoint, we can notice that coordination is one of the main issues in the field of multi-agent systems, and thus, our first methodology is worth being applied to other multi-agent applications. Indeed, we have seen in Table 3.1 that stream fluctuations may affect several kinds of distributed systems (different kinds of networks, multi-agent systems, economy. . .), and that the bullwhip effect is only one instance of the generic problem of stream fluctuations. In particular, we have shown in Subsection 5.6.2 how to apply our two principles to avoid the "slinky effect" in car platoons in intelligent highway systems.

To introduce our coordination-based mechanism, we have presented in Subsection 4.1.2 how the multi-agent community has faced this problematic. In particular, Boutilier [1996] proposed to classify coordination techniques into three classes, which are used by many authors. These three broad classes are communication-based coordination, convention-based coordination and learning-based coordination. Concerning the first class, communication in multi-agent systems provides agents with a better knowledge about the context in which they make decisions. In particular, Bond and Gasser [1988] note that communication allows agents to have an improved understanding of the goals, plans and activities of ther agents, of the problem domain, and of the temporal context (e.g., greater lookahead or history). That is, the coordination in a multi-agent system may be improved if agents do not rely only on local information, but if they also have a global view on the multi-agent system. We can note here that our two principles make agents communicate to have such an improved coordination. But the system may still lack of coherence, that is, it may lacks behaving like a unit, because of information transmission lead time. To improve more the coordination of the multi-agent system, we added information centralization, which makes all agents have the same information at the same time.

Bond and Gasser [1988] also note that "knowing when something (information about the external world, or information about the agent's internal state) has changed sufficiently to notify another agent is tricky". We can relate this sentence to our second principle, which states that companies should react only once to each market consumption change, or with multi-agent words, agents should react only once to each change in the environment. Indeed, this principle is easy to apply in our simulations, but in real life, some thresholds of the market consumption have still to be defined and known by everyone, which is tricky.

9.2.2 Methodology to Check That our Two Principles Will be Adopted

The second methodology uses multi-agent simulations to build games that we next analyze with game-theoretic concepts. The idea is to follow these three steps:

1. simulate all combinations of the possible choices among all agents (i.e., each agent had three possible choices, and there were six agents in Chapter 8);
2. put every simulation output in an entry of a matrix;
3. analyze this matrix as a game in the normal form.

As we can see, this methodology can be used to study a large variety of issues of incentives, and more generally, to all problems of interactions that we are able to simulate. The addition of multi-agent simulation to game theory allows enhancing the strengths of this theory to study much more complex interactions. We now discuss the respective advantages of analytical game theory (i.e., traditional game theory) and of simulation-based game theory (i.e., our second methodology).

The strengths of game theory are to provide a wide range of tools (representation of interactions as games, concepts of solution…) to study interactions. Traditionnally, all these tools have been used in analytical approaches. The advantage of analytical approaches is to be very pleasant because they allow finding solutions (Nash equilibria, Pareto-efficient joint strategies…) analytically, and next, to comment easily these analitycal results. In particular, singularities, such as discontinuity or particular cases, are easy to identify.

However, it is not always possible to represent interactions analytically, because dynamics are too complex. For example, it is very hard to represent the dynamics of a supply chain with a little set of equations because of the discontinuity of streams (i.e., stockouts may occur, which forces using what-if conditions in equations). This is the reason why the models in the literature only consider two levels of a supply chain, that is, one or several client(s) with one or several supplier(s) (cf. the models called "joint optimization" in Subsection 4.1.1). In such cases, our methodology can be used, because simulations are able to study much more complex dynamics than analytical approaches. But this ability to study more complex situations has also a price, that is due to the replacement of the analytical description of interactions by a simulation model. More precisely, since we rely on simulations, we are only able to run some specific scenarios. As a consequence, to study the influence of one specific parameter, we must carry out simulations for several values of the considered parameter. Furthermore, the interesting values of this parameter (discontinuity...) are not obvious. On the contrary to simulation-based approaches, the impact of every parameter is obvious with analytical approaches.

In short, using traditional game theory is more pleasant because we can prove results analytically, while using game theory with simulations (i.e., our second methodology) allows considering much more complex/realistic dynamics, but conclusions drawn from simulations are harder to generalize.

Finally, we can note that game theory is more and more used in computer science, but only a few softwares as Gambit are still available, and algorithms to compute the solutions described analytically by concepts of solutions are still under research, in particular to improve their efficiency.

9.3 Future Work

Many extensions of this work are possible. First, we have outlined some of them during this dissertation. For example, we have pointed out in Section 8.1, that the optimizations of the parameters of the ordering schemes were only performed for homogeneous supply chains, which thus incur the minimum of the overall supply chain cost. To fix this issue, we have proposed a method to optimize parameters for heterogeneous supply chains as well (cf. algorithm 8.4 in replacement of Algorithms 8.2 and 8.3), but the computation time will increase much.

Another example of extension proposed during this dissertation concerns the market consumption patterns. Except in Chapter 7 (patterns 10A to 10J were ruled by a statistic distribution), we have not considered that such patterns are stochastic. Although, such stochastic patterns correspond to the reality, and should thus be used in our simulations. Of course, this would much increase the number of simulations to carry out, in order to calculate the average and standard-deviation of our four metrics (standard deviation of placed orders, costs, number of backorders, and standard deviation of inventory level) for each instance of a stochastic market consumption pattern.

This would also much increase the computation time to find Nash equilibria. Moreover, Nash equilibria may be different for each instance of a stochastic pattern, but we think that this will not occur, because we hope that our results are robust.

We now propose some additional questions to extend our work.

9.3.1 How to Make Agents Take into Account their Impact on Other Agents?

The basic question

The first possible future work concerns the evaluation of the efficiency of company-agents. In fact, we have used two main metrics. In the first part of the thesis, we have considered the standard deviation of orders (however, we also consider three other metrics, i.e., costs, backorders and variation of inventories), because it is a direct measure of the bullwhip effect. In the second part of this dissertation, we used costs, which are an indirect measure of the bullwhip effect. While the first metric is accurate, because it is a fluctuation measure, and the bullwhip effect is the fluctuations of orders, the second metrics may be improved. Precisely, we could adapt our simulation model to assume company-agents want to maximize their utility/profit, instead of minimizing costs. In fact, minimizing costs may cause company-agents to miss some opportunities, because, for example, they do not invest money (which is a cost), while it should increase their profit. Moreover, this replacement is in conformand with game theory, and economics in general, in which agents are defined as utility maximizers, which corresponds to company-agents that maximize their profit.

In this case, costs would take production and inventory activities into account, and could be non-linear, e.g., in order to represent economies of scale. The supply chain would earn money each time it sells a product to the market, and lose sales when market consumption exceeds the quantity of products available by retailers. Backorders in the supply chain would no longer cost money: the only goal of each company would be that retailers have enough products. Therefore, the goal of companies would be quite different: instead of aiming at zero inventory and zero backorders, companies would try to maximize retailers' sales, while minimizing their own inventory, and taking advantage of non linear costs. Companies would only want retailers have no backorders and no longer take care of other companies' backorders. In fact, companies do not wish retailers to sell as much product as possible, but wish to maximize their utility.

How to Apply Mechanism Design to our Context?

As previously stated, company-agents are selfish, and thus, have difficulty taking into account that they are more profitable when the rest of the supply chain is more efficient. A method to make them consider the goal of the entire supply chain, i.e., to deliver products to end-customers, and thus, to have products available for retailers, is to incite them not to have too many backorders. To this end, Subsection B.4.2 ends on a related question: How will company-agents behave if we change the relative weight of inventory holding and backorders. For example, if backorders are considered as costless, are company-agents going to stop collaborating? Therefore, a possible set of experiments could study the price of backorders, where the optimum price would be defined either as the one in which products available by retailers exactly match the market demand, or as the one that maximizes the global supply chain benefit. In other words, we have to find the balance between two extremes:

- If backorders cost nothing, each agent only tries to minimize its inventory level. In fact, each agent prefers that retailers lose sales, than to pay for holding inventory.
- If backorders have an infinite cost, each agent only tries to avoid stockout. In this case, each agent has large inventory levels.

One interesting question here, is to know if the best price of backorders is the same for every agent, or if suppliers should pay a higher price than their clients. This question looks strongly related to Porteus' Responsibility Tokens presented in Subsection 4.1.1. We recall that responsibility tokens are used to evaluate the financial penalty to pay, when a backorder occurs. More precisely, when a company cannot ship the demanded products, it replaces the missing products by some responsibility tokens. When a company receives such tokens, it can ship them as if they were real products, when this company also incurs a stockout. These products are thus shipped as some products. When these tokens are eventually received by a retailer, they are transformed into a financial penalty for the company that has issued them, when this retailer does not have enough real products to fulfill its demand. On the other hand, these tokens do not incur such a financial penalty when no stockouts occur by the retailer. In consequence, companies only pay for backorders incurred by the retailer because of their own backorder, and not directly for their own backorders.

More generally, making agents take into account their impact on other agents deals with *mechanism design*. Mechanism design is the inverse of game theory, because game theory studies the players' behaviour when these players are given some rules, while mechanism design looks for the rules to give to players, so that these players exhibit the desired behaviour. More precisely, the input of a mechanism is every player's utility function and the function of social choice of these players (we have assumed in this dissertation, that this function of social choice was the sum of all players' utility functions), and the output of this mechanism is the rules to give to players. When players use such rules, they maximize the function of social choice (which is, in our case, the total cost of the supply chain), when they maximize their own utility function. In other words, these rules are designed so that company-players take into account their impact on the rest of the supply chain, when they make their decision, only by maximizing their utility function. We have given some examples of such mechanisms in this dissertation. For example, in Chapter 4, we have said, that some researchers have used them to price the use of shared resources.

In this context, mechanism design could be used to modify each company's benefit function, so that these companies maximize the overall supply chain benefit, when they also maximize their own benefit. Each company's benefit function would take into account a precise description of their expenditure (inventory holding, shippings, raw material purchases, wages...) and revenue (sales), and the company would thus be able make its decision as it is currently used to doing. The only difference would be that these decisions have no negative impacts on the rest of the supply chain.

9.3.2 Why are Agents Reluctant to Share Information?

As future second work, it would be interesting to investigate information sharing in real life. In fact, company-agents in our model have few disadvantages to sharing this information. In Chapter 7, in which all company-agents use the same ordering rule, we have seen that the more information is shared, the lower the overall cost of the supply chain is. But the distribution of this overall cost among agents in the supply chain also changes. This is the reason why we have studied individual incentives for information sharing because some agents may prefer that the whole supply chain have a higher cost, because this reduces their own cost.

Information sharing raises other problems than those related to such change in the distribution of the global cost. Adding these other problems requires extending the model a great deal to take into account:

- competitors, who are able to find this information, and to use it as a competitive advantage against the agents in our supply chain. This point may concern the security of computer systems, in which cryptography plays a central role, by securing the transmission of information on public networks as the Internet;

- negotiation between each agent and its supplier in our supply chain, in which the supplier could have more power in the negotiation if he has more information. That is, negotiating under assymetric information (i.e., agents have different information) may change the output of this negotiation.

In both cases, each agent would still make its decision to share its information or not, based on the decision taken by the other agents. Our methodology that applies some concepts from game theory to analyze simulation outputs, and that was detailed in Section 8.1, could be applied here. Other similar questions can also be studied with this methodology:

- What happens if an agents uses (O, Θ) orders, but lies about O?
- What happens if an agent using (O, Θ) orders does not respect the second principle by emitting Θ anytime, e.g., because it places order (O, Θ) in which Θ is chosen such as $O + \Theta$ is given by a (s, S) ordering policy?

As previoulsy stated with the remark that "knowing when something has changed sufficiently to notify another agent is tricky", another question concerning information sharing in real-life is the interpretation of our second principle "companies should only react once to each market consumption change". In fact, the market consumption is never steady. In this condition, how to say that the market consumption has just changed, and that company-agents are now allowed to emit Θ? Is it necessary that every company-agent use the same threshold to characterize such a change? Do they need to agree on a common threshold? Or does every agent to learn its own threshold with some learning algorithm as Q-learning or a Markov Decision Process (MDP)?

9.3.3 How to Smooth Demand in the Supply Chain?

The third proposed future work is the study of the application of demand smoothing techniques. Conversely to this thesis, other work [Dejonckheere et al., 2003, 2002] has studied such techniques to reduce the bullwhip effect. This research only focuses on how to smooth demand. On the contrary, we could check if companies have incentives to use such smoothing techniques. In fact, we have seen in Section 5.1, that companies in the QWSG amplify the bullwhip effect in order to stabilize their inventory, but they could also order in a steady way if they agreed to let their inventory fluctuate, and thus to have backorders. We can note that inventory level has to be increased when inventory variation also increases.

In this context, is it worth retailers holding excessive inventory to order in a steady way? Or is it better that wholesalers, or another position in the supply chain, do so? In fact, Figure 7.3 shows that the minimum of the overall supply chain cost C varies a great deal depending on the market demand pattern. We recall that these costs can be compared together, because they all represent the same real lumber and wood consumptions for a year. Therefore, if retailers (or another position) had enough inventory to absorb market fluctuations, they could place steady orders. These steady orders would allow minimizing the overall cost of the rest of the supply chain. A part of the money saved by the rest of the supply chain would then be given to retailers in order to pay them for agreeing to this overstocking. Intuitively, it is a win-win situation: the rest of the supply chain saves money, because retailers make an effort for them, and the retailers earn money, because the rest of the supply chain pays them for smoothing demand.

As in the second part of this thesis, each company would have to make its decision to participate or not to this system of paying retailers, and thus, we could study this question with the methodology in Section 8.1. The price of this system is determined by the level of retailers' overstocks. Of course, the companies agreeing to use this system would share this price between themselves, and this system would only be used if at least one company accepts to pay for it. In the case where only one agrees, it would do so by the whole price. In the case where two or more companies agree, the way to share the price has to be studied. Maybe the way to share this price would have an impact on the adaption of this system by each company. Intuitively, all companies would like this system to work, but all companies would also prefer that other companies pay for it. If this situation occurs, Clarke [1971]'s tax could help every company behaves according to the global behaviour of the supply chain. The idea of this tax is to separate the decision of a company and what this company pays, because the company pays for the social consequences of its decision.

Like our two principles to reduce the bullwhip effect, such a study may also be interesting for any distributed system, because the involved principles are very similar and reduce the same cause of stream fluctuations. In particular, we will be able to propose the same kind of translation dictionary as we have proposed to translate our solution to the bullwhip effect to multi-agent systems.

After this system is well designed, we will study its real-life implementation by merging it with some techniques from demand management. In particular, ATP (Available To Promise) [van der Eerden, 2002; Chien-Yu Chen and Ball, 2001] is a system to plan production and procurement. Precisely, instead of planning production only with production units, new units called ATP are used. ATP units are used to plan production in advance, and these units are allocated to customers at the last moment. For example, ATP units are used to plan a production for 30 items of product X for a set of clients 1, and 20 other ATP units of items Y for a set of clients 2. When this allocation is made, buyers of these 30 + 20 products are unknown, but the production capacity is allocated, which allows managing the uncertainty in demand while scheduling production. Next, when some clients 1 buy units of X, ATP is consumed to fulfill this demand, but if some clients 1 also buy units of Y, a new production capacity has to be allocated, because no ATP was planned for this demand. In practice, Gantt diagrams currently have one type of information per type of product that has to be produced. ATP provides new types of information to these diagrams so that to schedule the production for products that are not yet ordered by clients. In conclusion, the smoothing technique proposed in this subsection may plan the allocation of ATP units.

9.3.4 How to Consider a More Realistic Model?

Conversely to the three previous questions, a more practical issue deals with the company and supply chain models implemented in our two simulators. Indeed, we only simulate supply chains according to the QWSG. Two series of questions arise here:

- *Company model:* Answering to the first series of questions requires an adaptation of the company model in the QWSG in order to allow the processing of several types of products and improved company behaviour, i.e., a more realistic or more "intelligent" behaviour. An example of such an adaptation is the multi-item scenario was presented in Section 5.5, but it only illustrated how to use our two proposed ordering schemes, rather than simulating a supply chain using them. We now illustrate the questions that could be studied by such changes in the company model.
    - How to consider companies processing different types of items, instead of only one in the QWSG? In particular, how will companies make a trade-off between their requirements for different types of products? How should companies plan their production? We recall here, that planning is one of the most important applications of artificial intelligence. The problem in the context of multi-agent systems (and supply chains) is to plan with the others, instead of planning alone.

  On the other hand, how does the supply chain behave when (production and inventory) capacities are taken into account?

  - How to implement the method used by companies to choose their ordering scheme? In fact, we have used game theory to show which ordering scheme should be used by companies, by assuming that companies were able to change their scheme, if they could reduce their cost by using another scheme. In this process, we have specified neither how companies see that they should change their ordering scheme, nor how they choose another scheme. This corresponds to meta-rules in our simulators, that is, to the rules that choose which ordering rule to apply. We can note, that such meta-rules are implemented in JACK™.

  - How to improve the forecasting of future demand with some learning methods? In fact, current algorithms for demand forecasting are only based on past demand, but some researchers use learning methods, such as Q-learning and Markov Decision Process (MDP), to take more parameters into account, such as the behaviour of competition, the global state of the economy, the existence of substitution products, and so on. In this approach, the learning algorithm "learns" not only the past behaviour of the demand like in the current approach, but also many elements in the environment that may impact on the demand.

    This approach would be a great extension of Kimbrough et al. [2002] work, who have also applied learning techniques to supply chain management. Since agents learn how to place order to minimize an expected cost, we could compare how agents place orders when they minimize their own cost, and when they minimize the overall supply chain cost.

- *Supply chain model:* The importance of the second series of questions is illustrated by two points: we can notice in Chapter 7, that the difference in the paper and lumber sub-supply chains length can be a problem, and the study in Subsection C.4 of the Sawmill's order aggregation method, shows how difficult this problem is. Some questions about the supply chain model are:

  - How to change for another supply chain model, that is, with more or less companies and with more diverging and converging streams? In fact, we only consider a diverging stream at the output of the Sawmill, which already raises some questions about how to place orders, that are based on these two streams.

  - How to analyze the simulation outputs, when the simulation model has more companies? We have seen, that simulating and analyzing the simulation outputs with Gambit takes around one hour and a half for six companies, where each company has only three possible choices. If we want to consider a more extended supply chain, e.g., five retailers for each wholesaler, and two LumberWholesaler and three PaperWholesaler, the computation time is going to explode, because computing Nash equilibria is not an easy problem. In consequence, designing quicker algorithms to find equilibria would allow simulating and analyzing more realistic supply chains, in which companies could choose between more alternatives. Moreover, such algorithms are interesting for computer science, economics and supply chain management, because game theory is used in these three fields at least.

To answer these two series of questions, many enhancements of the QWSG have been implemented in our second simulator programmed with the JACK™ toolkit. In fact, JACK™ allows providing companies with the concepts from artificial intelligence, e.g., learning and planning, and from multi-agent systems, e.g., distributed simulations. This example of distributed simulation is important, because agents are a natural way of modelling companies. Conversely to mathematical models, such a simulation does not require any abstractions to understand which actions are performed by each company. In consequence, anybody (and thus, company managers) can understand how the whole supply chain behaves when each company has a particular behaviour, and how important the coordination is in a multi-agent system (and a supply chain).

Furthermore, since JACK™ can use any Java code, adding a learning algorithm to agents to forecast demand would be an easy task. Finally, we have previously said, mechanism design could be used in supply chains. Agent-based simulations could graphically show, how such a mechanism works in a supply chain.

In short, we propose to extend our model to make it more realistic. The increase of the realism will surely increase the complexity of both the algorithms used to simulate the supply chain, and the algorithm used by agents to make efficient decisions.

9.3.5 How to Formalize our Interactions?

We now question the possiblity of replacing our two simulators by a mathematical model. In fact, we use the ordering scheme A″ as if the demand is constant during the whole simulation, while parameters s and S should be set with a better method, for example, by enhancing the demonstration in Subsection 8.2.2 to consider several companies, instead of only one. After A″, this formalism would be adaptable to calculate the optimal initial inventories in schemes B and D, instead of using an optimization software on simulations as we did. If this formalisation is possible, simulation could be dropped and game theory could be used directly to study companies incentives to use A'', B or D.

Maybe this formalism would also explain how to set up parameters in Scheme A'. In fact, Figure 7.3 shows that the bullwhip effect is great, but over a very short period. This leads to a bad inventory management in Figure 7.4. If these overorders were controlled, inventory could be managed, and the bullwhip effect minimized without information sharing. The drawback to this scheme is, that it is only based on incoming orders. Therefore, it could be inefficient to manage inventories in practice, because it does not consider inventory level.

Such models already exist in the literature where they are called Joint Economic Lot Size (JELS) or multi-echelon inventory models, as presented in Subsection 4.1.1. We have also presented some applications to supply chain management of traffic flow theory [Daganzo, 2003], and of the use of z-transforms or Laplace transforms from control theory [Dejonckheere et al., 2004, 2003, 2002; Disney and Towill, 2003].

Bibliography

Advanced Micro Devices Inc. (2004). Company web site. http://www.amd.com (accessed 2 May 2004).

Agent Oriented Software Group (2003). JACK Intelligent Agent (TM) User Guide. Technical Report Release 4.1. http://www.agent-software.com/.

Anderson, G. and Morrice, D. J. (2000). A simulation game for service-oriented supply chain management: Does information sharing help managers with service capacity decisions? *The Journal of Production and Operations Management*, 9(1).

Angerhofer, B. J. and Angelides, M. C. (1999). System dynamics modelling in supply chain management: Research review. In Joines, J., Barton, R. R., Kang, K., and Fishwick, P. A., editors, *Proceedings of the 2000 Winter Simulation Conference*.

Apple Computer, Inc. (2004). Mac OS X. http://www.apple.com/macosx/ (accessed 22 April 2004).

Ariba, Inc. (2003). Web site. http://www.ariba.com (accessed 5 September 2003).

Arrow, K. J. (1951). *Social Choice and Individual Values*. New York: John Wiley.

Asselin, F. (2002). Formation de coalitions d'agents logiciels pour le groupement d'acheteurs. Master's thesis, Université Laval, Département d'Informatique et de Génie Logiciel, Quebec City (Quebec, Canada).

Asselin, F. and Chaib-draa, B. (2002). Towards a protocol for the formation of coalitions of buyers. In Crainic, T. G. and Gavish, B., editors, *Proc. Fifth International Conference on Electronic Commerce Research (ICECR-5)*.

Baan AG (2003). Web site. http://www.baan.com (accessed 5 September 2003).

Banerjee, A. (1986). A joint economic lot size model for purchaser and vendor. *Decision Science*, 17:292–311.

Barbuceanu, M. and Fox, M. (1995a). The architecture of an agent building shell. In *Intelligent Agents II, Springer-Verlag, LNAI 1037*, pages 235–250.

Barbuceanu, M. and Fox, M. (1995b). Cool: A language for describing coordination in multi-agent systems. In *Proc. First International Conference on Multi-Agent Systems*.

Baumgaertel, H., Brueckner, S., Parunak, V., Vanderbok, R., and Wilke, J. (2001). Agent models of supply network dynamics. *The Practice of Supply Chain Management*. URL http://www.anteaters.net/ sbrueckner/publications/2001/AgentModels-OfSupplyNetworkDynamics_submitted_20010209.pdf.

Beck, J. and Fox, M. (1994). Supply chain coordination via mediated constraint relaxation. In *Proc. 1st Canadian Workshop on Distributed Artificial Intelligence*, pages 61–72, Banff (Alberta, Canada).

Ben-Ameur, H., Chaib-draa, B., Gérin-Lajoie, R., Kropf, P. G., and Vaucher, S. (2002a). Towards an agent-based package e-procurement. Série scientifique du CIRANO de Montréal (Quebec, Canada).

Ben-Ameur, H., Chaib-draa, B., and Kropf, P. (2002b). Multi-items auctions for automatic negotiation. *Journal of Information and Software Technology*, 44(5):291–301.

Binmore, K. (1991). *Fun and Games - A Text on Game Theory*. D. C. Heath & Co.

Bond, A. H. and Gasser, L. (1988). An analysis of problems and research in DAI. In Bond, A. H. and Gasser, L., editors, *Readings in Distributed Artificial Intelligence*, chapter 1.

Bournez, C. and Gutknecht, O. (2001). Modèle de contrôle par émergence de coordinations dans un réseau de contrats multiagents. In *Journées Francophones de l'Intelligence Artificielle Distribuée et des Systèmes MultiAgents*, Montreal (Quebec, Canada).

Boutilier, C. (1996). Planning, learning and coordination in multiagent decision processes. In *Theoretical Aspects of Rationality and Knowledge*, pages 195–201.

Boutilier, C. (1999). Sequential optimality and coordination in multiagent systems. In *Proc. 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 478–485.

Boutilier, C., Shoham, Y., and Wellman, M. P. (1994). Economic principles on multiagent systems (editorial). *Artifical Intelligence*, 94(1-2).

Breban, S. and Vassileva, J. (2002). A coalition formation mechanism based on interagent trust relationships. In *Proc. 1st int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 306–307, Bologna, Italy.

British Telecommunications plc (2004). The ZEUS agent building toolkit. `http://more.btexact.com/projects/agents/zeus/` (accessed 21 May 2004).

Burns, J. F. and Sivazlian, B. D. (1978). Dynamic analysis of multi-echelon supply systems. *Computers & Industrial Engineering*, 2(4).

Buzacott, J. and Shanthikumar, G. (1992). A general approach for coordinating production in multiple-cell manufacturing systems. *Production and Operations Management*, 1(1):34–52.

Cachon, G. P. (1999). Managing supply chain demand variability with scheduled ordering policies. *Management Science*, 45(6):843–856.

Cachon, G. P. (2004). Supply chain coordination with contracts. This working paper is an invited chapter to be published in the "*Handbooks in Operations Research and Management Science: Supply Chain Management*", edited by Steve Graves and Ton de Kok and published by North-Holland. `http://opim.wharton.upenn.edu/~cachon/pdf/cachon_chp_scontract.pdf` (accessed Nov. 11, 2004).

Cachon, G. P. and Lariviere, M. A. (1999). Supply chain inventory management and the value of shared information. *Management Science*, 46(8):1032–1048.

Cachon, G. P. and Netessine, S. (2003). Game theory in supply chain analysis. In Simchi-Levi, D., Wu, S. D., and Shen, Z.-J. M., editors, *Supply Chain Analysis in the eBusiness Era*. Kluwer (to appear).

Cachon, G. P. and Zipkin, P. H. (1999). Competitive and cooperative inventory policies in a two-stage supply chain. *Management Science*, 45(7):936–953.

Callantine, T. J. (2003). Air traffic controller agents. In *Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 952–953, Melbourne (Victoria, Australia).

Cammarata, S., McArthur, D., and Steeb, R. (1983). Strategies of cooperation in distributed problem solving. In *Proc. 8th International Joint Conference on Artificial Intelligence (IJCAI-83)*, pages 767–770.

Carlsson, C. and Fullér, R. (2000). A fuzzy approach to the bullwhip effect. In *Proc. 15th European Meeting on Cybernetics and Systems Research*. Vienna (Austria).

Carlsson, C. and Fullér, R. (2001). Reducing the bullwhip effect by means of intelligent, soft computing methods. In *Proc. 34th Hawaii International Conference on System Sciences (HICSS-34)*, (Hawaii, USA).

Chaib-draa, B. (1995). Industrial applications of distributed ai. *Communication of ACM*, 38(11):49–53.

Chaib-draa, B., Moulin, B., and Jarras, I. (2001). Agent et systèmes multiagents. In J.P. Briot et Y Demazeau, editor, *Principes et Architecture des Systèmes Multi-Agents*. Hermes, Lavoisier.

Chaib-draa, B., Moulin, B., Mandiau, R., and Millot, P. (1992). Trends in distributed artificial intelligence. *Artificial Intelligence Review*, 6:35–66.

Chalkiadakis, G. and Boutilier, C. (2003). Coordination in multiagent reinforcement learning: A bayesian approach. In *Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, Melbourne (Victoria, Australia).

Chandra, P. and Fisher, M. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72.

Chatfield, D. C. (2001). *SISCO and SCML: Software tools for supply chain simulation modelling and information sharing.* Ph. D thesis, Pennsylvania State University (Pennsylvania, USA).

Chen, F., Drezner, Z., Ryan, J. K., and Simchi-Levi, D. (1994). The bullwhip effect: Managerial insights on the impact of forecasting and information on variability in a supply chain. In *Quantitative Models for Supply Chain Management*, chapter 14. Sridhar Tayur, Ram Ganeshan and Michael Magazine.

Chen, F., Drezner, Z., Ryan, J. K., and Simchi-Levi, D. (2000). Quantifying the bullwhip effect in a simple supply chain: The impact of forecasting, lead times, and information. *Management Science*, 46(3):436–443.

Chen, F. and Samroengraja, R. (2000). The stationary Beer Game. *Production and Operations Management*, 9(1):19–30.

Chen, Y., Peng, Y., Finin, T., Labrou, Y., and Cost, S. R. (1999). Negotiating agent for supply chain management.

Cheng, S.-F., Leung, E., Lochner, K. M., O'Maley, K., Reeves, D. M., Schvartzman, L. J., and Wellman, M. P. (2003). Walverine: A walrasian trading agent. In *Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 465–472, Melbourne (Victoria, Australia).

Chien, C.-C. and Ioannou, P. A. (1992). Automatic vehicle-following. In *Proc. 1992 American Control Conference*, pages 1748–1752, Chicago (Illinois, USA).

Chien-Yu Chen, Z.-Y. Z. and Ball, M. O. (2001). Quantity and due date quoting available to promise. *Information Systems Frontiers*, 3(4):477–488.

Cingil, I. and Dogac, A. (2001). An architecture for supply chain integration and automation on the internet. *Journal of Distributed and Parallel Databases*, 10(10):59–102.

Clark, A. J. and Scarf, H. E. (1960). Optimal policies for a multi-echelon inventory problem. *Management science*, 6:465–490.

Clarke, E. H. (1971). Multipart pricing of public goods. *Public Choice*, 18:19–33.

Cloutier, L., Frayret, J.-M., D'Amours, S., Espinasse, B., and Montreuil, B. (2001). A commitment-oriented framework for networked manufacturing coordination. *International Journal of Computer Integrated Manufacturing*, 14(6):522–534.

Commerce One Operations, Inc. (2003). Web site. `http://www.commerceone.com/` (accessed 19 September 2003).

Conitzer, V. and Sandholm, T. (2003). Complexity results about nash equilibria. In *Proc. 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 765–771.

Conseil de l'Industrie Forestière du Québec (2004). Web site. `http://www.cifq.qc.ca/` (accessed 16 February 2004).

Cooper, M. C., Lambert, D. M., and Pagh, J. D. (1997). Supply chain management: More than a new name for logistics. *International Journal of Logistics Management*, 8(1).

Corbett, C. J. and de Groote, X. (2000). A supplier's optimal quantity discount policy under asymmetric information. *Management Science*, 46(3):444–450.

Corbett, C. J., Zhou, D., and Tang, C. S. (2004). Designing supply contracts: Contract type and information asymmetry. *Management Science*, 50(4):550–559.

Croson, R. and Donohue, K. (2002). Experimental economics and supply-chain management. *Interfaces*, 32(5):74–82.

Crowther, J. F. (1964). Rationale for quantity discounts. *Harvard Business Review*, 42:121–127.

Daganzo, C. (2003). *A theory of supply chains*. Springer.

DAMAS (2004). Web site, research group at Université Laval, Quebec City (Quebec, Canada). `http://damas.ift.ulaval.ca` (accessed 29 July 2004).

D'Amours, S., Montreuil, B., Lefrançois, P., and Soumis, F. (1999). Networked manufacturing : The impact of information sharing. *International Journal of Production Economics*, 58:63–79.

DARPA (2003). DAML (DARPA Agent Markup Language). `http://www.daml.org/` (accessed 19 September 2003).

Davidsson, P. and Werstedt, F. (2002). A multi-agent system architecture for coordination of just-in-time production and distribution. *Knowledge Engineering Review*, 17(4):317–329.

Dejonckheere, J., Disney, S. M., Lambrecht, M. R., and Towill, D. R. (2002). Transfer function analysis of forecasting induced bullwhip in supply chains. *International Journal of Production Economics*, 78:133–144.

Dejonckheere, J., Disney, S. M., Lambrecht, M. R., and Towill, D. R. (2003). Measuring and avoiding the bullwhip effect: A control theoretic approach. *European Journal of Operational Research*, 147:567–590.

Dejonckheere, J., Disney, S. M., Lambrecht, M. R., and Towill, D. R. (2004). The impact of information enrichment on the bullwhip effect in supply chains: A control engineering perspective. *European Journal of Operational Research*, 153:727–750.

Delgado, J. (2002). Emergence of social conventions in complex networks. *Artificial Intelligence*, 141(1):171–185.

Disney, S. M. and Towill, D. R. (2003). On the bullwhip and inventory variance produced by an ordering policy. *The International Journal of Management Science*, 31:157–167.

Dodd, C. and Kumara, S. R. T. (2001). A distributed multi-agent model for value nets. *IEA/AIE 2001*, pages 718–727.

Dolan, R. J. (1987). Quantity discount: Managerial issues and research opportunities. *Marketing Science*, 6(1).

Durfee, E. H. (2001). Scaling up agent coordination strategies. *IEEE Computer*, 34(7).

Durfee, E. H. and Lesser, V. H. (2001). Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5). Special Issue on Distributed Sensor Networks.

Durfee, E. H. and Lesser, V. R. (1989). Negotiating task decomposition and allocation using partial global planning. In Gasser, L. and Huhns, M., editors, *Distributed Artificial Intelligence*, volume 2, pages 229–244. Morgan Kaufmann, San Mateo (California, USA).

eBay, Inc. (2004). Web site. `http://www.ebay.com/` (accessed 16 May 2004).

Eymann, T. (2001). Markets without makers - A framework for decentralized economic coordination in multiagent systems. In *Lecture Notes in Computer Science 2232*, pages 63–74. Proc. 2nd International Workshop WELCOM.

Fabrikant, A., Papadimitriou, C. H., and Talwar, K. (2004). The complexity of pure nash equilibria. In *Proc. STOC 2004 (extended abstract)*.

Fan, M., Stallaert, J., and Whinston, A. B. (2003). Decentralized mechanism design for supply chain organizations using an auction market. *Information Systems Research*, 14(1):1–22.

Ferber, J. (1995). *Les systemes multi-agents, vers une intelligence collective*. InterEditions, Paris (France).

Fjeld, D. E. (2001). The wood supply game as an educational application for simulating dynamics in the forest sector. In Sjostrom, K. and Rask, L.-O., editors, *Supply Chain Management for Paper and Timber Industries*, pages 241–251, Växjö (Sweden).

FOR@C (2004). Web site of the Research Consortium in e-business in the forest products industry at Université Laval, Quebec City (Quebec, Canada). http://www.forac.ulaval.ca (accessed 15 September 2004).

Forrester, J. W. (1958). Industrial dynamics - A major breakthrough for decision-makers. *Harvard Business Review*, 36(4):37–66.

Forrester, J. W. (1961). *Industrial Dynamics*. The M.I.T. Press - Massachusetts Institute of Technology - Cambridge (Massachusetts, USA).

Fox, M. S., Chionglo, J. F., and Barbuceanu, M. (1993). The integrated supply chain management. Internal report of the Enterprise Integration Laboratory, Department of Industrial Engineering, University of Toronto (Ontario, Canada.

Franchesquin, N. and Espinasse, B. (2000). Analyse multiagent de la gestion hydraulique de la camargue: Considérations méthodologieques. In *Proc. Journées Francophones des SMA et de l'IAD*, pages 325–342, St Etienne (France).

Fransoo, J. C. and Wouters, M. J. (2000). Measuring the bullwhip effect in the supply chain. *Supply Chain Management*, 5(2).

Frayret, J.-M. (2002). *A conceptual framework to operate collaborative manufacturing networks*. Ph. D. thesis, Université Laval, Quebec City (Quebec, Canada).

Free Software Foundation (2004a). Web site of GCC (GNU compiler collection). http://gcc.gnu.org/ (accessed 14 April 2004).

Free Software Foundation (2004b). Web site of the gnu general public license. `http://www.gnu.org/copyleft/gpl.html` (accessed 18 february 2004).

Garé, B. (2000). Entreprise, ouvre-toi ! *Logiciels & systèmes*, 52. Dossier spécial chaîne logistique.

Gasser, L. and Briot, J.-P. (1992). Object-based concurrent programming and distributed artificial intelligence. In Avouris, N. and Gasser, L., editors, *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer Academic Publishers.

Geoffrion, A. M. and Krishnan, R. (2003). E-business and management science - Mutual impacts (part 2 of 2). *Management Science*, 49(11):1445–1456.

Gottlob, G., Greco, G., and Scarcello, F. (2003). Pure nash equilibria: Hard and easy games. In *Proc. 9th conference on Theoretical Aspects of Rationality and Knowledge, (TARK'03)*, pages 215–230.

Grand, S. and Cliff, D. (1998). Creatures: Entertainment software agents with artificial life. *Autonomous Agents and Multi-Agent Systems*, 1(1).

Gruver, W. A., Kotak, D. B., van Leeuwen, E. H., and Norrie, D. H. (2003). Holonic manufacturing systems: Phase 2. In *Holonic and Multi-Agent Systems for Manufacturing - 1st International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2003*, pages 1–14, Prague (Czech Republic) `http://gerstner.felk.cvut.cz/HoloMAS/2003/`.

Guerrien, B. (1995). *La Théorie des Jeux - Second Édition*. Economica.

Guessoum, Z. and Briot, J.-P. (1999). From active objects to autonomous agents. *IEEE Concurrency*, 7(3).

Haartveit, E. Y. and Fjeld, D. E. (2002). Experimenting with industrial dynamics in the forest sector - A Beer Game application. In *Symposium on Systems and Models in Forestry*, Punta de Tralca (Chile).

Hallé, S., Chaib-draa, B., and Laumonier, J. (2003). Car platoons simulated as a multiagent system. In *Proc. 4th Workshop on Agent-Based Simulation*, pages 57–63, Montpellier (France). SCS.

Hax, A. and Candea, D. (1984). *Production and Inventory Management*. Prentice-Hall, New York (New York, USA).

Hayes-Roth, B., Hewitt, M., Washington, R., and Seiver, A. (1989). Distributing intelligence with an individual. In Gasser, L. and Huhns, M., editors, *Distributed Artificial Intelligence, Vol. II*, pages 385–412.

Hillier, F. S. and Lieberman, G. J. (1997). *Introduction to Operations Research*. Sixth Edition, McGraw Hill, New York (New York, USA).

Hines, P. and Holweg, M. (2000). Waves, beaches, breakwaters and rip currents - A three-dimensional view of supply chain dynamics. *International Journal of Physical Distribution and Logistics Management*, 31(10):827–846.

Hiroaki, K. (2000). Robocup rescue: A grand challenge for multi-agent systems. In *Proc. International Conference on Multi-Agent Systems (ICMAS 2000)*.

Huhns, M. N. and Stephens, L. M. (1999). Multi-agent systems and societies of agents. In Weiss, G., editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, pages 79–120. MIT Press, Cambridge (Massachusetts, USA).

Iaccoca Institute (1991). 21st Century Enterprise Strategy: An Industry Led View. Lehig University, Bethlehem (Pennsylvania, USA).

IBM Research (2003). BPMAT: A business process modeling and analysis tool. `http://www.research.ibm.com/pdtr/bpmat.htm` (accessed 19 September 2003).

Ioannou, P. A. and Chien, C.-C. (1993). Autonomous intelligent cruise control. *IEEE Trans. on Vehicular Technology*, 42(4):657–672.

Jacobs, F. R. (2000). Playing the beer distribution game over the internet. *Production and Operations Management*, 9(1):31–39.

Jehle, G. A. and Reny, P. J. (2000). *Advanced Microeconomic Theory*. Addison Wesley. 2d Edition.

Jennings, N. (1993a). Specification and implementation of a Belief Desire Joint-Intention architecture for collaborative problem solving. *Int. Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318.

Jennings, N. (2000). On agent-based software engineering. *Artificial Intelligence*, 117(2).

Jennings, N., Sycara, K., and Wooldridge, M. (1998). A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38.

Jennings, N. and Wooldridge, M. (1998). Applications of intelligent agents. In *Agent technology: Foundations, applications and markets*, pages 3–28. Springer-Verlag New York, Inc.

Jennings, N. R. (1993b). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250.

John Taras CPIM (2003). VMI Web site. `http://www.vendormanagedinventory.com/` (accessed 8 September 2003).

Johnson, L. A. and Montgomery, D. C. (1974). *Operations Research in Production Planning, Scheduling, and Inventory Control*. John Wiley & Sons, Inc., New York, London, Sydney, Toronto.

Kahn, J. A. (1987). Inventories and the volatility of production. *The American Economic Review*, 77(4):667–679.

Kelle, P. and Milne, A. (1999). The effect of (s, S) ordering policy on the supply chain. *International Journal of Production Economics*, 59:113–122.

Khan, A. (2004). Java Excel API 2.3.12. A Java API to read, write and modify Excel spreadsheets `http://www.andykhan.com/` (accessed 2 March 2003).

Kimbrough, S. O., Wu, D., and Zhong, F. (2002). Computers play the Beer Game: Can artificial agents manage supply chains? *Decision Support Systems*, 33(3):323–333.

Kosadat, A. and Liman, S. D. (2002). Joint economic lot-size model with backordering policy. *International Journal of Production Research*. (in submission).

Kotler, P. (2001). *Principles of Marketing*. Prentice Hall Europe. Essex (UK).

Kruger, I. (2004). Company web site. `http://www.kruger.com` (accessed 3 May 2004).

Labarthe, O. (2000). Modélisation multi-agent de chaînes logistiques. Mémoire de Diplôme d'Études Approfondies, Université d'Aix-Marseille 3, Marseilles (France).

Lasdon, L. S., Waren, A. D., Jain, A., and Ratner, M. (1978). Design and testing of a generalized reduced gradient code for nonlinear programming. *ACM Transactions on Mathematical Software*, 4(1):34–50.

Lee, H. L. (2001). E-business and supply chain integration. Stanford Global Supply Chain Forum `http://www.stanford.edu/group/scforum/Welcome/EB_SCI.pdf`.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997a). The bullwhip effect in supply chain. *Sloan Management Review*, 38(3):93–102.

Lee, H. L., Padmanabhan, V., and Whang, S. (1997b). Information distortion in a supply chain: The bullwhip effect. *Management Science*, 43(4):546–558.

Lee, H. L. and Whang, S. (1998). Information sharing in a supply chain. Research paper No 1549, Graduate school of business, Stanford university.

Lee, H. L. and Whang, S. (1999). Decentralized multi-echelon supply chains: incentives and information. *Management Science*, 45(5):633–640.

Liberopoulous, G. and Dallery, Y. (2000). A unified framework for pull control mechanisms in multi-stage manufacturing systems. *Annals of Operations Research*, 93(1):325–355.

Lipton, R. J. and Markakis, E. (2004). Nash equilibria via polynomial equations. In *Proc. LATIN 2004: Theoretical Informatics, 6th Latin American Symposium*, pages 413–422.

Lizotte, S. and Chaib-draa, B. (1997). Coordination in CE systems: An approach based on the management of dependencies between agents. *Concurrent Engineering: Research and Applications (CERA)*, 5(4):367–377.

Ljungberg, M. and Lucas, A. (1992). The oasis air traffic management system. In *Proc. 2nd Pacific Rim International Conference on Artificial Intelligence (PRICAI '92).*

Lyonnais, P. and Montreuil, B. (2001). Un environnement distribué de simulation à base d'agents. In *3e conférence francophone de MOdélisation et SIMulation - conception, analyse et gestion des systm̀es industriels*, Troyes (France).

MacHo-Stadler, I., Perez-Castrillo, D. J., and Watt, R. (2001). *An Introduction to the Economics of Information: Incentives and Contracts*. Oxford Press, 2nd edition.

MacKie-Mason, J. K. and Varian, H. R. (1995). Pricing congestible network resources. *IEEE Journal on Selected Areas in Communications*, 13(7).

Mallya, S. N. (1999). *Demand Diven Production and Distribution: Seamless Integration of Downstream Supply Chain Decisions using Electronic Data Interchange*. Ph. D. thesis, Drexel University (Philadelphia, PA, USA).

Malone, T. W. and Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing surveys*, 26(1):87–119.

Maturana, F., Shen, W., and Norrie, D. (1999). Metamorph: an adaptive agent-based architecture for intelligent manufacturing. *International Journal of Production Research*, 37(10):2159–2173.

McKelvey, R. D. and McLennan, A. (1996). Computation of equilibria in finite games. In Amman, H. M., Kendrick, D. A., and Rust, J., editors, *Handbook of Computational Economics*, pages 87–142. Elsevier.

McKelvey, R. D., McLennan, A. M., and Turocy, T. L. (2004). Gambit: Software tools for game theory, version 0.97.0.4. `http://econweb.tamu.edu/gambit/` (accessed 16 October 2003).

Microsoft Corp. (2004a). Pack MS-Office 2000. `http://www.microsoft.com/office/` (accessed 29 January 2004).

Microsoft Corp. (2004b). Web page “XL2000: Solver uses generalized reduced gradient algorithm from Leon Lasdon, of the University of Texas at Austin, and Allan Waren, of Cleveland State University”. "http://support.microsoft.com:80/support/-kb/articles/Q214/1/15.ASP&NoWebContent=1 (accessed 29 January 2004).

Microsoft Corp. (2004c). Windows operating system. `http://www.microsoft.com/windows/` (accessed 22 April 2004).

Min, J. U. and Bjornsson, H. (2000). Agent-based supply chain management automation. In *Proc. 8th International Conference on Computing in Civil and Building Engineering (ICCCBE-VIII)*.

Mintzberg, H. (1978). *The Structuring of Organizations: A Synthesis of the Research*. Englewood Cliffs, N.J.: Prentice-Hall.

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2003a). Coordination à base de jetons pour réduire l’amplification de la variabilité de la demande dans une chaîne logistique. In *Proc. 5th International Industrial Engineering Conference*, The Loews Le Concorde, Quebec City (Quebec, Canada).

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2003b). Multi-agent coordination based on tokens: Reduction of the bullwhip effect in a forest supply chain. In *Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 670–677, Melbourne (Victoria, Australia).

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2004a). An agent simulation model for the québec forest supply chain. In *Proc. 8th International Workshop on Cooperative Information Agents (CIA2004)*, volume 3191 of *Lecture Notes in Computer Science*, pages 226–241, Erfurt (Germany).

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2004b). Experimental study of incentives for collaboration in the quebec wood supply game. *Management Science*. (submitted).

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2004c). The impact of information sharing on the efficiency of an ordering approach in reducing the bullwhip effect. *IEEE Trans. on Systems, Man, and Cybernetics*. (submitted).

Moyaux, T., Chaib-draa, B., and D’Amours, S. (2004d). Multi-agent simulation of collaborative strategies in a supply chain. In *Proc. 3rd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, New York (New York, USA).

Muckstadt, J., Murray, D., Rappold, J., and Collins, D. (2001). Guidelines for collaborative supply chain system design and operation. *Information Systems Frontiers*, 3(4):427–435.

Nahmias, S. (1997). *Production and Operations Analysis.* The McGraw-Hill Companies, Inc., 3^{rd} edition.

Naish, H. F. (1994). Production smoothing in the linear quadratic inventory model. *The Economic Journal*, 104(425):864–875.

Nash, J. F. (1951). Non-cooperative games. *Annals of Mathematics*, 54:286–295.

National Science Foundation (2004). Program announcement 98-168 for advanced computational research. URL http://www.nsf.gov/pubs/1999/nsf98168/nsf98168.htm (accessed 16 March 2004).

Nwana, H., Lyndon, L., and Jennings, N. (1996). Co-ordination in software agent systems. *The British Telecom Technical Journal*, 14(4):79–88.

OASIS (2003). The XML cover pages - XML Common Business Library (xCBL). `www.oasis-open.org/cover/cbl.html` (accessed 19 September 2003).

Papadimitriou, C. H. (2001). Algorithms, games, and the internet. In *Proc. STOC 2001 (extended abstract in ICALP 2001).*

Papadimitriou, C. H. and Roughgarden, T. (2004). Computing equilibria in multiplayer games. Algorithmic results for the special case of symmetric and other succinctly representable games: correlated equilibria can be found in polynomial time for a broad class that includes all previously known cases.

Paquet, S. (2001). Coordination de plans d'agents: Application à la gestion des ressources d'une frégate. Master's thesis, Université Laval, Quebec City (Quebec, Canada).

Paquet, S. (2003). Learning Coordination in RoboCupRescue. In Xiang, Y. and Chaib-draa, B., editors, *Lecture Notes in Artificial Intelligence*, volume 2671, pages 627–628. Springer-Verlag. Proc. 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2003).

Parunak, H. V. D. (1996). Applications of distributed artificial intelligence in industry. In O'Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed Artificial Intelligence*, pages 71–76. John Wiley & Sons.

Parunak, H. V. D. (1999). Industrial and practical applications of DAI. In Weiss, G., editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, pages 377–424. MIT Press, Cambridge (Massachusetts, USA).

Parunak, H. V. D. and VanderBok, R. (1998). Modeling the extended supply network. In *ISA-Tech'98 (Houston)*, Industrial Technology Institute.

PeopleSoft, Inc and J. D. Edwards & Co. (2003). Web site. http://www.peoplesoft.com and http://www.jdedwards.com (accessed 19 September 2003).

Plamondon, P. (2003). A frigate survival approach based on real-time multiagent planning. Master's thesis, Université Laval, Quebec City (Quebec, Canada).

Plamondon, P., Chaib-draa, B., Beaumont, P., and Blodgett, D. (2003). A frigate movement survival agent-based approach. In *Lecture Notes in Artificial Intelligence*, 7th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2003), University of Oxford (UK). Springer.

Poirier, C. C. and Reiter, S. E. (2001). *La Supply Chain - Optimiser la Chaîne Logistique et le Réseau Interentreprises.* Dunod (Les Réalités de l'entreprise). Translation of "The Supply Chain Optimization: Building the Strongest Total Business Network" (1996) by Monique Sperry.

Porteus, E. L. (2000). Responsibility tokens in supply chain management. *Manufacturing & service operations, Informs*, 2(2):203–279.

Prasad, M. N. and Lesser, V. R. (1999). Learning situation-specific coordination in cooperative multi-agent systems. In *Proc. 2nd Autonomous Agents and Multi-Agent Systems (AAMAS).*

Procter & Gamble Company (2004). Web site. http://www.pg.com/ (accessed 5 April 2004).

Rasmusen, E. (1994). *Game and Information - An Introduction to Game Theory.* Blackwell, Cambridge MA & Oxford UK.

Rigby, C., Day, M., Forrester, P., and Burnett, J. (2000). Agile supply: Rethinking systems thinking, systems practice. *International Journal of Agile Management Systems*, 2(3):178–186.

Rosenschein, J. S. and Zlotkin, G. (1994). *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers.* MIT Press, Cambridge (Massachusetts, USA).

Rubin, P. A. and Benton, W. C. (2003). A generalized framework for quantity discount pricing schedules. *Decision Sciences*, 34(1):173.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence - A modern approch.* Englewood Cliffs, NJ: Prentice-Hall (2d Edition).

Ryan, J. K. (1997). *Analysis of inventory models with limited demand information.* Ph. D. thesis, Northwestern University, Evanston (Illinois, USA).

Sadeh, N., Hildum, D., Kjenstad, D., and Tseng, A. (1999). MASCOT: An agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling. In *Workshop notes, Agent-Based Decision Support for Managing the Internet-Enabled Supply Chain, 3rd International Conference on Autonomous Agents (Agents' 99)*.

Sandholm, T. W. (1999). Distributed rational decision making. In Weiss, G., editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. MIT Press, Cambridge (Massachusetts, USA).

Savelsbergh, M. W. P. (1985). Local search in routing problems with time windows. *Annuals of Operations Research*, 4(6):265–305.

Scholl, H. J. (2001). Looking across the fence: Comparing findings from SD modeling efforts with those of other modeling techniques. In *19th International Conference of the System Dynamics Society*, Atlanta (Georgia, USA).

Search for Extra-Terrestrial Intelligence (2004). Web site of SETI@home. `http://setiathome.ssl.berkeley.edu/` (accessed 17 May 2004).

Shapiro, J. F. (2000). *Modeling the Supply Chain*. Duxbury.

Sheikholeslam, S. and Desoer, C. A. (1990). Longitudinal control of a platoon of vehicles. In *Proc. 1990 ACC*.

Shen, W. and Norrie, D. H. (1999a). Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems*, 1(2):129–156.

Shen, W. and Norrie, D. H. (1999b). Implementing internet enabled virtual enterprises using collaborative agents. In *Infrastructures for Virtual Enterprises*, pages 343–352. Camarinha-Matos, L.M.

Shen, W., Norrie, D. H., and Barthès, J.-P. A. (2001). *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. Taylor & Francis (London and New York).

Shen, Y. (2001). Impact of asymmetric information on inventory policy. Term paper to Dr. Miller for the partial fulfillment of "Production Control".

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60:51–92.

Shubik, M. (1960). Games decisions and industrial organization. *Management Science*, 6(4).

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E. (2000). *Designing and managing the supply chain*. McGraw-Hill Higher Education.

Sjöström, K. (2001). Costs and lead time of material flows in pulp and paper manufacturing line. In Sjöström, K. and Rask, L.-O., editors, *Supply Chain Management for Paper and Timber Industries*, pages 192–202, Växjö (Sweden).

Smith, R. G. (1980). The contract net protocol: High level communication and control in distributed problem solver. *IEEE Trans. on Computers*, C-29(12):1104–1113.

Soucy, M. (2004). Gestion des ressources en temps réel : Cas des frégates canadiennes. Master's thesis, Université Laval, Quebec (Quebec, Canada).

Stadtler, H. (2000). Supply chain management - an overview. In Stadtler, H. and Kilger, C., editors, *Supply Chain Management and Advanced Planning - Concepts, Models, Software and Case Studies*, pages 7–28. Springer-Verlag.

Sterman, J. D. (1989). Modeling managerial behavior: Misperceptions of feedback in a dynamic decision making experiment. *Management Science*, 35(3):321–339.

Sugawara, T. and Lesser, V. R. (1998). Learning to Improve Coordinated Actions in Cooperative Distributed Problem-Solving Environments. *Machine Learning*.

Sun microsystems (2003). Java 2 platform. `http://java.sun.com/` (accessed 19 September 2003).

Supply Chain Council (2001). SCOR Version 5.0! introduction web cast. http://www.-supplychainworld.org/WebCast/SCOR50_overview.ppt.

Swaminathan, J., Sadeh-Koniecpol, N., and Smith, S. (1997). Effects of sharing supplier capacity information.

Swaminathan, J. M., Smith, S. F., and Sadeh, N. M. (1994). Modeling the dynamics of supply chains. In *Proc. AAAI-94 Workshop on reasoning about the shop floor*, Seattle (Washington, USA).

Swaminathan, J. M., Smith, S. F., and Sadeh, N. M. (1998). Modeling supply chain dynamics: A multiagent approach. *Decision Sciences*, 29(3):607–632.

Taylor, D. (1999). Measurement and analysis of demand amplification across the supply chain. *The International Journal of Logistics Management*, 10(2):55–70.

Taylor, D. (2000). Demand amplification: Has it got us? *International Journal of Physical Distribution and Logistics Management*, 30(6):515–533.

Taylor, D. (2001). An approach to the identification and elimination of demand amplification across the supply chain. In Taylor, D. and Brunt, D., editors, *Manufacturing Operations and Supply Chain Management*, pages 155–174. Thomson Learning.

Teigen, R. and Barbuceanu, M. (1996). The supply chain demonstrator, Rapport technique et guide de l'utilisateur.

Trading Agent Competition (2004). Web site. `http://www.sics.se/tac/` (accessed 16 May 2004).

Tsay, A. A., Nahmias, S., and Agrawal, N. (1999). Modelling supply chain contracts: A review. In et al., T., editor, *Quantitative Models for Supply Chain Management*, pages 299–336. Kluwer Academic Publishers, Dordrecht.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.

Umbhauer, G. (2002). *Théorie des Jeux Appliquée à la Gestion*. EMS Management & Société, France.

United Devices Grid (2004). Company web site. `http://www.grid.org/` (accessed 17 May 2004).

van der Eerden, H. (2002). Maximizing service and profit in customer order promising. In *Proc. Fusion: Knowledge and People (APICS 2002)*.

Varian, H. R. (1995). Economic mechanism design for computerized agents. In Usenix, e., editor, *Proc. Usenix Conference on Electronic Commerce (New York, NY, USA)*.

Vetsikas, I. A. and Selman, B. (2003). A principled study of the design tradeoffs for autonomous trading agents. In *Proc. 2nd int. joint conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 473–480, Melbourne (Victoria, Australia).

VICS Association (2003). Web site. `http://www.cpfr.org` (accessed 10 September 2003).

Viswanathan, S. and Piplani, R. (2001). Coordinating supply chain inventories through common replenishment epochs. *European Journal of Operational Research*, 129:277–286.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton, N.J.: Princeton University Press, 1980, c1972.

Wal-Mart Stores, Inc. (2004). Web site. `http://www.walmart.com/` (accessed 5 April 2004).

Waller, M., Johnson, M. E., and Davis, T. (1999). Vendor-managed inventory in the retail supply-chain. *Journal of business logistics*, 20(1):183–203.

Walsh, W. E. (2001). *Market protocols for decentralized supply chain formation*. Ph. D. thesis, University of Michigan, Ann Arbor (Michigan, USA).

Walsh, W. E. and Wellman, M. P. (1998). A market protocol for decentralized task allocation. In *Proc. 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, Department of computer science and information systems, University of Hong Kong.

Walsh, W. E. and Wellman, M. P. (1999). Modeling supply chain formation in multiagent systems. In *Agent mediated electronic commerce (IJCAI workshop)*, pages 94–101.

Weiss, G. (1999). *Multiagent systems. A modern approach to distributed artificial intelligence*. The MIT Press.

Wilding, R. (1998). The supply chain complexity triangle - uncertainty generation in the supply chain. *International Journal of Physical Distribution & Logistics Management*, 28(8):599–616.

Wooldridge, M. (1999). Intelligent agents. In Weiss, G., editor, *Multiagent Systems - A Modern Approach to Distributed Artificial Intelligence*, pages 28–77. MIT Press, Cambridge (Massachusetts, USA).

Wooldridge, M. (2001). *An Introduction to Multi-Agent Systems*. John Wiley & Sons, Inc. New York, NY, USA.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152.

World Wide Web Consortium (2003a). Resource Description Framework 1.0. `www.w3.org/TR/rdf-schema/` & `www.w3.org/TR/REC-rdf-syntax` (accessed 10 September 2003).

World Wide Web Consortium (2003b). Web site. `http://www.w3c.org` (accessed 10 September 2003).

Yan, L. (2001). Impact of lead-time distribution on the bullwhip effect and supply chain performance. In *Proc. Americas Conference on Information Systems*.

Yang, J.-S. and Pan, J. C.-H. (2004). Just-in-time purchasing: An integrated inventory model involving deterministic variable lead time and quality improvement investment. *International Journal of Production Research*, 42(5):853–863.

Yao, D.-Q. (2001). *Study of Bullwhip Effect and Channel Design in Supply Chains*. Ph. D. thesis, University of Wisconsin-Milwaukee (Wisconsin, USA).

Yildizoglu, M. (2003). *Introduction à la Théorie des Jeux*. Dunod, Paris (France).

Yokoo, M. and Hirayama, K. (2000). Algorithms for distributed constraint satisfaction: A review. In *Autonomous Agents and Multi-Agent Systems*, volume 3.

Yu, Z., Yan, H., and Cheng, T. E. (2001). Benefits of information sharing with supply chain partnerships. *Industrial Management & Data Systems*, 101(3).

Yung, S. and Yang, C. (1999a). Intelligent multi-agents for supply chain management. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2.

Yung, S. K. and Yang, C. C. (1999b). A new approach to solve supply chain management problem by integrating multi-agent technology and constraint network. In *Proc. 32nd Hawaii international conference on system sciences*, University of Hong Kong.

Appendix A

Simulation Model of the Québec Wood Supply Game

In this appendix, we describe in detail the implementation of a company in the Québec Wood Supply Game (QWSG) in a spreadsheet program, as an extension of Section 6.2. To this end, we first recall the notations in Section A.1, then we introduce in Section A.2 the equations that do not depend on which ordering scheme is used by the considered company, and finally, we describe the implementation of the seven ordering schemes A, A', A", B, B', C and D in Section A.3. Note that this implementation has also been presented in [Moyaux et al., 2004a, 2003a].

A.1 Recall of Notations

We recall that the notations introduced in Subsection 6.2.1 are as follows:

$To^i_w =$ company i's $o$utgoing Transport in week w.

$Too^i_w =$ company i's $o$utgoing Transport in week w corresp. to current O.

$Tob^i_w =$ company i's $o$utgoing Transport in week w corresp. to backordered O.

$To\Theta^i_w =$ company i's $o$utgoing Transport in w corresp. to current and backordered Θ.

$Ti^i_w =$ company i's $i$ncoming Transport in week w.

$I^i_w =$ company i's Inventory in week w.

Op^i_w = company i's *P*laced *O*rders X in week w.

Oo^i_w = company i's *o*utgoing *O*rders O in week w.

Oi^i_w = company i's *i*ncoming *O*rders O in week w.

Ob^i_w = company i's *b*ackordered O in week w.

Θp^i_w = company i's sent Θ in week w.

Θo^i_w = company i's *o*utgoing Θ in week w.

Θi^i_w = company i's *i*ncoming Θ in week w.

Θb^i_w = company i's *b*ackordered Θ in week w.

$D^{\text{lumber}}_w = Oi^1_w$ = lumber market consumption in Week w.

$D^{\text{paper}}_w = Oi^2_w$ = paper market consumption in Week w.

As stated in Subsection 6.2.1, except the inventory I and the two market consumptions D, the first letter in the name of these variables indicates if the considered variable is for the shipping stream (T for transportation) or for the ordering stream (O or Θ), and the second letter indicates if it is an *i*ncoming, *o*utgoing, *p*laced or *b*ackordered value of this stream. For the sake of simplicity, the quantity of products to ship To is plit into three parts: Too and Tob for orders O, and $To\Theta$ for orders Θ.

We now present the equations used by any companies, next the equations that are used only by the companies that use a particular ordering scheme.

A.2 Scheme-independent equations

In this appendix, for the sake of simplicity, $i = 1$ represents a retailer (any of both retailers) and $i + 1$ is i's supplier, such as $i = 2$ is a wholesaler, even if this is not compatible with Figure 6.3 and with the rest of this dissertation. Therefore, Oi^1_w denotes retailer's demand, that is, the market consumption (D^{lumber}_w or D^{paper}_w), while we have $Oo^1_w = D^{\text{lumber}}_w$ and $Oo^2_w = D^{\text{paper}}_w$ in the rest of this dissertation. Relations between variables, that do not depend on the ordering scheme used, are the same as in [Moyaux et al., 2004c, 2003a,b], except some adaptations needed by Scheme A", that we describe in the next subsection. Notice that the simulation begins in Week $w = 1$.

To_w^n represents products sent to its client i-1 by the company i in week w. To make the calculation of this quantity easy, it is divided into three parts, as indicated by Equation A.1: Too_w^i represents products that are first sent to fulfill the current order (or the quantity of products the company is able to ship when inventory and incoming transports are not enough), as reflected by Equation A.2. Then, the company i ships the quantity Tob_w^i of products to reduce its backorders. Finally, when orders are fulfilled and there is no backorder left, $To\Theta_w^i$ products are sent to reduce backordered Θ, called Θb_w^i (Equation A.3).

$$To_w^i = Too_w^i + Tob_w^i + To\Theta_w^i \tag{A.1}$$

$$Too_w^i = \begin{cases} Oi_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i \geq Oi_w^i \\ I_{w-1}^i + Ti_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i < Oi_w^i \\ Oi_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i \geq Oi_w^i \\ Ti_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i < Oi_w^i \end{cases} \tag{A.2}$$

$$Tob_w^i = \begin{cases} Ob_{w-1}^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i - Too_w^i \geq Ob_{w-1}^i \\ I_{w-1}^i + Ti_w^i - Too_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i - Too_w^i < Ob_{w-1}^i \\ -I_{w-1}^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i - Too_w^i \geq -I_{w-1}^i \\ Ti_w^i - Too_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i - Too_w^i < -I_{w-1}^i \end{cases} \tag{A.3}$$

$$To\Theta_w^i = \begin{cases} -Too_w^i - Tob_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i - Too_w^i \\ & -Tob_w^i \geq \Theta b_{w-1}^i + \Theta i_w^i \text{ and } Too_w^i \\ & +Tob_w^i + \Theta b_{w-1}^i + \Theta i_w^i < 0 \\ \\ \Theta b_{w-1}^i + \Theta i_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i - Too_w^i \\ & -Tob_w^i \geq \Theta b_{w-1}^i + \Theta i_w^i \text{ and } Too_w^i \\ & +Tob_w^i + \Theta b_{w-1}^i + \Theta i_w^i \geq 0 \\ \\ I_{w-1}^i + Ti_w^i - Too_w^i - Tob_w^i & \text{if } I_{w-1}^i \geq 0 \text{ and } I_{w-1}^i + Ti_w^i - Too_w^i \\ & -Tob_w^i < \Theta b_{w-1}^i + \Theta i_w^i \\ \\ -Too_w^i - Tob_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i - Too_w^i - Tob_w^i \\ & \geq \Theta b_{w-1}^i + \Theta i_w^i \text{ and } Too_w^i + Tob_w^i \\ & +\Theta b_{w-1}^i + \Theta i_w^i < 0 \\ \\ \Theta b_{w-1}^i + \Theta i_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i - Too_w^i - Tob_w^i \\ & \geq \Theta b_{w-1}^i + \Theta i_w^i \text{ and } Too_w^i + Tob_w^i \\ & +\Theta b_{w-1}^i + \Theta i_w^i \geq 0 \\ \\ Ti_w^i - Too_w^i - Tob_w^i & \text{if } I_{w-1}^i < 0 \text{ and } Ti_w^i - Too_w^i - Tob_w^i \\ & < \Theta b_{w-1}^i + \Theta i_w^i \end{cases} \tag{A.4}$$

Backorders correspond to products needed by clients, but that cannot currently be shipped. Their role is to memorize that products should have been shipped in the past or in the current week, and that have to be shipped as soon as possible. In the QWSG, backorders are noted as negative inventories, but to simplify the implementation, backorders are rather implemented in two separate variables: Ob_w^i in Equation A.5 represents backorders created by unfulfilled O, and Θb_w^i in Equation A.6 backorders created by unfulfilled Θ.

$$Ob_w^i = Ob_{w-1}^i + (Oi_w^i - Too_w^i) - Tob_w^i \tag{A.5}$$

$$\Theta b_w^i = \begin{cases} \Theta b_{w-1}^i + \Theta i_w^i - To\Theta_w^i + Too_w^i + Tob_w^i + To\Theta_w^i & \text{if } Too_w^i + Tob_w^i \\ & +To\Theta_w^i < 0 \\ \\ \Theta b_{w-1}^i + \Theta i_w^i - To\Theta_w^i & \text{if } Too_w^i + Tob_w^i \\ & +To\Theta_w^i \geq 0 \end{cases} \tag{A.6}$$

Incoming transport is supplier's (i.e., company $i + 1$) last week outgoing transport, therefore:

$$Ti_w^i = To_{w-1}^{i+1} \tag{A.7}$$

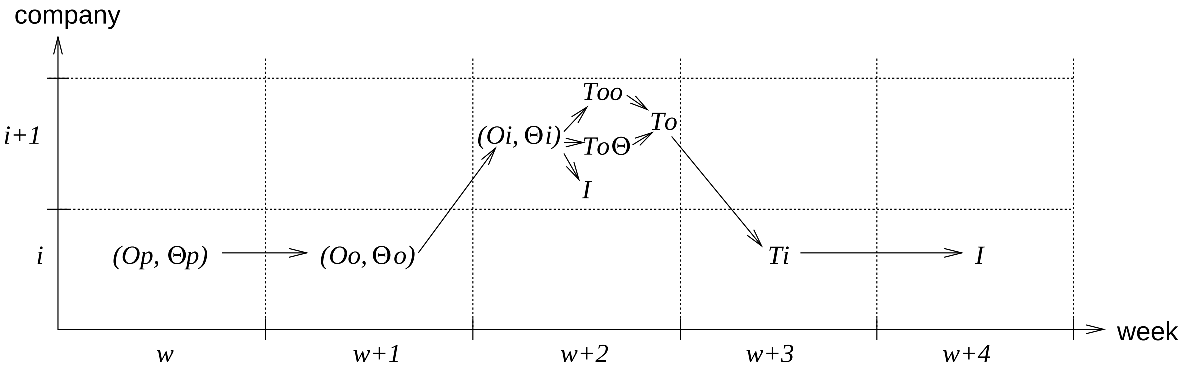


Figure A.1: Some relations between variables (recall of Figure 6.4).

Inventory level is previous inventory level plus inputs minus outputs:

$$I_w^i = I_{w-1}^i + Ti_w^i - To_w^i \tag{A.8}$$

Figure 6.4 (recalled in Figure A.1) shows how orders are delayed between a client i and its supplier $i+1$. Each order is placed in $(Op_w^i, \Theta p_w^i)$, goes next in $(Oo_{w+1}^i, \Theta o_{w+1}^i)$ to simulate the first week of delay, and is finally put in supplier's $(Oi_{w+2}^{i+1}, \Theta i_{w+2}^{i+1})$ to simulate the second week of delay. This explains why incoming order (O, Θ) is the last week client's (company $(i-1)$) outgoing transport (Equations A.9 and A.10).

$$Oi_w^i = Oo_{w-1}^{i-1} \tag{A.9}$$

$$\Theta i_w^i = \Theta o_{w-1}^{i-1} \tag{A.10}$$

Figure A.1 also explains how Oo_w^i and Θo_w^i are setup in Equations A.11 and A.12.

$$Oo_w^i = Op_{w-1}^i \tag{A.11}$$

$$\Theta o_w^i = \Theta p_{w-1}^i \tag{A.12}$$

Every company is setup in the same way, i.e., with Equations A.1 to A.12, except the Sawmill which has the six following pairs of variables, and thus, some of the previous equations are used twice by the Sawmill:

1. $Oi_w^{\text{6-lumber}}$ and $Oi_w^{\text{6-paper}}$;
2. $\Theta i_w^{\text{6-lumber}}$ and $\Theta i_w^{\text{6-paper}}$;
3. $Op_w^{\text{6-lumber}}$ and $Op_w^{\text{6-paper}}$;

4. $\Theta p_w^{\text{6-lumber}}$ and $\Theta p_w^{\text{6-paper}}$;
5. $I_w^{\text{6-lumber}}$ and $I_w^{\text{6-paper}}$;
6. $To_w^{\text{6-lumber}}$ and $To_w^{\text{6-paper}}$.

The Sawmill also has some additional variables Op_w^6 and Θp_w^6 to handle the interface between these pairs of variables and the Sawmill's supplier, i.e., the Forest. $Oo_w^6, \Theta o_w^6$ and Ti_w^6 are unique. In fact, we operate as if there were a paper Sawmill and a lumber Sawmill sharing the same product input Ti_w^6 (each part of the Sawmill receives what is in the incoming transport Ti_w^6, because one unit of wood coming from the Forest gives one unit of lumber and one unit of paper). Then, the rest of the lumber Sawmill is distinct from the paper Sawmill. In particular, the lumber Sawmill would like to place orders $(Op_w^{\text{6-lumber}}, \Theta p_w^{\text{6-lumber}})$, while the paper Sawmill would like to place orders $(Op_w^{\text{6-paper}}, \Theta p_w^{\text{6-paper}})$. A problem is how to aggregate these two needs. In this paper, we assume that:

- $Op_w^6 = (Op_w^{6-lumber} + Op_w^{6-paper})/2$ and $\Theta p_w^6 = (\Theta p_w^{6-lumber} + \Theta p_w^{6-paper})/2$

On the contrary, we have used in our previous papers [Moyaux et al., 2004c, 2003a,b]:

- $Op_w^6 = \max(Op_w^{6-lumber}; Op_w^{6-paper})$ and $\Theta p_w^6 = \max(\Theta p_w^{6-lumber}; \Theta p_w^{6-paper})$

The way to aggregate efficiently $Op_w^{\text{6-lumber}}$ / $Op_w^{\text{6-paper}}$ and $\Theta p_w^{\text{6-lumber}}$ / $\Theta p_w^{\text{6-paper}}$ into Op_w^6 and Θp_w^6 will be studied in Subsection C.4.

A.3 Scheme-dependent equations

The previous equations describe the shipping and ordering flows in the company i. We now focus on the unique decision made by companies in each week, which concerns the choice of O and Θ to place, i.e., Op and Θp. This decision depends on the used ordering scheme: the next equations are scheme-dependent. It is worth noting that incoming Θ (called Θi) are always fulfilled automatically by shipping items to the client as real orders. This is achieved by $To\Theta$ in Equation A.11. In other words, all companies can receive Θ, even when the considered company does not emit Θ.

Scheme A: Since (O, Θ) orders are not applied in Scheme A, Θ is not used, because orders are only placed in O, as reflected by Equations A.13.

$$\theta p^i_w = 0 \tag{A.13}$$

Orders are placed so that to keep a steady inventory, except that negative orders (i.e., order cancellations) are forbidden (and thus, inventory level remains equal or increases over time), as described in Equation A.14.

$$Op^i_w = \begin{cases} 0 & \text{if } Oi^i_w + (I^i_{w-1} - I^i_w) + (Ob^i_w - Ob^i_{w-1}) < 0 \\ Oi^i_w + (I^i_{w-1} - I^i_w) + (Ob^i_w - Ob^i_{w-1}) & \text{else} \end{cases} \tag{A.14}$$

Scheme A′: Companies that use Scheme A′ place orders in a similar way than companies ordering with B (B is our first proposed scheme, that applies our two principles), except that Θ is not used (Equation A.15).

$$\theta p^i_w = 0 \tag{A.15}$$

Over- and underorders to stabilize inventory in Scheme B are achieved by the emission of Θ. Since rule A′ does not use Θ, over- and underorders are achieved in O, in the same way as Scheme B, except that negative orders (i.e., order cancellations) are forbidden, which is implemented by Equation A.16.

$$Op^i_w = \begin{cases} 0 & \text{if } Oi^i_w - \lambda * (Oi^i_{w-1} - Oi^i_w) < 0 \\ Oi^i_w - \lambda * (Oi^i_{w-1} - Oi^i_w) & \text{else} \end{cases} \tag{A.16}$$

We use the same λ as in Scheme B. We have seen in Chapter 7 that this ordering scheme reduces much the bullwhip effect without requiring information sharing, i.e., without requiring the market consumption information to be communicated to companies, but inventory is not managed efficiently.

Scheme A″: This scheme implements an (s, S) ordering policy, and thus, no Θ are sent in Op (Equation A.17.

$$\theta p^i_w = 0 \tag{A.17}$$

Each week, the company checks if inventory is lower than s. When it is the case, the company orders products to fill its inventory up to S, as presented in Equation A.18.

$$Op^i_w = \begin{cases} S - I^i_w & \text{if } I^i_w < s \\ 0 & \text{else} \end{cases} = \begin{cases} Oi^i_w - I^i_w & \text{if } I^i_w < 0 \\ 0 & \text{else} \end{cases} \tag{A.18}$$

We pointed out in Subsection 8.2.2 that we look for the optimal parameters s and S in Appendix B.2. We find that taking $s = 0$ and $S = Oi^i_w$ is optimal for a steady demand, and we carry out our two series of simulations with these parameters, i.e., we do as if the demand remained steady for all the market consumption patterns used in this thesis.

Scheme B: This is the first ordering scheme that we propose to reduce the bullwhip effect by stabilizing order stream as most as possible. Incoming O, which is the market consumption when all clients use Schemes B and/or D, is transmitted to the client, according to the lot-for-lot policy (cf. our first principle), as reflected in Equation A.19.

$$Op^i_w = Oi^i_w \tag{A.19}$$

Next, company's requirements are added to incoming Θ, and this sum is sent as Θ to the supplier (see Equation A.20). $\lambda * (Oi^i_{w-1} - Oi^i_w)$ is an estimation of the inventory decrease caused by the variation of Oi, which is caused by the market consumption variation. We can note that our second principle (i.e., companies order differently from the lot-for-lot policy only when O changes) is fulfilled, because Θ are only sent when O changes.

$$\theta p^i_w = \theta i^i_w - \lambda * (Oi^i_{w-1} - Oi^i_w) \tag{A.20}$$

Two parameters have to be set up in B: λ and initial inventories I^i_1 for every company i. λ is chosen so that the inventory eventually stabilizes on its initial level when the demand has been constant for a sufficient period. λ is a factor proportionnal to the estimated number of required products and to the variation of O. On the other hand, each I^i_1 results from an optimization. How to set up these two parameters is explained in Subsection B.3. In particular, we find that we should use $\lambda = 4$ for all companies. We use empty initial inventories in the first series of experiments, and the values of I^i_1 given in Figure C.7 in the second series of experiments.

Scheme B': This scheme satisfies our first principle (because lot-for-lot orders are used to manage O, as reflected by equation A.21), but not our second principle because companies may react any number of times to each market consumption change by sending Θ. Precisely, Θ are sent to keep a steady inventory like in Scheme A, but this emission of Θ does not depend on the variation of O, as outlined in Equation A.22.

$$Op^i_w = Oi^i_w \tag{A.21}$$

$$\Theta p^i_w = \Theta i^i_{w-1} + (I^i_{w-1} - I^i_w) + (Ob^i_w - Ob^i_{w-1}) \tag{A.22}$$

Scheme C: Scheme C does not use (O, Θ) orders, as reflected by Equation A.23.

$$\Theta p^i_w = 0 \tag{A.23}$$

Next, C is similar to A, except that information centralization is used, like in D. Therefore, Oi^i_w in Equation A.14 is replaced by Oi^1_w in Equation A.24. Since C is only simulated with a homogeneous supply chain (i.e., all companies use the same ordering scheme), we do not need to check whether the retailer of the considered company agrees to share the market consumption information. As we will now see, this simplification is not possible in D, which is the reason of the what-if conditions in Equations A.25 and A.26

$$Op^i_w = Oi^1_w + (I^i_{w-1} - I^i_w) + (Ob^i_w - Ob^i_{w-1}) \tag{A.24}$$

Scheme D: The last scheme is very similar to Scheme B, and is our second scheme proposed to reduce the bullwhip effect. The difference with B is in the way to choose O and Θ: the company using D can base O and Θ on market consumption (i.e., Oi^1_w, because $i = 1$ is one of both markets in this appendix) when information centralization is achieved by the retailer, else this company uses wholesaler's incoming order as a basis when retailer does not multi-cast its incoming order but the wholesaler do,... else the company uses its own incoming order when none of its clients multi-cast its incoming orders, as implemented by Equation A.25:

$$Op^i_w = \begin{cases} Oi^1_w & \text{if the retailer } (i=1) \text{ multi-casts its incoming orders.} \\ Oi^2_w & \text{if the retailer does not multi-cast its incoming orders,} \\ & \text{but the wholesaler } (i=2) \text{ does.} \\ \ldots & \ldots \\ Oi^i_w & \text{if no companies multi-cast its incoming orders.} \end{cases} \tag{A.25}$$

In the same way, Oi^i_w in B (Equation A.20 is replaced by the market consumption Oi^1_w when it is available, or by a client's incoming order (Equation A.26):

$$\theta p^i_w = \begin{cases} \theta i^i_w - \lambda * (Oi^1_{w-1} - Oi^1_w) & \text{if the retailer multi-casts its incoming orders.} \\ \theta i^i_w - \lambda * (Oi^2_{w-1} - Oi^2_w) & \text{if the retailer does not multi-cast its} \\ & \text{incoming orders, but the wholesaler does.} \\ \ldots & \ldots \\ \theta i^i_w - \lambda * (Oi^i_{w-1} - Oi^i_w) & \text{if no company multi-casts its incoming orders.} \end{cases} \tag{A.26}$$

Parameters λ and I^i_1 are chosen like in Scheme B. The optimization of these parameters is explained in Subsection B.4. We find that we should use $\lambda = 2$ for all companies except retailers, and $\lambda = 4$ for retailers. We use empty initial inventories in the first series of experiments, and the values of I^i_1 given in Figure C.7 in the second series of experiments.

A.4 Implementation of the JACK™ agents

We now describe our second simulator. More precisely, we detail the code of the plans presented in Figure 6.9 of Subsection 6.3.3 (this figure is recalled in Figure A.2). These

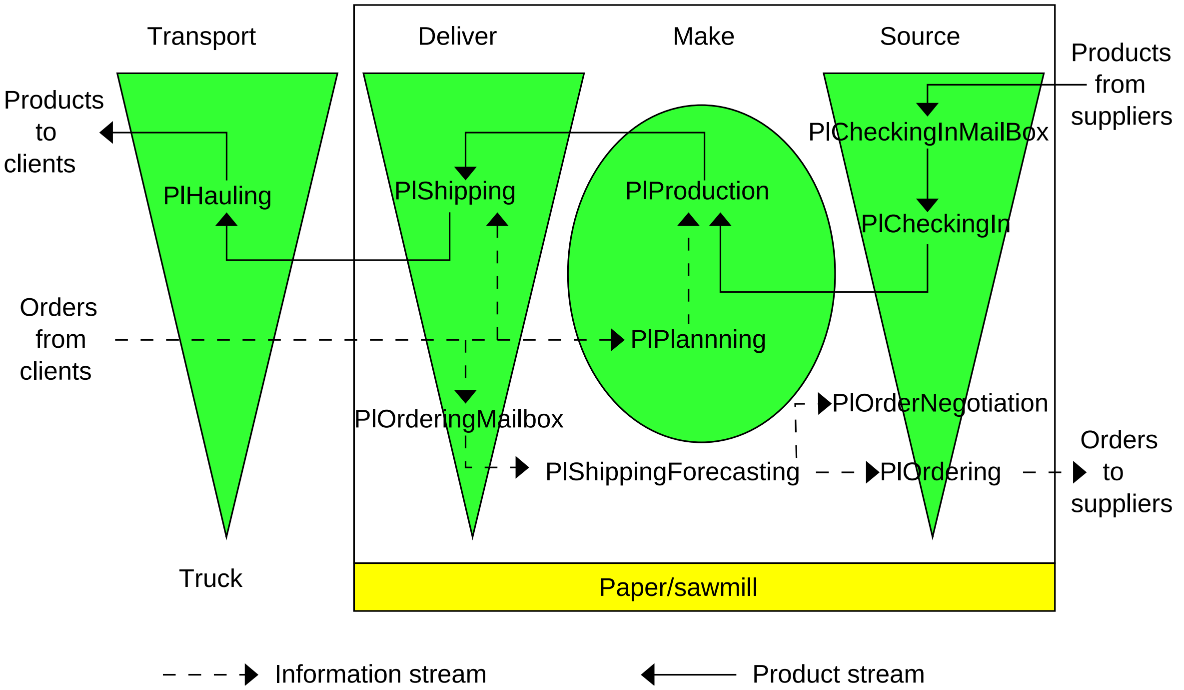


Figure A.2: JACK™ plans in the PaperMill and the Sawmill (recall of Figure 6.9).

plans simulate and improve the five days per week of the QWSG. Since the company-agents are event-driven, every JACK™ plan is triggered by a particular event. For this reason, we first present the events, then the plans in our agents. The events represent physical quantities in the simulation, and are of the following three types:

- EvOrder: These are message events, that represent orders with the following fields:
    1. _placedOrder represents the quantity O;
    2. _placedToken represents the quantity Θ;
    3. _orderedItem is a character string representing the type of the ordered item (paper or lumber).
- EvShipping: These events are messages, that represent shippings with the following fields:
    1. _shippedQuantity represents the quantity of products of the shipping;
    2. _shippedItem is a character string representing the type of the ordered item (paper or lumber).

- EvTime: These events are messages representing the time multicasted by ClockAndGui to drive all company-agents' plans. They have two fields:
  1. _week;
  2. _day.

There are two other types of message events used for interactions between the ClockAndGui agent and the company-agents. These two additional types of message have no physical meaning, but are needed by the simulation in the following way:

- EvAskForUpdateGUI: This message event is sent by company-agents to ClockAndGui in order to display on screen the company-agent's state;
- EvInitGUI: This message is sent by company-agents to ClockAndGui at the beginning of the simulation to initialize the display, and is similar to EvAskForUpdateGUI, except that it also contains capacity parameters.

We now present the plans triggered by these events. We describe these plans below, according to their role (a general description of the function achieved by the rule), preconditions (which event drives the plan) and their action (what the agent does when the precondition is true)[1]. Note that all company-agents' plans are driven by events EvTime, except plans whose name finishes by "MailBox" that are driven be a message event EvOrder or EvShipping.

- PlCheckingInMailBox

  Role: Receive products coming from the supplier's truck and put them in company's raw material inventory.

  Precondition: arrival of products, i.e., an event EvShipping is received.

  Action: the quantity of products indicated in EvShipping is added to company's variable _rawMaterialInventory.

- PlHauling

  Role: deliver products to the company's client.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 1st day in week.

  Action: send to the client an event EvShipping indicating the last quantity shipped in _productsInTruck.

---

[1] See Figure 6.7 (recalled in Figure A.3) for the signification of notations.

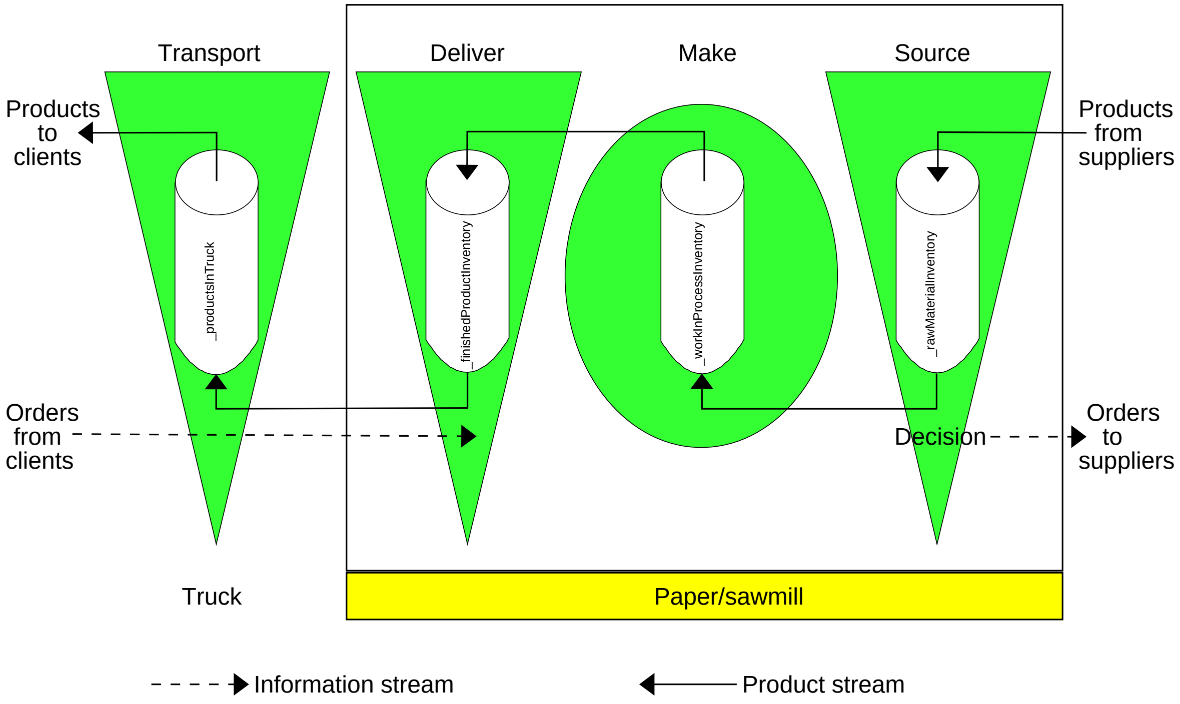


Figure A.3: The PaperMill and Sawmill model (recall of Figure 6.7).

- PlOrderNegociation

  Role: Negociate the price, quantity, shipping date... with the supplier.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 2nd day in week.

  Implementation: This plan has not been implemented.

- PlOrdering

  Role: order to suppliers the current company's demand, i.e., apply a Lot-for-Lot ordering rule without information sharing with (O, Θ) orders.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 3rd day in week.

  Implementation: send to the supplier an event EvOrder indicating the last arrived O and Θ received by PlOrderingMailBox.

- PlOrderingSS

  Role: place an order to suppliers with a (s, S) ordering policy, i.e., the ordering scheme A". This plan replaces PlOrdering.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 3rd day in week and _rawMaterialInventory is lower than s (the parameter in the (s, S) ordering policy).

Implementation: send to the supplier an event EvOrder indicating the difference between S, i.e., the parameter in the (s, S) ordering policy, and _rawMaterialInventory.

- PlOrderingMailBox

  Role: orders placed by clients arrive here. As orders may arrive anytime, they are not received by PlOrdering or PlOrderingSS, but by this MailBox.

  Precondition: arrival of an order, i.e., an event EvOrder is received.

  Implementation: O and Θ indicated in EvOrder are memorized.

- PlProduction

  Role: describe how and when to produce a new batch of products.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 4^{th} day in week.

  Implementation: if production time has elapsed, i.e. _week in EvTime is superior to company's _beginningProductionWeek + _productionDuration, (i) move items from _workInProcessInventory into _finishedProductInventory, and (ii) if a new batch of products can be launched, process it, else set _beginningProductionWeek such as this plan triggers the following week.

- PlShipping

  Role: Ship products demanded in last order by client. There is exactly one incoming order per week because if the client wants nothing, it orders zero.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 5^{th} day in week.

  Action: (i) try to fulfill current O and backordered O, (ii) try to fulfill current Θ and backordered Θ, and (iii) put these four quantities into company's _productsInTruck.

- PlPlanning

  Role: Plan which product to produce if the company processes different kinds of products.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 6^{th} day in week.

  Implementation: This plan has not been implemented.

- PlShippingForecasting

  Role: Anticipate the future demand of the customer.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 7^{th} day in week.

  Implementation: This plan has not been implemented.

All these plans belong to the PaperMill and the Sawmill. As the four companies in the distribution network and the two Customers do not produce anything, the following plan replaces PlProduce:

- PlNoProduction

  Role: move products from _rawMaterialInventory to _finishedProductInventory without delays, when PlProduction is not used.

  Precondition: EvTime sent by the ClockAndGUI agent indicates the 4th day in week.

  Implementation: move items from _rawMaterialInventory into _finishedProductInventory if _finishedProductInventory has enough room.

Finally, both Customers have neither PlHauling nor PlShipping, and they have a special version of the plan PlOrdering:

- PLOrdering

  Role: place orders according to the market consumption pattern.

  Precondition:EvTime sent by the ClockAndGUI agent indicates the 3rd day in week.

  Implementation: send to the supplier (here, a Customer) an event EvOrder indicating the market consumption.

Appendix B

Detail about Simulations with a Homogeneous Supply Chain

This appendix presents the simulation of the homogeneous supply chain, that is, the first series of experiments of Chapter 7. To this end, we first detail our nineteen market consumption patterns in Section B.1.

Then, we show how we choose the two parameters of Scheme A″ in Section B.2, of A′ and B in Section B.3, and of D in Section B.4. We should notice here that we do not optimize the unique parameter (i.e., the initial inventory level) of Schemes A, B′ and C, because we optimize no initial inventory level in this appendix. Indeed, we take empty initial inventories for all schemes in the first series of experiments. We only optimize this parameter for the second series of experiments in Appendix C, because it depends on the method used to calculate cost, that is, on either C^i_{QWSG}, C^i_{improved} (cf. Section 7.2), or $C^i_{\text{realistic}}$ (cf. Paragraph 8.2.1). Since we only use realistic costs in the second series of experiments (i.e., our most realistic series), we do not perform this optimization of initial inventory levels for the first series of experiments.

Finally, we gives all simulation outputs in Section B.5.

B.1 Market Consumption Patterns

Table B.1 gives for each week the consumption of both (lumber and paper) markets under the nine first patterns. Note that both markets have the same consumption, which

Week	1. Step	2. Inv. Step	3. Dirac	4. Inv. Dirac	5. Increase	6. Decrease	7. Weak seas.	8. Medium seas.	9. Strong seas.
1	11	17	11	17	11	57	11	11	11
2	11	17	11	17	11	57	11	11	11
3	11	17	11	17	11	57	11	11	11
4	11	17	11	17	11	57	11	11	11
5	17	11	17	11	12	56	12	12	12
6	17	11	11	17	13	55	11	13	13
7	17	11	11	17	14	54	10	12	14
8	17	11	11	17	15	53	11	11	13
9	17	11	11	17	16	52	12	10	12
10	17	11	11	17	17	51	11	9	11
11	17	11	11	17	18	50	10	10	10
12	17	11	11	17	19	49	11	11	9
13	17	11	11	17	20	48	12	12	8
14	17	11	11	17	21	47	11	13	9
15	17	11	11	17	22	46	10	12	10
16	17	11	11	17	23	45	11	11	11
17	17	11	11	17	24	44	12	10	12
18	17	11	11	17	25	43	11	9	13
19	17	11	11	17	26	42	10	10	14
20	17	11	11	17	27	41	11	11	13
21	17	11	11	17	28	40	12	12	12
22	17	11	11	17	29	39	11	13	11
23	17	11	11	17	30	38	10	12	10
24	17	11	11	17	31	37	11	11	9
25	17	11	11	17	32	36	12	10	8
26	17	11	11	17	33	35	11	9	9
27	17	11	11	17	34	34	10	10	10
28	17	11	11	17	35	33	11	11	11
29	17	11	11	17	36	32	12	12	12
30	17	11	11	17	37	31	11	13	13
31	17	11	11	17	38	30	10	12	14
32	17	11	11	17	39	29	11	11	13
33	17	11	11	17	40	28	12	10	12
34	17	11	11	17	41	27	11	9	11
35	17	11	11	17	42	26	10	10	10
36	17	11	11	17	43	25	11	11	9
37	17	11	11	17	44	24	12	12	8
38	17	11	11	17	45	23	11	13	9
39	17	11	11	17	46	22	10	12	10
40	17	11	11	17	47	21	11	11	11
41	17	11	11	17	48	20	12	10	12
42	17	11	11	17	49	19	11	9	13
43	17	11	11	17	50	18	10	10	14
44	17	11	11	17	51	17	11	11	13
45	17	11	11	17	52	16	12	12	12
46	17	11	11	17	53	15	11	13	11
47	17	11	11	17	54	14	10	12	10
48	17	11	11	17	55	13	11	11	9
49	17	11	11	17	56	12	12	10	8
50	17	11	11	17	57	11	11	9	9

Table B.1: Details of the nine lumber consumptions patterns in Table 7.1.

is not the case under the ten other patterns in Tables B.2 and B.3. Indeed, Table B.2 details the lumber market consumption under the ten Uniform random patterns, and Table B.3 describes the paper consumption under these ten same patterns. Tables B.2 and B.3 are two different instances of the same uniform random distribution over $[11; 17]$. We generate these ten instances once in order to simulate our seven ordering schemes under the same demands.

We now indicate how we have set up the parameters in Scheme A".

B.2 Parameters of Scheme A"

Since Scheme A" is an (s, S) ordering policy (we recall that, in this policy, each company orders for $S - I$ products when its inventory level I is lower than s), we can use a mathematical model called Economic Order Quantity (EOQ) to optimize its parameters s and S. We recall that we stop using notations introduced in Subsections 6.2.1 and we use instead notations of the EOQ model presented in Paragraph 2.1.2. The EOQ model describes the evolution of the level of an inventory system. With this description, we can determine the optimal quantity Q^* to order and the optimal backorder level b^* to minimize the cost of the considered inventory system. After, s and S are deduced of Q^* and b^*, which is our goal.

We show first why we need to adapt the classic EOQ model to our simulation of the QWSG, then we actually perform this adaptation.

B.2.1 Why the Classic EOQ Model Has to Be Adapted to the QWSG

We first try to apply directly the results in Paragraph 2.1.2 to fit to our simulation of the QWSG. To this end, we evaluate Q^* in Equation 2.8 (recalled in Equation B.1), and b^* in Equation 2.9 (recalled in Equation B.2).

$$Q^* = \sqrt{\frac{2AD}{h(1 - D/P)} - \frac{(\pi D)^2}{h(h + \hat{\pi})}} \sqrt{\frac{h + \hat{\pi}}{\hat{\pi}}} \tag{B.1}$$

$$b^* = \frac{(hQ^* - \pi D)(1 - D/P)}{h + \hat{\pi}} \tag{B.2}$$

When we know Q^* and b^*, we find s and S as follows:

Week	10A	10B	10C	10D	10E	10F	10G	10H	10I	10J
1	16	15	11	16	13	16	17	12	15	14
2	13	15	15	13	12	13	15	13	14	15
3	16	12	16	12	16	13	16	16	12	15
4	11	17	15	12	14	16	17	17	11	12
5	15	16	15	16	14	12	12	16	14	15
6	14	12	16	15	14	12	13	14	12	12
7	17	16	16	14	17	12	13	13	11	13
8	16	17	13	13	11	13	16	13	12	13
9	13	15	13	15	12	17	16	14	16	16
10	15	17	11	15	12	13	16	12	12	15
11	17	15	13	15	15	12	11	17	16	16
12	12	16	14	12	15	11	16	12	16	17
13	15	16	15	16	12	12	14	14	14	11
14	11	14	15	11	14	12	12	11	15	11
15	15	16	12	13	13	14	12	16	11	17
16	11	16	12	11	16	15	14	17	17	15
17	15	13	13	15	17	15	13	16	14	12
18	12	14	16	14	11	12	15	16	14	16
19	14	11	16	15	13	15	16	12	12	15
20	11	14	17	12	12	16	13	16	15	16
21	13	13	12	14	16	11	16	14	15	17
22	13	12	11	11	16	12	14	12	14	17
23	12	16	12	11	16	15	12	14	17	14
24	13	11	13	14	16	15	16	16	13	14
25	15	12	16	14	13	16	11	15	16	13
26	17	14	16	14	15	16	11	11	13	11
27	17	15	15	15	12	16	17	15	16	13
28	14	13	15	14	17	14	16	11	11	16
29	14	15	15	14	15	12	14	16	16	14
30	13	16	15	15	17	11	16	16	13	16
31	12	14	12	12	12	12	16	14	14	14
32	13	13	15	11	14	12	16	16	15	13
33	12	16	15	13	16	16	15	15	16	14
34	15	14	14	15	13	11	15	12	14	13
35	15	11	12	14	12	13	12	15	12	14
36	13	15	11	12	13	12	16	16	16	14
37	16	16	14	14	17	12	14	11	13	16
38	15	13	12	17	14	13	11	14	15	12
39	15	16	13	17	16	15	12	12	14	13
40	15	16	14	15	17	14	12	13	14	15
41	12	14	16	13	11	12	16	11	14	13
42	11	15	12	12	15	13	14	12	11	15
43	17	17	16	17	12	11	13	12	13	11
44	13	11	12	13	15	17	16	13	11	16
45	13	11	16	16	15	13	16	14	13	15
46	15	15	16	12	15	13	17	13	12	13
47	15	12	14	11	16	14	15	12	11	17
48	12	13	14	16	12	16	16	15	16	16
49	16	12	13	14	13	17	17	12	16	15
50	15	12	14	15	11	14	17	16	13	16

Table B.2: Details of the ten Uniform random 10A to 10J consumptions patterns for the lumber.

Week	10A	10B	10C	10D	10E	10F	10G	10H	10I	10J
1	15	15	14	17	15	14	12	17	12	13
2	15	15	14	12	16	17	17	12	11	12
3	14	14	16	13	14	14	16	13	17	13
4	15	12	15	12	16	16	12	12	16	13
5	14	12	12	13	17	16	12	13	14	11
6	16	15	15	11	13	14	16	16	13	12
7	15	16	17	13	16	15	13	15	16	12
8	12	16	14	12	15	15	13	11	11	12
9	12	14	12	11	15	16	14	16	14	14
10	15	15	15	12	17	17	16	15	16	15
11	16	15	13	14	15	14	11	16	16	12
12	15	15	15	14	12	17	15	15	14	12
13	13	17	11	12	12	13	12	11	11	13
14	15	13	13	14	15	13	17	13	12	11
15	17	14	11	13	15	13	16	12	15	17
16	14	14	12	12	16	15	14	11	14	16
17	12	16	17	11	13	14	13	15	17	15
18	12	13	13	13	12	14	14	13	13	15
19	12	13	15	16	11	16	15	15	15	15
20	13	16	12	16	12	14	13	13	15	12
21	15	12	15	11	15	11	16	15	13	16
22	14	16	13	13	13	12	15	14	12	16
23	16	14	15	16	12	15	14	16	15	16
24	12	14	16	13	15	17	11	13	12	11
25	17	13	15	15	13	14	13	12	13	12
26	15	14	14	17	14	16	16	15	14	12
27	13	11	13	11	17	13	17	17	16	15
28	14	16	16	12	11	12	15	11	14	11
29	16	17	15	15	17	13	15	13	13	14
30	11	13	16	14	15	14	17	13	12	16
31	16	13	16	14	16	12	16	11	14	14
32	12	15	15	15	13	15	15	11	14	16
33	16	12	14	15	16	14	15	16	14	14
34	15	12	16	17	15	11	12	15	15	15
35	13	14	15	16	16	12	15	15	14	11
36	17	15	14	13	12	13	16	15	17	14
37	17	13	15	14	16	17	13	14	13	11
38	16	13	17	14	17	13	13	15	15	15
39	14	15	17	13	15	16	15	15	17	13
40	16	15	12	13	14	11	15	11	14	12
41	11	16	11	14	17	13	16	16	13	15
42	11	14	14	16	12	13	12	14	12	12
43	16	16	12	13	14	14	12	12	17	15
44	12	16	13	11	11	14	17	13	12	13
45	17	12	16	14	12	12	12	15	17	12
46	13	13	12	11	16	12	15	16	15	12
47	14	14	12	15	12	11	11	12	12	15
48	14	15	13	12	12	13	15	11	15	15
49	11	12	17	14	12	13	12	14	14	13
50	13	17	15	13	11	16	16	16	16	16

Table B.3: Details of the ten Uniform random 10A to 10J consumptions patterns for the paper.

- s is given by Equation 2.10, recalled in Equation B.3. Note that $m = [\tau/T]$, where $[x]$ is the integer part of x, e.g., $[1.9] = 1$;

$$s = \begin{cases} \tau D - mQ - b & \text{if } \tau - mT \leq T_3 + T_4 \\ ((m+1)(Q/D) - \tau)(P - D) - b & \text{else} \end{cases} \tag{B.3}$$

- S is calculated with Equation B.4.

$$S = s + Q^* \tag{B.4}$$

In our simulation of the QWSG, we assume that:

- there is no production, that is, production rate is infinite: $P \rightarrow +\infty$;
- since there is no production, production cost is zero: $C = 0$;
- order placing is free: $A = 0$;
- the inventory carrying cost per unit per week is taken into account: $h \neq 0$ (e.g., $h = \$1$ in C_{QWSG}).
- the duration-independent shortage cost is zero: $\pi = 0$;
- the shortage cost per item short week is taken into account: $\hat{\pi} \neq 0$ (e.g., $\hat{\pi} = \$2$ in C_{QWSG});
- the time between the placement and the receipt of an order is four week: $\tau = 4$.

A problem occurs when we replace P, C, A and π by the previous values in Equation B.1 and B.2 in order to find the optimal values of Q^* and b^* in the QWSG. In fact, if $A = \pi = 0$ then $Q^* = 0$ in Equation B.1.

This result is compliant with real-life: when order placing is expensive (great A), companies accept to pay higher inventory holding cost, that is, companies order for more products (great Q) in order to order less often. On the other hand, if order placing is free ($A = 0$), companies would like to order all the time (time is continue in the basic EOQ, and thus $T = Q/D = 0$ is possible), but for very few items (almost zero), and therefore $Q = 0$ minimizes the annual inventory system cost. Unfortunately, time is not continue in the QWSG and $T = Q/D = 0$ is thus impossible: companies are allowed to place each week at most one order, rather than an infinite number of orders.

We can also see this problem from another viewpoint. The QWSG does not take ordering cost into account ($A = 0$), and therefore, the total cost is only made of costs proportional to Q, which are minimum when $Q = 0$. We recall that the idea of EOQ is that the optimal order quantity Q^* corresponds to a trade-off between costs proportional to Q (inventory holding, backorder...) that increase with Q, and costs proportional to $1/Q$ (ordering cost...) that decrease with Q. Since we only have costs proportional to Q in the QWSG, the optimal value is $Q^* = 0$ and the total cost of the inventory system is zero. But we have just seen that $Q^* = 0$ is not possible, because it implies $T = Q/D = 0$ and time continuity. Moreover, it is obvious that the company needs to order for $Q^* > 0$ in order to fulfill its demand!

To solve this problem, we rewrite entirely the EOQ model under discrete time from the beginning.

B.2.2 Adaptation of the EOQ model to the QWSG

We adapt the EOQ model from its most basic formulation, rather than from the more complete one used above and described in Subsection 2.1.2. Precisely, backorders and production costs are not taken into account. Like in the classic EOQ model, the following assumptions apply [Hax and Candea, 1984]:

1. demand is continuous at a constant rate;
2. the process continues infinitely;
3. no constraints are imposed (on quantitites ordered, storage capacity, available capital, etc,);
4. replenishment is instantaneous (the entire order quantity is received all at once as soon as the order is released)[1];
5. all costs are time-invariant;
6. no shortages are allowed;
7. quantity discounts are not available.

---

[1] We have seen in Subsection 7.3 that this assumption may be a problem in supply chain management, because the supplier may have backorders. This assumption is also a problem in the QWSG and in our simulations. Unfortunately, removing this assumption requires to model the suppliers' inventory system, in order to know when it is backordered, which is not an easy problem. However, the models called "Joint Optimization" in Subsection 4.1.1 allow doing so.

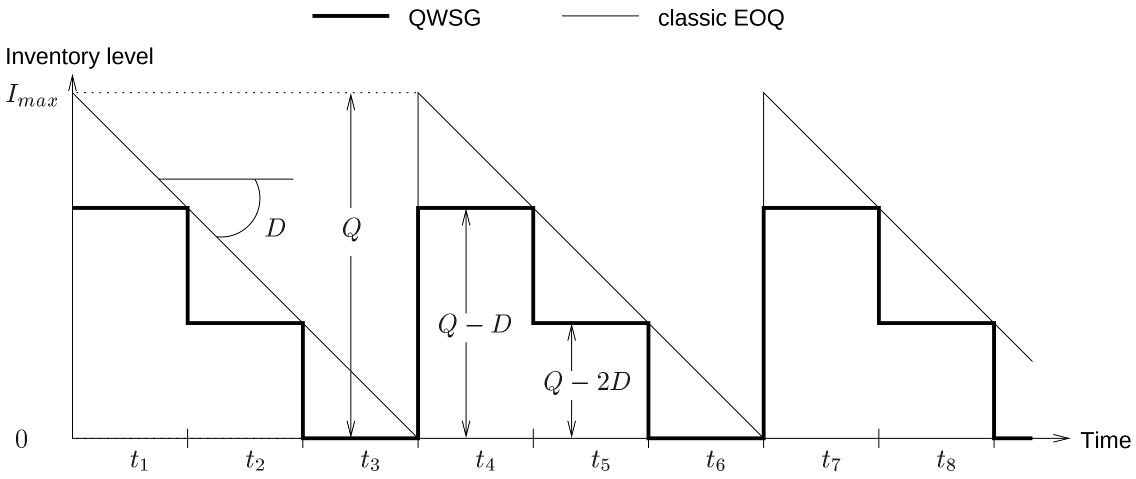


Figure B.1: A cycle for the inventory system in the QWSG.

In this context, Figure B.1 represents the behaviour of the considered inventory system when demand is discrete: the thin line represents the inventory level of a real inventory system (like in the classic EOQ model), while the thick line represents the inventory considered in the QWSG. The difference between the reality and the QWSG is due to the discretization of time in the QWSG. We use the same notations as the classic EOQ model:

- Q is the ordered quantity, i.e., the quantity of incoming transports;
- D is the demand rate, i.e., the number of products demanded by the client per time unit t_w (there are fifty time units t_1 to t_{50} during a fifty week simulation).

The company receives Q products at the beginning of t_4 but only pays for holding $Q - D$ products in inventory (cf. level indicated by the thick line) because D products are shipped to the client. Precisely, Q products are in inventory during a part of t_4, but we ignore that because time is discrete, and we only consider the inventory level at the end of t_4.

The cycle $\{t_1, t_2, t_3\}$ is identical to the cycle $\{t_4, t_5, t_6\}$. The duration of each cycle is Q/D; to see that, consider that the less you order in Q, the more frequent you place orders to fulfill the demand D. A fifty-week simulation has thus $50 * D/Q$ cycles. We note w the week such as t_w is Week w, e.g., Week $w = 1$ in t_1. The charged inventory level in Week t_w is $Q - wD$, which thus costs $h(Q - wD)$. Since we note h the inventory carrying cost per item per week, the annual total cost K of the inventory system is therefore given by Equation B.5 (note that the annual total cost K corresponds to C^i

in the rest of this dissertation).

$$K = 50\frac{D}{Q}\sum_{w=1}^{Q/D} h(Q - wD) = 50\frac{D}{Q}\left[hQ\frac{Q}{D} - hD\frac{\frac{Q}{D}(\frac{Q}{D}+1)}{2}\right] = 25h(Q - D) \tag{B.5}$$

Clearly, $K = 0$ when $Q^* = D$, that is, the inventory system incurs no cost when the company orders what is ordered by its client. Since our model does not accept backorders (cf. assumption (6) above), $Q < D$ is not allowed, and therefore $K > 0$, which is a confirmation that the company cannot earn money by holding inventory! Since we cannot obtain a lower K than $K = 0$, we do not try to relax the assumption (6) (no backorders allowed) by checking if companies should sometimes incur backorder to save money. Since the duration of a cycle is $Q^*/D = D/D = 1$, the company has to place an order each week, and not only at Weeks t_3 and t_6 as illustrated in Figure B.1 (i.e., Figure B.1 does not represent the optimal behaviour of the inventory system in the QWASG).

Finally, we can now apply Equations B.3 and B.4 to determine s and S in Scheme A″. In our case, $m = [\tau/T] = [4/1] = 4$, next $\tau - mT = 4 - 4 * 1 = 0$ and $T_3 + T_4 = 1 + 0 = 1$, thus $\tau - mT \leq T_3 + T_4$. Therefore, $s = \tau D - mQ^* - b = 4D - 4D - 0 = 0$. Next, according to Hax and Candea [1984], $S = s + Q^* = 0 + D = D$.

If we stop using notations specific to the model EOQ and we use instead the same notations as the rest of this dissertation, the optimal parameters are $s = 0$ and $S = Oi^i_w$, where Oi^i_w is the company i's incoming order in Week w. Moreover, all initial inventories are empty: $\forall i, I^i_1 = 0$. We now present the seting of the parameters in Scheme B.

B.3 Parameters of Scheme B

Scheme B has two parameters. The first parameter, λ, represents the quantity of Θ to send to stabilize eventually the inventory on its initial level when the market consumption has been steady for a sufficient period. This parameter only depends on the ordering and shipping delays between the considered company and its direct supplier. The second parameter, I^i_1 (company i's inventory in the first week), represents the initial inventory level. This parameter depends on the structure of the supply chain and on the cost function. As stated in the introduction of this appendix, we only seek the good value of λ, because we assume in our first series of experiments (cf. Chapter 7) that all initial inventories are empty. However, Appendix C.1 will show how we carry out this optimization for our second series of experiments.

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	2	3	4	5
2 week ordering delay	3	4	5	6
3 week ordering delay	4	5	6	7

Table B.4: Correct λ for ordering scheme B for different ordering and shipping delays for all market demand patterns (also applicable for retailers using Scheme D).

B.3.1 Value of λ

The longer the delay between the order placement and the shipping reception is, the greater the inventory level fluctuates. Since Θ is sent to counter this inventory fluctuation, the value of Θ only depends on the ordering and shipping delay. All companies use the same value of λ to calculate Θ. Table B.4 gives the value of λ to use. These values are determined by simulations in which we adapt λ for each company such as inventory levels eventually stabilize on their initial levels under the Step market consumption pattern (we use this pattern, because it is steady after a unique change at the beginning of the simulation). For example, the last entry in Table B.4 is 7, which means that all companies must use $\lambda = 7$ when there is a four week shipping delay and a three week ordering delay. This figure suggests that λ is equal to the sum of the ordering delay and of the shipping delay.

In short, the accurate value of λ only depends on the delays between the considered company and its direct supplier. The determination of this value does not require any optimization. The second parameter depends, at least, on the market consumption and has therefore to be adapted to the situation.

B.3.2 Initial inventory levels

As previously stated, we take empty initial inventories in the first series of experiments: for any company i, $I_1^i = 0$.

We now carry out the same job with Scheme D.

B.4 Parameters of Scheme D

The rule D is very similar to B and has therefore the same two parameters λ and I_1^i to be set up. These two setups have to be carried out in the same order: first λ, next I_1^i.

B.4.1 Value of λ

We have seen with Scheme B that λ is equal to the sum of ordering and shipping delays. Since Scheme D uses information centralization, market consumption information travels instantaneously and in real-time in the supply chain, and therefore, the ordering delay has not to be taken into account to set λ. Consequently, λ is now only equal to the shipping delay.

This is true for all companies, except for both retailers. In fact, retailers have to use the same λ as with Scheme B, i.e., $\lambda^{\text{retailer}}$ is equal to the sum of the shipping delay and of the ordering delay. This difference between retailers and all other companies is due to the fact that retailers have to overorder more (by sending $\Theta \neq 0$) than other companies because they do two things when the market consumption changes:

- they emit Θ;
- they ship products corresponding to the new consumption.

On the contrary, all other companies only emit Θ. In fact, orders adressed to other companies will have the new value of the market when the ordering delay will have elapsed. In particular, wholesalers have to wait the ordering delay before they receive orders from their retailer with the new value of the market consumption. Other companies also have to wait the ordering delays before they receive orders with the new market consumption. At this time, other companies will ship products corresponding to the new consumption, but they do so some weeks after retailers[2]. Therefore, retailer's inventory fluctuates more than others' inventory, and thus, they have to send more positive or negative Θ. Table B.5 gives the value of λ that eventually stabilizes inventories on their initial level for all companies except retailers. Retailers still take λ in Table B.4.

---

[2] As a consequence, the ordering stream in the supply chain reacts instantaneously to the changes of the market consumption, while the ordering stream only react two weeks later. It could be interesting to find an ordering scheme in which the ordering stream also reacts instantaneously to the change of the market consumption, because this should increase the efficiency of the supply chain.

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	1	2	3	4
2 week ordering delay	1	2	3	4
3 week ordering delay	1	2	3	4

Table B.5: Correct λ for ordering scheme for all market demand patterns D for different ordering and shipping delays (not applicable to retailers).

B.4.2 Initial inventory levels

Like B, we take empty initial inventories for Scheme D in Chapter 7: for every company i, $I_1^i = 0$.

We now detail the simulation outcomes that were only outlined in Chapter 7.

B.5 Simulation outcomes

This section details for all companies the simulation outcomes, that were only presented for the entire supply chain or for the Sawmill in Section 7.3. Similarly to Chapter 7, these simulations are carried out in the first series of experiments with a homogeneous supply chain, that is, all companies place orders with the same ordering scheme. The goal of this first series is to check the efficiency for reducing the bullwhip effect of our two ordering schemes, and therefore, of the two principles on which they are based. As the bullwhip effect is measured as the standard deviation of placed orders, Tables B.6, B.7, B.8 and B.9 first present this metric. Next, we focus on costs induced by the bullwhip effect when the "improved" method for cost evaluation is applied. Tables B.10, B.11, B.12 and B.13 detail all company's costs. Then, we focus on the two parameters aggregated in this cost: backorders and inventory levels. Tables B.14, B.15, B.16 and B.17 first give each company's sum of backorders, while Tables B.18, B.19, B.20 and B.21 show inventory fluctuations measured as a standard deviation.

	Market consumption	Scheme A	Scheme A′
1 Step	1.6	14.3; 13.3; 28.2; 26.5; 51; 42.6	3.8; 3.8; 20.7; 20.7; 105.7; 274.6
2 Inversed Step	1.6	7.9; 7.9; 17.9; 17.3; 33.2; 37.3	2.3; 2.3; 6.7; 6.7; 37.4; 97.9
3 Dirac	0.8	4.1; 3.9; 8.8; 8.8; 18.7; 15.8	4.5; 4.5; 22.1; 22.1; 111.8; 301.2
4 Inversed Dirac	0.8	5.4; 4.8; 14.1; 12.1; 25.6; 25.2	4.2; 4.2; 26.9; 26.9; 142.8; 372
5 Increase	14.4	33.4; 30.6; 59.6; 54.9; 108.1; 85.6	14.9; 14.9; 15.4; 15.4; 22.2; 45.1
6 Decrease	14.4	14.7; 14.7; 18.6; 17.4; 27.9; 31	14.9; 14.9; 15.4; 15.4; 20.3; 41.1
7 Weak seasonality	0.7	1.2; 1.3; 1.8; 2.3; 3.6; 2.7	4.4; 4.4; 19.7; 19.7; 98.4; 228.6
8 Medium seasonality	1.2	4.5; 6.1; 10.6; 13.6; 25.3; 28.1	4.5; 4.5; 15; 15; 73.7; 186.7
9 Strong seasonality	1.8	7.8; 6.3; 16.4; 13.2; 27.1; 30.4	4.7; 4.7; 13; 13; 59.3; 151.8
10A Uniform random	1.8	9.1; 6.9; 20; 15.4; 30.3; 27.4	11.1; 11; 54.9; 55.9; 275.4; 601.1
10B Uniform random	1.7	9.3; 10.6; 20.1; 21.8; 39.1; 36.5	10.4; 9.5; 54.1; 42.5; 206.4; 524.8
10C Uniform random	1.7	10.7; 9; 22.1; 19.2; 36.1; 42.7	8.7; 9.9; 42.6; 44.9; 215.1; 541.7
10D Uniform random	1.7	9.8; 9; 22.3; 19.2; 35; 51	10.4; 9; 49.5; 40.6; 181.8; 445.3
10E Uniform random	1.9	5.2; 9.9; 12.2; 20.9; 38.9; 35.2	10.8; 10.7; 53.2; 50.5; 244.3; 633.3
10F Uniform random	1.8	8.6; 9; 18.8; 19; 35.3; 33.1	9,4; 9.4; 42.9; 42.9; 214.4; 579.7
10G Uniform random	1.9	7.4; 9.1; 18; 19.3; 37.9; 33.4	10.2; 11.4; 52; 53.5; 246.1; 572.9
10H Uniform random	1.9	8.2; 11.2; 18.3; 23.7; 39.5; 45.8	11; 10.2; 57.4; 49.7; 243.6; 628.2
10I Uniform random	1.8	9.6; 9.4; 21.3; 19.2; 36.1; 33.9	11.6; 10.9; 59.9; 54.1; 263; 676.1
10J Uniform random	1.8	7.3; 6.7; 16.9; 15; 28.9; 30.6	10.1; 10.1; 50.7; 51.9; 261.4; 638.5

Table B.6: Standard-deviation of orders, i.e., $\sigma_{Op^i_w + \Theta p^i_w}$ (extension of Table 7.2, 1/4).

B.5.1 Comparison of the Bullwhip Effect

Let us look at the first metric. Each entry in Tables B.6, B.7, B.8 and B.9 presents the standard deviation of orders placed by each company in the order of Figure 6.3. For example, for Scheme A under the Step demand in Table B.6 (which also corresponds to the curves in Figures 7.1 and 7.2 in Section 7.3), 14.3; 13.3; 28.2; 26.5; 51; 42.6 means that the orders placed by the PaperMill have a standard deviation of 51, while the second column in Table B.6 says that the standard deviation of the market consumption was only 1.6. Therefore, orders placed by the PaperMill in this experiments are much more variable than the market consumption.

These data have to be compared line by line in the four Tables B.6, B.7, B.8 and B.9 (there are four tables because there are not enough room on the paper, but these tables could be merged in a unique one). For example, under the Step demand, the Sawmill's standard deviation of orders is 42.6 with Scheme A, 274.6 with Scheme A′, 795.7 with Scheme A″, 8.7 with Scheme B, 58.5 with Scheme B′, 4.9 with Scheme C and 4.2 with Scheme D. Therefore, Scheme D has the lowest bullwhip effect among these seven ordering schemes, followed by C and B. When we repeat this reasoning for the other companies and for the eighteen other market consumption patterns, we note that the conclusion of Subsection 7.3.2 holds for every company. In fact, Scheme D generally incurs the lowest bullwhip effect among the seven tested ordering schemes, except for the ten random distributions of demand in which C is as good as D. When information centralization is not used, B incurs the lowest bullwhip effect (C is better

	Scheme A''
1 Step	27; 31.7; 105.1; 155.7; 547.7; 795.7
2 Inversed step	17.8; 15.8; 64.5; 74.2; 345.6; 530.7
3 Dirac	14.4; 12.7; 20.7; 24.8; 90.8; 132.7
4 Inversed Dirac	30.7; 35.5; 145.2; 200.8; 778.2; 1155.2
5 Increase	57.4; 67.8; 105.8; 181.3; 590.8; 861.3
6 Decrease	73; 77.2; 351.3; 450.5; 1,928.2; 2915.9
7 Weak seasonality	16.8; 15.6; 48.7; 61.2; 205.5; 278.9
8 Medium seasonality	13.4; 15.8; 17.7; 37.6; 38.7; 57.9
9 Strong seasonality	13.9; 13.3; 22.4; 23.7; 100.7; 155.5
10A Uniform random	25.9; 28.3; 126.9; 149.3; 576.1; 828.6
10B Uniform random	26.3; 28.9; 127; 162; 644.8; 960.8
10C Uniform random	20.3; 19.7; 63.4; 50.5; 43.3; 112
10D Uniform random	22.7; 20.4; 107.4; 93.2; 349.2; 502.1
10E Uniform random	19.2; 16.5; 55.5; 23.2; 64; 115.5
10F Uniform random	18,7; 16.1; 28; 39; 125.9; 179.2
10G Uniform random	24; 19.7; 60.7; 72.2; 281.8; 453.7
10H Uniform random	18.5; 23.8; 41.2; 112.1; 378.6; 642.6
10I Uniform random	26.3; 32.2; 97.8; 151.8; 540.1; 835.7
10J Uniform random	22.4; 27.6; 75.4; 173.7; 754.5; 1240.1

Table B.7: Standard-deviation of orders, i.e., $\sigma_{Op^i_w+\Theta p^i_w}$ (extension of Table 7.2, 2/4).

than B, but C uses information centralization). Therefore, our two principles on which B and D are based are not contradicted, except when demand is random and information centralization is applied, in which case, our two principles need to be enhanced to take the efficiency of C into account.

B.5.2 Comparison of the Costs Incurred

The second metric, the induced costs, is an indirect measure of the bullwhip effect reduction. However, it is more important for companies than the standard deviation of orders, because their goal is to maximize their profit rather than reducing the bullwhip effect. Therefore, Tables B.10, B.11, B.12 and B.13 present the costs for the same experiments than Tables B.6, B.7, B.8 and B.9.

Numbers in the tables are costs for each company evaluated with the method called "improved" in Section 7.2: each unit in inventory costs weekly half the cost of backordered units. Data in Tables B.10, B.11, B.12 and B.13 are thus the individual and costs of inventories and backorders in the whole supply chain for fifty weeks. The format of the data is $C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$, e.g., for the Scheme A under the Step demand in Table B.10, the PaperMill has a cost $C^5 = 8,526$ \$ and the whole supply chain $C = 40,055$ \$.

These costs have to be compared line by line in Tables B.10, B.11, B.12 and B.13

	Scheme B	Scheme B′
1 Step	3.8; 3.8; 7.2; 7.2; 10.6; 8.7	24.6; 28.8; 48.4; 45; 42.7; 58.5
2 Inversed step	3.8; 3.8; 7.2; 7.2; 10.6; 8.7	22.4; 10.5; 29.3; 21.3; 28.8; 100.4
3 Dirac	5.5; 5.5; 10.3; 10.3; 15.2; 12.6	6.5; 6.2; 9.3; 12; 17; 18.4
4 Inversed Dirac	5.5; 5.5; 10.3; 10.3; 15.2; 12.6	27.6; 20.6; 40.6; 33.9; 43.5; 66.5
5 Increase	14.9; 14.9; 15.7; 15.7; 16.7; 17.3	14.8; 15.8; 15.5; 17; 16.6; 15.6
6 Decrease	14.9; 14.9; 15.7; 15.7; 16.7; 17.3	14.7; 14.7; 20; 18.6; 41.7; 79.1
7 Weak seasonality	4.4; 4.4; 8.1; 8.1; 11.6; 2.2	4.5; 4.8; 10.5; 10.1; 10.6; 22.7
8 Medium seasonality	4.5; 4.5; 8.1; 8.1; 11.7; 9.6	16.7; 7.8; 27; 14.2; 15.2; 30.1
9 Strong seasonality	4.7; 4.7; 8.3; 8.3; 11.7; 10.9	12.7; 10.6; 17; 16; 19.8; 18.9
10A Uniform random	12.1; 12.1; 22.2; 22.3; 32.8; 21.8	17.7; 22.7; 25.6; 30.8; 37.7; 81
10B Uniform random	11.4; 10; 21.4; 17.5; 25.5; 20.7	32.2; 19.7; 46.1; 28.9; 38.8; 89.9
10C Uniform random	9.4; 10.6; 17.5; 19.1; 28.1; 22.1	33.9; 21.7; 43.7; 33.5; 41.4; 130.9
10D Uniform random	10.9; 10.2; 20.4; 19.1; 26.6; 21.4	34.5; 25.4; 53.6; 34.8; 50.1; 87.6
10E Uniform random	12.6; 11.8; 23.5; 22; 31.5; 26.6	13.9; 23.9; 21.5; 28.6; 51.9; 110.2
10F Uniform random	10; 10; 18.4; 18.4; 26.8; 23.2	13.6; 14.6; 24; 28.3; 37.6; 28.1
10G Uniform random	11.2; 12.4; 21; 22.5; 31.5; 25.3	8.4; 15.4; 21.1; 22.8; 23.1; 24.6
10H Uniform random	12; 11.6; 21.7; 21.3; 30.5; 25.7	65.1; 39.8; 83.4; 56.9; 68.9; 116.9
10I Uniform random	12.2; 11.7; 22.7; 21.9; 31.3; 30.2	20.7; 32.1; 32.9; 42; 46.8; 100.3
10J Uniform random	11; 10.8; 20.5; 19.7; 28.4; 21.3	13.3; 13.6; 21.5; 24.2; 33.5; 27.5

Table B.8: Standard-deviation of orders, i.e., $\sigma_{Op^i_w + \Theta p^i_w}$ (extension of Table 7.2, 3/4).

(again, these four tables could be merged in a unique one): for example, under the Step demand, the overall cost is $C = 40,055$ \$ for Scheme A, $C = 181,641$ \$ for Scheme A′, $C = 1,152,612$ \$ for Scheme A″, $C = 9,035$ \$ for Scheme B, $C = 29,556$ \$ for Scheme B′, $C = 21,833$ \$ for Scheme C and $C = 6,626$ \$ for Scheme D. The lowest of these seven costs is 6,626 \$, therefore it is underlined in Table B.13. We can note that the value of C incurred by D are often underlined when the demand is not random, which shows that using both our two principles and information sharing incurs the lowest overall supply chain cost. However, this result is not so obvious when demand is random, in which case C also has good results, which shows that, in this case, information centralization is the reason for the reduction of costs. When information centralization is not allowed, B generally incurs the lowest individual and supply chain costs.

B.5.3 Comparison of the Customer Service Levels

The third metric is the sum of backorders, which is a metric for customer service levels. This sum has to be minimized, because when it is zero, clients have the products they wish, else they have to wait for their availability. This measure is also included in costs, but we separate it for now. In fact, backorders can be avoided by overstocking, which increases costs but reduces the sum of backorders. In particular, the following data are obtained with empty initial inventories. Therefore, costs are more important than the sum of backorders, but the way of pricing a backorder depends on manager's choice. In particular, a manager could choose that backorders cost nothing, while we choose they

	Scheme C	Scheme D
1 Step	5.8; 5.4; 7.9; 7.7; 7.1; 4.9	3.8; 3.8; 4.2; 4.2; 4.5; 4.2
2 Inversed Step	5; 4.7; 10.4; 10.9; 13.1; 12.3	3.8; 3.8; 4.2; 4.2; 4.5; 4.2
3 Dirac	3.4; 3.3; 4.4; 4.2; 4.4; 3	5.5; 5.5; 5.7; 5.7; 6.2; 5.9
4 Inversed Dirac	4.5; 4.6; 6.8; 7; 8.1; 5.8	5.5; 5.5; 5.7; 5.7; 6.2; 5.9
5 Increase	15.1; 15.5; 15.4; 16.3; 16.3; 15.2	14.9; 14.9; 15.4; 15.4; 16; 16.4
6 Decrease	14.7; 14.7; 16.6; 16.6; 20.6; 22	14.9; 14.9; 15.4; 15.4; 16; 16.4
7 Weak seasonality	1.1; 1.1; 0.5; 0.5; 0.9; 0.5	4.4; 4.4; 1.6; 1.6; 4.2; 1.7
8 Medium seasonality	4.8; 4.2; 5.6; 5.3; 4.7; 3.9	4.5; 4.5; 5.4; 5.4; 3.2; 1.4
9 Strong seasonality	7.9; 6.5; 11.8; 10.2; 12; 10.4	4.7; 4.7; 6.2; 6.2; 6.5; 5.4
10A Uniform random	7.2; 5.3; 12.6; 9.1; 10.1; 9.1	12.1; 12.1; 13.8; 13; 13.6; 10.7
10B Uniform random	6.9; 7.7; 9.1; 10.6; 11.2; 8.1	11.4; 10; 9.8; 9.3; 10.6; 6
10C Uniform random	6.5; 6.1; 7.8; 9.1; 9.4; 6.5	9.4; 10.6; 10.4; 10.4; 11.7; 7.5
10D Uniform random	4.9; 4.8; 9.4; 9.8; 13.5; 11.8	10.9; 10.2; 12; 9.6; 10.2; 8.5
10E Uniform random	7.9; 6.4; 10.4; 8.9; 9.4; 7.3	12.6; 11.8; 14.2; 12.5; 13; 9.4
10F Uniform random	6.1; 5.2; 7.3; 6.6; 7.1; 5.5	10; 10; 11.1; 11.1; 11.9; 11.4
10G Uniform random	7.9; 6.5; 11.6; 6.7; 6.7; 8.4	11.2; 12.4; 10.6; 12.9; 14.7; 11
10H Uniform random	6.4; 7.1; 7.7; 10.9; 10; 8.5	12; 11.6; 13.1; 10.9; 11.7; 9.8
10I Uniform random	5.4; 6.2; 8.1; 7.1; 7.3; 6.2	12.2; 11.7; 14.1; 12.3; 13; 12.6
10J Uniform random	5.6; 6.1; 6; 7.9; 8.8; 5.7	11; 10.8; 10.5; 11.8; 12.9; 7.2

Table B.9: Standard-deviation of orders, i.e., $\sigma_{Op^i_w+\Theta p^i_w}$ (extension of Table 7.2, 4/4).

cost twice the inventory holding price.

Next, backordered O and Θ (Ob and Θb) are both taken into account and summed on the fifty weeks of a simulation. This sum is presented in Tables B.14, B.15, B.16 and B.17 in the same format than the standard deviation of orders in Tables B.6, B.7, B.8 and B.9. For example, for Scheme A under the Step demand pattern in Table B.14, 603; 752; 698; 950; 1,418; 2,476 means that during the simulation, LumberRetailer has 603 delays for shipping products... and the Sawmill 2,476 delays. This latter number does not mean 2,476 orders were lately fulfilled. In fact, the sum of backorders increase of one unit if an order cannot be fulfilled the first week, the sum of backorders increase of two units if an order cannot be fulfilled the first and the second week, etc.

Again Tables B.14, B.15, B.16 and B.17 could be merged in one table, and data have to be read line by line. For example, under the Step demand pattern, the LumberRetailer has a sum of backorders $\sum_w Ob^1_w + \Theta b^1_w = 603$ with Scheme A, $\sum_w Ob^1_w + \Theta b^1_w = 468$ with Scheme A', $\sum_w Ob^1_w + \Theta b^1_w = 522$ with Scheme A'', $\sum_w Ob^1_w + \Theta b^1_w = 549$ with Scheme B, $\sum_w Ob^1_w + \Theta b^1_w = 1,867$ with Scheme B', $\sum_w Ob^1_w + \Theta b^1_w = 1,237$ with Scheme C and $\sum_w Ob^1_w + \Theta b^1_w = 348$ with Scheme D. In general, Scheme D has the lowest sum of backorders, i.e., the best customer service levels, and C is as good as D, but results are hard to generalize.

	Scheme A	Scheme A′
1 Step	2,048+2,401+6,716+6,184 +8,526+14,180=40,055 $	950+1,288+3,687+3,883 +24,285+147,548=181,641 $
2 Inversed Step	604+608+5,318+5,274 +8,018+17,212=37,034 $	615+620+2,744+2,755 +8,159+50,700=65,593 $
3 Dirac	373+370+963+974 +2,112+4,414=9,206 $	595+597+6,276+6,067 +30,148+182,627=226,310 $
4 Inversed Dirac	409+467+2,114+1,609 +2,942+6,539=14,080 $	546+552+5,364+5,169 +32,156+193,287=237,074 $
5 Increase	5,595+6,261+6,735+6,666 +9,715+16,709=51,681 $	2,193+2,651+1,331+1,779 +1,832+21,285=31,071 $
6 Decrease	2,241+2,257+2,204+2,204 +4,433+11,683=25,022 $	447+450+463+467 +3,080+19,204=24,111 $
7 Weak seasonality	43+47+52+74 +102+360=678 $	130+131+6,080+6,027 +34,607+197,492=244,467 $
8 Medium season,	271+367+744+1,506 +2,517+7,214=12,619 $	384+391+3,101+3,132 +18,175+115,309=140,492 $
9 Strong season,	597+575+2,972+1,964 +4,271+10,395=20,774 $	538+542+2,240+2,208 +13,484+85,906=104,918 $
10A Uniform random	577+316+3,459+1,290 +3,083+10,800=19,525 $	1,087+639+32,977+19,613 +89,427+522,438=666,181 $
10B Uniform random	710+1,056+3,353+4,962 +5,934+11,944=27,959 $	625+547+25,417+18,756 +91,070+485,764=622,179 $
10C Uniform random	1,435+632+4,515+2,743 +4,887+12,290=26,502 $	967+581+13,941+25,554 +121,170+589,506=751,719 $
10D Uniform random	836+505+5,244+4,831 +7,898+20,786=40,100 $	496+920+23,065+18,792 +82,375+493,257=618,905 $
10E Uniform random	452+768+861+3,976 +4,212+13,647=23,916 $	1,237+716+26,488+24,341 +107,933+560,033=720,748 $
10F Uniform random	674+767+3,537+2,620 +4,637+8,711=20,946 $	617+670+22,995+22,951 +105,647+574,952=727,832 $
10G Uniform random	430+906+2,654+2,971 +4,702+10,722=22,385 $	892+973+23,803+27,143 +117,002+620,073=789,886 $
10H Uniform random	1,058+1,177+2,425+8,330 +7,115+19,982=40,087 $	1,121+1,088+25,297+25,598 +124,447+643,499=821,050 $
10I Uniform random	650+1,275+4,000+3,910 +6,198+11,218=27,251 $	714+1,095+30,739+23,237 +115,290+653,941=825,016 $
10J Uniform random	610+501+1,924+2,541 +4,131+10,496=20,203 $	610+603+23,303+19,819 +101,265+556,209=701,809 $

Table B.10: Supply chain costs C_{improved} (detail of Table 7.3, 1/4).

B.5.4 Comparison of the Inventory Level Variations

The last metric is the standard deviation of inventories when backordered are measured as negative inventory levels. This metric is used to choose the target inventory level. In fact, when the standard deviation of inventories increases, the target inventory level has to increase to avoid stockouts/backorders. Therefore, when this measure decreases, inventory levels also decreases, which reduces companies' costs without increasing their sum of backorders, i.e, without reducing the level of customer services.

Data in Tables B.18, B.19, B.20 and B.21 have the same format as usually, and represent the same experiments (in particular, initial inventories are empty). For ex-

	Scheme A″
1 Step	9,826+15,544+54,587+92,350 +256,175+724,130=1,152,612 $
2 Inversed Step	2,733+3,324+32,124+40,052 +130,279+367,581=576,093 $
3 Dirac	1,725+1,667+6,974+12,125 +44,937+125,497=192,925 $
4 Inversed Dirac	12,490+17,709+73,953+108,319 +315,815+878,362=1,406,648 $
5 Increase	9,700+13,770+40,584+92,410 +248,327+725,541=1,130,332 $
6 Decrease	29,010+38,524+185,796+263,649 +785,848+2239,739=3,542,566 $
7 Weak seasonality	3,868+3,025+13,815+15,731 +39,353+107,174=182,966 $
8 Medium seasonality	1,260+1,495+3,698+3,263 +18,158+49,029=76,903 $
9 Strong seasonality	1,536+1,658+7,180+10,199 +50,809+141,923=213,305 $
10A Uniform random	10,979+14,090+69,758+86,357 +245,792+680,554=1,107,530 $
10B Uniform random	11,549+14,745+69,592+96,380 +284,016+777,811=1,254,093 $
10C Uniform random	4,719+2,166+35,485+6,720 +13,944+95,361=158,395 $
10D Uniform random	8,332+6,858+57,637+50,517 +149,477+368,053=640,874 $
10E Uniform random	3,482+2,081+26,223+7,259 +29,063+76,913=145,021 $
10F Uniform random	2497+2479+10,148+20,082 +64,881+180,851=280,938 $
10G Uniform random	5,291+5,702+21,517+45,037 +155,594+446,370=679,511 $
10H Uniform random	2,798+9,207+20,086+54,819 +177,438+555,669=820,017 $
10I Uniform random	11,057+18,478+45,104+90,048 +248,705+751,017=1,164,409 $
10J Uniform random	6,532+12,154+29,710+107,759 +355,061+1,057,264=1,568,480 $

Table B.11: Supply chain costs C_{improved} (detail of Table 7.3, 2/4).

ample, under the Step demand with Scheme A in Table B.18, 33.7; 38.3; 140.2; 119.6; 180.2; 267.8 means the LumberRetailer's standard deviation of inventory is 33.7... and the Sawmill's standard deviation of inventory is 267.8. The best possible value is zero, i.e., inventory always steady, but this is only possible to get this best value when the whole demand is prefectly known in the future, which is not possible in practice because there is always a forecasting error.

Again, Tables B.18, B.19, B.20 and B.21 could be merged and data have to be compared line by line. For example, under the Step demand, the LumberRetailer's standard deviation of inventory is 33.7 with Scheme A, 19.7 with A′, 187.7 with A″, 20.5 with B, 28.7 with B′, 14.6 with C and 13.4 with D. In general, Schemes B, C and D have the best results.

	Scheme B	Scheme B′
1 Step	1,114+1,871+985+1,760 +1,457+1,848=9,035 $	3,674+4,120+4,369+5,670 +4,800+6,923=29,556 $
2 Inversed Step	466+109+651+274 +489+1,248=3,237 $	1,574+704+2,058+1,442 +2,262+9,397=17,437 $
3 Dirac	152+178+151+177 +175+486=1,319 $	664+684+1,074+1,261 +1,233+1,882=6,798 $
4 Inversed Dirac	276+301+276+300 +300+486=1,939 $	2,415+2,128+3,497+3,479 +3,538+6,316=21,373 $
5 Increase	5,967+10,093+5,230+9,710 +7,906+7,848=46,754 $	11,257+14,847+13,458+20,501 +17,897+17,348=95,308 $
6 Decrease	447+450+1,108+1,116 +1,709+3,924=8,754 $	2,241+2,257+2,133+2,176 +1,959+7,122=17,888 $
7 Weak seasonality	288+378+234+315 +273+675=2,163 $	240+271+342+349 +336+871=2,409 $
8 Medium seasonality	1,085+1,096+1,151+1,118 +1,070+2,418=7,938 $	1,081+586+1,339+778 +906+2,565=7,255 $
9 Strong seasonality	1,638+1,599+1,388+1,700 +2,066+3,579=11,970 $	1,416+1,610+1,809+2,618 +2,969+4,320=14,742 $
10A Uniform random	787+1,663+823+1,740 +1,528+2,693=9,234 $	1,392+1,191+2,201+1,574 +1,457+5,115=12,930 $
10B Uniform random	1,899+1,792+1,986+1,868 +1,947+2,648=12,140 $	2,093+1,784+2,253+2,887 +3,098+5,814=17,929 $
10C Uniform random	2,991+1,575+1,965+1,540 +1,489+3,208=12,768 $	3,133+2,137+4,451+3,497 +4,016+8,785=26,019 $
10D Uniform random	1,934+1,134+1,948+1,224 +1,248+2,687=10,175 $	2,683+1,291+3,518+2,153 +2,944+8,276=20,865 $
10E Uniform random	1,604+1,976+1,376+2,049 +2,203+2,936=12,144 $	1,111+1,586+978+1,805 +2,473+5,984=13,937 $
10F Uniform random	873+1,198+991+1,265 +1,201+2,548=8,076 $	1,231+1,470+1,658+2,163 +2,377+3,868=12,767 $
10G Uniform random	1,092+2,203+1,118+1,783 +1,610+2,871=10,677 $	374+1,734+988+2,265 +1,909+2,910=10,180 $
10H Uniform random	2,910+1,046+2,672+1,169 +1,169+3,043=12,009 $	3,075+2,095+3,357+3,189 +3,669+9,164=24,549 $
10I Uniform random	988+2,737+793+2,560 +2,586+3,057=12,721 $	1,610+2,650+1,462+3,401 +3,616+6,801=19,540 $
10J Uniform random	1,759+1,111+2,087+1,108 +1,224+2,729=10,018 $	1,331+1,256+2,134+1,873 +2,133+4,300=13,027 $

Table B.12: Supply chain costs C_{improved} (detail of Table 7.3, 3/4).

Notice that standard deviation of inventories is greater for upstream suppliers than for retailers for all ordering schemes. This is induced by the bullwhip effect. For schemes B and D, this hides the fact that inventory level fluctuates for a shorter period by upstream suppliers than by retailers. In fact, upstream suppliers' standard deviation is bigger because their fluctuation is greater, which is not true for the other ordering schemes.

Finally, all these tables show that Scheme D is often the best choice for all companies.

	Scheme C	Scheme D
1 Step	2,510+3,326+2,718+4,322 +4,193+4,764=21,833 $	706+1,460+580+1,352 +1,136+1,392=6,626 $
2 Inversed Step	627+632+1,385+1,576 +1,364+2,842=8,426 $	466+77+542+132 +180+1,020=2,417 $
3 Dirac	347+374+499+540 +487+683=2,930 $	120+166+132+164 +191+216=989 $
4 Inversed Dirac	404+452+572+669 +693+1,097=3,887 $	221+268+210+256 +198+216=1,369 $
5 Increase	9,196+12,812+9,396+16,533 +16,029+16,804=80,770 $	3,586+7,819+2,750+7,317 +6,008+5,140=32,620 $
6 Decrease	2,241+2,257+2,164+2,180 +2,526+6,068=17,436 $	447+450+750+822 +1,126+2,612=6,207 $
7 Weak seasonality	25+27+46+56 +43+82=279 $	213+249+128+193 +146+265=1,194 $
8 Medium seasonality	421+334+572+550 +473+887=3,237 $	497+564+611+636 +624+767=3,699 $
9 Strong seasonality	821+814+1,371+1,309 +1,724+2,992=9,031 $	997+1,125+1,129+1,255 +1,261+1,760=7,527 $
10A Uniform random	436+367+885+721 +852+1,756=5,017 $	632+980+697+1,021 +918+1,313=5,561 $
10B Uniform random	635+525+1,014+910 +998+1,629=5,711 $	836+704+915+689 +509+1,031=4,684 $
10C Uniform random	1,396+603+1,540+1,068 +1,374+2,845=8,826 $	1,359+771+1,307+828 +623+1,641=6,529 $
10D Uniform random	312+347+816+797 +1,195+2,296=5,763 $	1,263+716+1,263+736 +590+1,879=6,447 $
10E Uniform random	666+389+790+608 +872+1,692=5,017 $	909+779+972+852 +742+1,320=5,574 $
10F Uniform random	597+694+848+1,155 +1,150+1,448=5,892 $	609+698+648+752 +648+885=4,240 $
10G Uniform random	366+1,205+910+1,515 +1,271+1,599=6,866 $	728+1,351+705+1,320 +1,270+2,018=7,392 $
10H Uniform random	992+483+1,088+1,629 +1,041+2,740=7,973 $	2,493+799+1,811+938 +630+2,615=9,286 $
10I Uniform random	406+1,221+714+1,760 +1,881+1,989=7,971 $	595+1,396+585+1,394 +1,314+1,900=7,184 $
10J Uniform random	743+660+1,189+884 +728+2,280=6,484 $	1,058+757+1,146+776 +601+1,573=5,911 $

Table B.13: Supply chain costs C_{improved} (detail of Table 7.3, 4/4).

	Scheme A	Scheme A'
1 Step	603; 752; 698; 950; 1,418; 2,476	468; 630; 408; 570; 905; 2,962
2 Inversed step	0; 0; 23; 23; 157; 536	0; 0; 0; 0; 65; 736
3 Dirac	144; 148; 138; 174; 246; 366	72; 84; 163; 201; 1,021; 3,395
4 Inversed Dirac	155; 162; 152; 223; 300; 545	25; 36; 82; 113; 967; 3,725
5 Increase	2,707; 3,011; 1,953; 2,510; 3,259; 4,727	1,081; 1,297; 661; 877; 417; 594
6 Decrease	0; 0; 0; 0; 48; 298	0; 0; 0; 0; 0; 7
7 Weak seasonality	18; 22; 20; 33; 43; 43	46; 48; 12; 18; 203; 671
8 Medium seasonality	112; 140; 107; 147; 247; 514	140; 143; 42; 47; 107; 430
9 Strong seasonality	167; 167; 181; 236; 255; 665	201; 201; 91; 91; 189; 508
10A Uniform random	46; 75; 94; 118; 221; 317	15; 88; 14; 30; 233; 1,189
10B Uniform random	98; 171; 165; 303; 447; 1,162	72; 111; 76; 87; 247; 880
10C Uniform random	447; 161; 553; 249; 388; 1,111	332; 118; 255; 106; 473; 1,319
10D Uniform random	57; 24; 133; 60; 118; 650	43; 4; 0; 0; 605; 2,151
10E Uniform random	210; 151; 140; 253; 623; 2,205	81; 107; 106; 108; 714; 2,085
10F Uniform random	208; 206; 306; 294; 437; 760	153; 182; 97; 137; 779; 2,616
10G Uniform random	15; 348; 79; 367; 549; 798	3; 216; 9; 242; 1,323; 2,745
10H Uniform random	355; 57; 369; 151; 352; 1,764	183; 5; 146; 3; 772; 2,144
10I Uniform random	78; 363; 114; 514; 807; 1,137	51; 299; 14; 334; 1,638; 3,706
10J Uniform random	136; 177; 184; 185; 361; 708	51; 86; 26; 23; 217; 1,487

Table B.14: Sum of backorders, i.e., $\sum_w (Ob^i_w + \Theta b^i_w)$ (extension of Table 7.4, 1/4).

	Scheme A''
1 Step	522; 708; 1,758; 3,057; 8,943; 25,304
2 Inversed step	403; 295; 1,199; 1,622; 5,755; 15,681
3 Dirac	332; 259; 306; 456; 1,461; 4,000
4 Inversed Dirac	671; 880; 2,682; 4,200; 12,942; 35,277
5 Increase	1250; 1595; 1687; 3,390; 9,617; 27,617
6 Decrease	1431; 1766; 6,448; 9,807; 32,178; 87,360
7 Weak seasonality	309; 286; 838; 1,119; 3,129; 8,739
8 Medium seasonality	279; 360; 324; 521; 608; 1,629
9 Strong seasonality	309; 217; 353; 437; 1,643; 4,561
10A Uniform random	555; 676; 2,336; 3,025; 9,257; 24,653
10B Uniform random	563; 695; 2,337; 3,403; 10,616; 28,671
10C Uniform random	469; 496; 1,076; 787; 531; 3,055
10D Uniform random	460; 422; 1,979; 1,830; 5,433; 13,853
10E Uniform random	525; 490; 1,261; 2,180; 6,127; 16,941
10F Uniform random	434; 330; 404; 678; 1,996; 5,538
10G Uniform random	527; 493; 747; 1,464; 4,999; 14,453
10H Uniform random	372; 498; 677; 1,946; 6,451; 20,681
10I Uniform random	537; 707; 2,145; 3,375; 10,221; 27,849
10J Uniform random	418; 627; 1,128; 3,826; 13,596; 39,577

Table B.15: Sum of backorders, i.e., $\sum_w (Ob^i_w + \Theta b^i_w)$ (extension of Table 7.4, 2/4).

	Scheme B	Scheme B′
1 Step	549; 915; 249; 387; 135; 84	1,867; 1,937; 872; 1,125; 747; 1,726
2 Inversed step	192; 16; 204; 16; 16; 293	486; 65; 567; 189; 215; 2,879
3 Dirac	75; 87; 51; 63; 39; 42	322; 319; 202; 238; 186; 518
4 Inversed Dirac	124; 135; 113; 124; 113; 264	905; 771; 865; 679; 473; 1,694
5 Increase	2,940; 4,936; 352; 1,006; 126; 40	5,546; 7,261; 1,535; 3,430; 664; 407
6 Decrease	0; 0; 0; 0; 0; 0	0; 0; 0; 0; 70; 615
7 Weak seasonality	140; 179; 120; 173; 143; 69	103; 118; 117; 144; 154; 453
8 Medium seasonality	535; 536; 340; 372; 374; 300	533; 248; 572; 319; 333; 791
9 Strong seasonality	807; 782; 453; 615; 422; 321	627; 716; 553; 671; 556; 737
10A Uniform random	338; 802; 192; 629; 366; 572	482; 526; 325; 594; 581; 2,191
10B Uniform random	949; 867; 746; 715; 604; 224	929; 785; 522; 672; 542; 2,020
10C Uniform random	1,474; 764; 686; 543; 343; 513	1,424; 849; 719; 695; 511; 3,655
10D Uniform random	925; 518; 818; 337; 360; 846	1,301; 418; 745; 557; 593; 1,768
10E Uniform random	789; 967; 457; 597; 393; 612	504; 745; 424; 629; 505; 1,782
10F Uniform random	428; 584; 245; 368; 321; 553	557; 642; 534; 431; 259; 1,062
10G Uniform random	494; 1072; 356; 625; 600; 940	85; 863; 154; 567; 551; 726
10H Uniform random	1,434; 467; 1,025; 412; 367; 1,003	1,459; 836; 1369; 752; 1,033; 2,396
10I Uniform random	493; 1,387; 210; 958; 922; 611	725; 1313; 399; 1,072; 966; 2,570
10J Uniform random	865; 529; 780; 296; 209; 706	681; 553; 480; 355; 229; 941

Table B.16: Sum of backorders, i.e., $\sum_w (Ob^i_w + \Theta b^i_w)$ (extension of Table 7.4, 3/4).

	Scheme C	Scheme D
1 Step	1,237; 1,627; 1,343; 2,129; 2,081; 2,342	348; 714; 84; 198; 84; 36
2 Inversed step	0; 0; 40; 42; 157; 329	192; 0; 204; 0; 0; 216
3 Dirac	153; 165; 216; 233; 206; 274	59; 81; 35; 57; 45; 127
4 Inversed Dirac	167; 184; 223; 259; 233; 384	97; 119; 86; 108; 91; 142
5 Increase	4,531; 6,266; 4,664; 8,146; 7,956; 8,402	1767; 3824; 96; 382; 96; 22
6 Decrease	0; 0; 0; 0; 0; 77	0; 0; 0; 0; 0; 91
7 Weak seasonality	12; 13; 19; 24; 22; 33	100; 122; 61; 86; 48; 63
8 Medium seasonality	178; 147; 247; 237; 206; 345	245; 276; 87; 122; 81; 30
9 Strong seasonality	308; 351; 420; 509; 651; 1295	482; 550; 182; 275; 164; 85
10A Uniform random	48; 127; 114; 254; 297; 445	272; 471; 111; 251; 262; 342
10B Uniform random	237; 194; 381; 334; 345; 536	397; 330; 161; 175; 115; 150
10C Uniform random	670; 266; 749; 396; 632; 1,308	666; 364; 284; 136; 158; 279
10D Uniform random	51; 5; 135; 74; 179; 433	605; 313; 386; 147; 104; 375
10E Uniform random	306; 136; 322; 242; 374; 700	444; 373; 210; 149; 175; 366
10F Uniform random	247; 299; 338; 482; 474; 573	288; 332; 173; 167; 132; 265
10G Uniform random	64; 585; 175; 745; 630; 434	310; 659; 115; 375; 369; 396
10H Uniform random	469; 33; 514; 126; 393; 1,122	1256; 356; 510; 120; 88; 515
10I Uniform random	122; 589; 181; 866; 933; 752	242; 682; 126; 343; 287; 331
10J Uniform random	350; 293; 584; 391; 300; 874	519; 356; 238; 173; 104; 225

Table B.17: Sum of backorders, i.e., $\sum_w (Ob^i_w + \Theta b^i_w)$ (extension of Table 7.4, 4/4).

	Scheme A	Scheme A'
1 Step	33.7; 38.3; 140.2; 119.6; 180.2; 267.8	19.7; 24.2; 51.2; 56.8; 324.7; 1,869.7
2 Inversed step	6.2; 6.2; 88.6; 85.9; 153.6; 308.3	4.1; 4.1; 24.2; 24.2; 108.5; 663.6
3 Dirac	4.6; 4.4; 18.7; 18.9; 42.6; 74	8.4; 8.7; 79.9; 83.4; 405.7; 2,330.8
4 Inversed Dirac	6; 7.1; 39.3; 28.1; 54.1; 110.2	6.2; 6.8; 70.2; 73.2; 441.6; 2,551.7
5 Increase	51; 59.4; 141.8; 107.6; 185.2; 288.1	14.9; 22.9; 13.3; 21.5; 30.8; 275
6 Decrease	29.1; 29.1; 32.5; 31.3; 97.5; 233.6	3.1; 3.1; 4.3; 4.3; 30.2; 234.7
7 Weak seasonality	0.7; 0.7; 1.1; 1.2; 2; 6.2	2.5; 2.4; 100; 102.1; 608.4; 3,618
8 Medium seasonality	3.9; 5.9; 16.7; 40.3; 69.9; 152.4	5.9; 5.9; 52.5; 52.7; 321.6; 2,104
9 Strong seasonality	11.1; 10.2; 66.1; 44.3; 98; 226.1	7.7; 7.7; 39.5; 40; 238.8; 1,524.4
10A Uniform random	9.4; 6.5; 63.9; 27.2; 72.4; 224.2	12.6; 13.3; 451.6; 363.9; 1,714; 9,793.8
10B Uniform random	12.2; 17.8; 71.3; 97.7; 114.2; 215.6	11.6; 10; 382.6; 339.9; 1,671; 9,337.8
10C Uniform random	22.6; 11.3; 92; 62.3; 110.2; 248.3	15.6; 10.1; 222.1; 378; 1,979.7; 10,145.8
10D Uniform random	12; 8; 98.9; 81.3; 137; 419	8.5; 12.3; 344.9; 285.4; 1,303.9; 8,131.2
10E Uniform random	4.5; 15.5; 17.2; 100.5; 96; 211	19.7; 14.2; 432.3; 415.4; 1,894.2; 10,039.2
10F Uniform random	10.9; 12.8; 71.1; 55.8; 101.6; 180	10.9; 11.4; 359.2; 366.2; 1,672.7; 9127.8
10G Uniform random	7; 12.5; 50.5; 62.3; 107.6; 220	11.1; 15.5; 378.2; 392.2; 1,771.9; 10,016.5
10H Uniform random	16.2; 15.8; 45.2; 145.5; 124.7; 312.9	18.1; 12.8; 429.4; 390.3; 2,045.9; 11,264.4
10I Uniform random	11.7; 20.3; 86.1; 75.3; 134.1; 195.5	11.3; 17.9; 506.6; 368.7; 1,787.3; 10,731.5
10J Uniform random	11.4; 9; 41.9; 48.1; 99; 223.3	10.2; 11.1; 354; 366.8; 1,902.5; 10,438.5

Table B.18: Standard-deviation of inventories, i.e., $\sigma_{I+Op+\Theta p}$ (extension of Table 7.5, 1/4).

	Scheme A''
1 Step	187.7; 270.1; 831; 1,542.2; 4,495.5; 1,2315.9
2 Inversed step	61.5; 76.5; 606.9; 818.1; 2,766.7; 7,658.4
3 Dirac	35.5; 35.3; 128; 198.6; 717.4; 1,954.3
4 Inversed Dirac	236.8; 344.5; 1,318.8; 2,101.4; 6,370; 17,247.5
5 Increase	204.4; 310.5; 724.8; 1,678.5; 4,764.8; 13,438.6
6 Decrease	520; 727.7; 3,269.2; 5,034.4; 15,538.9; 43,311.6
7 Weak seasonality	89.4; 77.2; 378.9; 465.4; 1,235.8; 3,242.8
8 Medium seasonality	26.8; 32.7; 78; 82.5; 289; 756.1
9 Strong seasonality	33.8; 38.9; 136.4; 178.7; 840.3; 2,299.5
10A Uniform random	195.6; 253.8; 1,138.7; 1,529.4; 4,524.3; 12,269.2
10B Uniform random	205.6; 266.8; 1,153.2; 1,729.8; 5,310.2; 14,176.9
10C Uniform random	94.4; 43.3; 492.6; 141.2; 194; 1,317.9
10D Uniform random	157.5; 137.1; 950.5; 878.1; 2,713.4; 6,574.3
10E Uniform random	71.3; 45.2; 393.2; 147.1; 436.1; 1,169.1
10F Uniform random	50.2; 51.7; 168.6; 304.3; 994.8; 2690.2
10G Uniform random	111.6; 123.1; 300.4; 688; 2,478.1; 6,879.7
10H Uniform random	55.7; 167.8; 296.8; 909.4; 3,346.4; 10,052.2
10I Uniform random	190.8; 317.8; 682.9; 1,505.3; 4,318.6; 12,597.5
10J Uniform random	135.5; 227.1; 493.7; 1,966.4; 6,753.4; 19,373.8

Table B.19: Standard-deviation of inventories, i.e., $\sigma_{I+Op+\Theta p}$ (extension of Table 7.5, 2/4).

	Scheme B	Scheme B′
1Step	20.5; 23.2; 20.6; 23.5; 22.8; 35.5	28.7; 38.2; 51.4; 71; 65.8; 140.3
2Inversed step	6.6; 5.2; 14.1; 12.5; 20.8; 35.5	28.7; 13.3; 41.1; 26.3; 42.2; 179.7
3Dirac	2.6; 2.7; 4.6; 4.7; 7.8; 22	7.1; 7.2; 13.2; 15.4; 19.5; 27.3
4Inversed Dirac	7.9; 8; 9; 9.1; 11.1; 22	43.1; 34.9; 74.4; 61.8; 69.2; 117.6
5Increase	29.9; 55.7; 25.6; 51.2; 42.2; 39.4	74.4; 103.3; 91; 147.2; 131.3; 123.4
6Decrease	3.1; 3.1; 9; 9; 16; 39.4	29.1; 29.1; 33.3; 30.3; 42.8; 163.5
7Weak seasonality	4.2; 5.5; 2.7; 3.5; 3.1; 2.7	4.5; 4.8; 7.6; 6.7; 6.4; 21.8
8Medium seasonality	8.8; 8.8; 11.4; 10.5; 13.8; 39.1	17.5; 8.9; 31.2; 12.5; 13.8; 49.6
9Strong seasonality	10.8; 10.1; 14.3; 13; 27.2; 57.7	19; 22.2; 30; 39.3; 44.4; 61.9
10A Uniform random	9.3; 13.4; 14.1; 15.8; 21.8; 42.5	22.9; 20.5; 38.6; 34.1; 31.6; 101.7
10B Uniform random	13.5; 14.4; 18.1; 16.5; 22.4; 41.2	26.1; 24.1; 45.2; 43.5; 57.1; 121.7
10C Uniform random	18.3; 13.5; 16.8; 16; 20.9; 53.2	40.9; 37.8; 97.8; 72.8; 95.5; 200.7
10D Uniform random	17.2; 12.1; 19.6; 17; 23.9; 37.6	36.3; 23; 73.7; 40.7; 63; 158.9
10E Uniform random	12.2; 16.4; 18.4; 17; 23.2; 49.1	14.6; 25.5; 18.9; 30.4; 43.9; 155.3
10F Uniform random	6.3; 7.2; 12.1; 12.6; 17.4; 40.6	13.9; 18.5; 25.6; 35.3; 51.6; 66.1
10G Uniform random	12.7; 15.1; 16.8; 15.3; 22.5; 42.2	7.2; 15.8; 17.7; 26.9; 25.6; 46.1
10H Uniform random	16.9; 12.2; 17.7; 18.3; 23.1; 45.3	47.1; 33.7; 80.4; 64.3; 69.6; 211.1
10I Uniform random	10.2; 15.2; 14.1; 17; 22.3; 47.6	19.6; 28.3; 27; 44.8; 50.2; 137.7
10J Uniform random	14.7; 10.8; 20.7; 15; 20.9; 44	16.3; 15.7; 31.7; 29.1; 42.2; 65.7

Table B.20: Standard-deviation of inventories, i.e., $\sigma_{I+Op+\Theta p}$ (extension of Table 7.5, 3/4).

	Scheme C	Scheme D
1 Step	14.6; 16.9; 20.5; 25.2; 23.9; 27.2	13.4; 16; 13.1; 16.2; 14.4; 18.3
2 Inversed step	5.6; 5.6; 20.2; 23.3; 24.1; 47.3	6.6; 4.8; 9.8; 8.1; 9.4; 18.3
3 Dirac	3.9; 4.1; 6.3; 6.6; 6.9; 9	2.6; 3; 5; 4.9; 6; 8.4
4 Inversed Dirac	5.3; 5.7; 8.5; 9.8; 12.5; 15.6	5.4; 6.1; 6.6; 7.2; 6; 8.4
5 Increase	59.2; 86.6; 60.6; 115.2; 114.3; 116.3	17.6; 41; 14.4; 37.1; 30.5; 22.4
6 Decrease	29.1; 29.1; 30.6; 30.6; 50.6; 125.7	3.1; 3.1; 6.5; 6.5; 9.6; 22.4
7 Weak seasonality	0.4; 0.4; 0.8; 0.8; 0.5; 1.1	2.8; 3.5; 1.8; 2.4; 1.4; 4.3
8 Medium seasonality	5.6; 4.1; 8.3; 7.5; 6.7; 13.1	3.9; 3.5; 8.3; 7.4; 8.9; 11.6
9 Strong seasonality	12.3; 9.9; 25.3; 20.4; 27.8; 44.1	7.8; 6.9; 14.2; 12.9; 17.4; 27.8
10A Uniform random	7.2; 6.1; 17.5; 11.8; 15.2; 29.8	7.5; 8.4; 11.7; 11.9; 12.9; 16.5
10B Uniform random	9.2; 7.9; 15; 14.1; 16.3; 26.2	6.9; 6.3; 11.6; 9.6; 8.3; 13.1
10C Uniform random	11; 7.5; 15.6; 16.3; 17.4; 22.7	8.1; 6.7; 12.3; 10.9; 9.7; 15.6
10D Uniform random	5.4; 4.8; 14.1; 14; 23.2; 34.8	9.3; 8.1; 11.9; 11.9; 10.7; 15
10E Uniform random	7.3; 6.4; 12.1; 9.3; 11.2; 17.9	6.7; 6.4; 12; 10.1; 10.4; 14.3
10F Uniform random	7.3; 7.9; 11.4; 14.2; 15.1; 17.3	5.2; 5; 9.7; 9.5; 10.2; 14.5
10G Uniform random	6.9; 7.5; 17.1; 7.9; 8; 25.1	8.7; 7.9; 12.3; 10.3; 11.2; 15.3
10H Uniform random	8.3; 7.4; 10.8; 24.2; 16.5; 25.2	10; 8.1; 11.9; 13.6; 11.1; 12.3
10I Uniform random	6.6; 7.8; 11.9; 9.9; 10.6; 16.3	7.7; 7; 10.5; 10.1; 11.3; 13.9
10J Uniform random	7.5; 7.5; 10.1; 11.3; 11; 15.4	7.7; 6.6; 12.3; 10.8; 10.4; 14.9

Table B.21: Standard-deviation of inventories and backorders, i.e., $\sigma_{I+Ob+\Theta b}$ (extension of Table 7.5, 4/4).

Appendix C

Detail about Simulations with a Heterogeneous Supply Chain

Since simulation outputs depend on the parameters and on the initial conditions, we optimize them in this appendix for the case of the heterogenous supply chain (cf. second series of experiments in Chapter 8). We only consider Scheme A″, B and D in Chapter 8, but Scheme A″ has been optimized in Section B.2. As a consequence, we only present now the optimization of the parameters of Schemes B and D. We discuss the results of these optimizations in Section C.3. Finally, we present the problem that the Sawmill faces when it places orders based on its two (lumber and paper) requirements in Section C.4.

C.1 Optimization of Scheme B

Like in Section B.3 for homogeneous supply chains, the first parameter of Scheme B is λ, which represents the quantity of Θ to send to eventually stabilize the inventory on its initial level when the market consumption has been steady for a sufficient period. This parameter only depends on the ordering and shipping delays between the considered company and its direct supplier. The second parameter, I_1^i (company i's inventory in the first week) represents the initial inventory level. This parameter depends on the structure of the supply chain and on the cost function. We have first to seek the good value of λ because it rules the process of inventory stabilization, and next the good value of the initial inventory level because the incurred cost depend on this level, which is also where inventory stabilizes.

C.1.1 Value of λ

We have already explained how to choose λ in Section B.3. Therefore, we take its value in Table B.4. In particular, we take $\lambda = 4$ in our simulation, because there are two weeks of ordering and shipping delays, like in the first series of experiments.

C.1.2 Initial inventory levels

Since inventory levels fluctuate during a longer period by the retailer than by its suppliers and this fluctuation is more important by the suppliers than by the retailer, the optimal initial inventory level is not the same for all companies. Since the ordering scheme B is collaborative, we define the optimal initial inventories as the inventory levels incurring the lowest cost C for the whole supply chain, which may be different from the optimal initial inventory for each company under the assumption that their supplier has never backorders (this second assumption was made to optimize A" in Section B.2). This optimum is experimentally obtained on our simulation with the Solver in Microsoft Excel. We have studied three different ways of optimizing:

1. all initial inventory levels may be different, even lumber and paper Sawmill initial inventories (Table C.1). The format of data is $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1+C^2+C^3+C^4+C^5+C^6 = C$. For example, when ordering and shipping delays are both equal to two weeks, [0, 0, 30, 0, 0, 0, 78] means that the supply chain cost C is minimum when the two retailers, the PaperWholesaler and the PaperMill have empty initial inventory, the LumberWholesaler has 30 items, the Sawmill no units of lumber and 78 of paper. This initial state of the supply chain leads to a cost $C = 1,944$ k$ for the supply chain, where, for example, the Sawmill incurs a cost of $C^6 = 480$ k$ for its two inventories.

2. all initial inventories may be different, except that lumber and paper Sawmill inventories are equal (Table C.2). This is the optimization method used next for our experiments. The format of data is the same as above: $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$. We can note that the constraint $I_1^{6-lumber} = I^{6-paper}$ is satisfied for all entries in Table C.2.

3. all companies have the same initial inventory level, in order to check if it worth to differentiate companies. Table C.3 show the results. The format of data is the same as above: $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$. We can note that the constraint $I_1^1 = I_1^2 = I_1^3 = I_1^4 = I_1^5 = I_1^{6-lumber} = I^{6-paper}$ is satisfied for all entries in Table C.3.

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[0, 0, 0, 0, 2, 0, 1]→ 81+146+70+131 102+102=632 k$	[0, 0, 0, 0, 1, 0, 5]→ 165+328+144+298 228+189=1,352 k$	[0, 0, 0, 0, 0, 0, 84]→ 278+373+244+324 200+502=1,921 k$	[0, 0, 42, 36, 0, 0, 78]→ 171+284+569+732 387+591=2,734 k$
2 week ordering delay	[0, 0, 0, 0, 2, 0, 1]→ 183+307+162+277 211+172=1,312 k$	[0, 0, 30, 0, 0, 0, 78]→ 137+373+430+324 200+480=1,944 k$	[0, 0, 48, 36, 0, 0, 72]→ 143+284+595+739 373+569=2,703 k$	[0, 0, 54, 48, 30, 54, 6]→ 198+217+663+772 858+703=3,411 k$
3 week ordering delay	[0, 0, 36, 0, 0, 0, 51]→ 112+390+463+340 217+394=1,916 k$	[0, 0, 42, 37, 0, 9, 65]→ 145+276+521+756 356+611=2,665 k$	[0, 0, 54, 48, 30, 9, 48]→ 188+217+640+782 837+703=3,367 k$	[5, 1, 60, 57, 44, 30, 48]→ 193+242+663+845 1119+972=4,034 k$

Table C.1: Optimal initial inventory for ordering scheme B for different ordering and shipping delays and for the Step pattern (all initial inventories may be different).

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[0, 0, 0, 0, 1, 2, 2]→ 77+147+67+132 101+118=642 k$	[0, 0, 0, 2, 3, 3, 3]→ 158+310+137+311 228+231=1,375 k$	[0, 0, 0, 31, 8, 39, 39]→ 171+163+137+584 395+744=2,194 k$	[0, 0, 16, 36, 36, 32, 32]→ 220+158+339+597 840+753=2,907 k$
2 week ordering delay	[0, 0, 0, 0, 2, 1, 1]→ 180+307+159+277 211+181=1,315 k$	[0, 0, 0, 30, 0, 39, 39]→ 173+198+139+609 282+703=2,104 k$	[0, 0, 16, 36, 30, 36, 36] → 209+170+325+620 738+761=2,823 k$	[0, 0, 47, 47, 43, 36, 36]→ 141+184+548+722 1,009+873=3,477 k$
3 week ordering delay	[0, 0, 0, 9, 0, 42, 42]→ 166+333+132+421 237+700=1,989 k$	[0, 0, 35, 36, 30, 30, 30]→ 121+170+435+630 719+671=2,746 k$	[0, 0, 45, 48, 36, 39, 39]→ 143+196+525+759 908+869=3,400 k$	[5, 3, 54, 54, 50, 43, 43]→ 187+245+608+810 1,166+1,032=4,048 k$

Table C.2: Optimal initial inventory for ordering scheme B for different ordering and shipping delays and for the Step pattern (all initial inventories may be different, except by the Sawmill).

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[1, 1, 1, 1, 1, 1, 1]→ 84+154+78+131 +133+113=693 k$	[3, 3, 3, 3, 3, 3, 3]→ 163+319+161+320 +228+231=1,422 k$	[13, 13, 13, 13, 13, 13, 13]→ 224+386+290+510 +526+429=2,365 k$	[24, 24, 24, 24, 24, 24, 24]→ 280+392+402+643 +778+664=3,159 k$
2 week ordering delay	[2, 2, 2, 2, 2, 2, 2]→ 178+302+171+292 +223+186=1,352 k$	[13, 13, 13, 13, 13, 13, 13]→ 222+354+295+477 +494+408=2,250 k$	[22, 22, 22, 22, 22, 22, 22]→ 274+404+397+622 +730+616=3,043 k$	[30, 30, 30, 30, 30, 30, 30]→ 328+440+491+766 +966+818=3,809 k$
3 week ordering delay	[12, 12, 12, 12, 12, 12, 12]→ 229+337+301+446 +452+382=2,147 k$	[21, 21, 21, 21, 21, 21, 21]→ 271+405+398+593 +685+586=2,938 k$	[29, 29, 29, 29, 29, 29, 29]→ 324+445+487+746 +917+784=3,703 k$	[36, 36, 36, 36, 36, 36, 36]→ 371+487+578+890 +1,157+980=4,463 k$

Table C.3: Optimal initial inventory for ordering scheme B for different ordering and shipping delays and for the Step pattern (all initial inventories are identical).

We do not discuss Tables C.1, C.2 and C.3, but we can rapidly note that companies do not prefer the same optimization scenario: some companies prefer Table C.1, and others C.2 and C.3. Moreover, both costs and optimum levels of initial inventory increase with the length of delays.

C.2 Optimization of Scheme D

Scheme D is very similar to B and has therefore the same two parameters λ and I_1^i to be set up. These two setups have to be carried out in the same order: first λ, next I_1^i.

C.2.1 Value of λ

We have previously determined λ in Section B.4. We take its value in Table B.5. More precisely, since there are two weeks of ordering and shipping delays in our simulation, we take $\lambda = 2$ for all companies, except both retailers for which we take $\lambda = 4$.

C.2.2 Initial inventory levels

The difference between rules B and D is the use of information centralization which allows to improve the reactivity of the supply chain to changes on the market. Precisely, this improvement reduces inventory fluctuations, and therefore, the duration of backorders and of overstockings. Since inventories fluctuate less with Scheme B than with D, every company saves money if it chooses a lower initial inventory with Scheme D. The optimal initial inventory is obtained again with the Solver in Microsoft Excel for the same three different ways of optimizing as in Subsection B.3:

1. all initial inventory levels may be different, even by lumber and paper Sawmill (Table C.4). The format of data is again $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1+C^2+C^3+C^4+C^5+C^6 = C$. For example, when ordering and shipping delays both are two weeks, the following notation [0, 0, 0, 0, 12, 6, 24] means that the supply chain cost C is minimum when all Retailers and Wholesalers have empty initial inventory, the PaperMill has 12 units in inventory, and the Sawmill 6 units of lumber and 24 of paper in its two inventories. This initial state of the supply

chain leads to a cost $C = 1,183$ k\$ for the supply chain, where, for example, the Sawmill incurs a cost of $C^6 = 255$ k\$ for its two inventories.

2. all initial inventories may be different, except that lumber and paper Sawmill inventories are equal (Table C.5). The format of data is the same as above: $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$ with the constraint $I_1^{6-lumber} = I^{6-paper}$. We use this method of optimization in our experiments.

3. all companies have the same initial inventory level, in order to check if it is worth differentiating companies. Table C.6 shows the results. The format of data is the same as above: $[I_1^1, I_1^2, I_1^3, I_1^4, I_1^5, I_1^{6-lumber}, I^{6-paper}] \rightarrow C^1 + C^2 + C^3 + C^4 + C^5 + C^6 = C$ with the contraint $I_1^1 = I_1^2 = I_1^3 = I_1^4 = I_1^5 = I_1^{6-lumber} = I^{6-paper}$.

C.3 Discussion and Implications for our Simulations

Parameter λ is easier to choose than the initial inventory level, because λ only depends on the delays between each company and its direct supplier(s), while the initial inventory level depends on the demand, on the pricing function and on the dynamics/structure of the supply chain. For example, if backorders were considered as free, the optimal value of initial inventory level would be zero, because companies would only want to avoid holding products. During the optimization process of initial inventories, the Solver often converges on local optima. These local optima leads to very similar values of C, but the distribution of this cost among companies change. In other words, for some companies i, C^i is higher in some optimum than in others, while C remains quite constant for the whole supply chain. This may be a problem in our experiments, because we check individual companies' incentives for using an ordering scheme instead of another. In fact, two investigations could be done here:

- *First investigation:* The first possbile investigation is to find the global optimum with an exhaustive search. Since we only use integer values for initial inventories, and if we assume the optimal value of initial inventories cannot exceed one hundred items, we have to test one hundred values for each decision variable of this problem, where each initial inventory level is a decision variable. For example, the first optimization method assumes all inventories are independent. Therefore, there are seven decision variables in this method, and 100^7 values have to be tested for the whole supply chain. With the second optimization method, both Sawmill's inventories are equal. In this case, 100^6 values have to be tested. Finally, the

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[0, 0, 0, 0, 1, 0, 12]→ 51+72+41+58 +52+89=363 k$	[0, 0, 0, 0, 0, 0, 36]→ 120+175+100+145 +88+226=855 k$	[0, 0, 0, 0, 0, 0, 60]→ 219+326+185+277 +172+378=1,557 k$	[0, 0, 30, 35, 0, 0, 62]→ 181+228+457+672 +326+477=2,342 k$
2 week ordering delay	[0, 0, 0, 0, 0, 0, 0]→ 102+284+82+256 +210+106=1,039 k$	[0, 0, 0, 0, 12, 6, 24]→ 165+208+131+159 +265+255=1,183 k$	[0, 0, 24, 6, 24, 5, 36]→ 152+269+363+289 +484+367=1,924 k$	[0, 0, 30, 31, 29, 6, 43]→ 227+213+468+565 +642+496=2,611 k$
3 week ordering delay	[0, 0, 0, 0, 6, 9, 12]→ 127+203+93+155 +169+195=942 k$	[0, 0, 18, 11, 18, 12, 24]→ 115+187+262+288 +352+312=1,516 k$	[0, 0, 11, 13, 24, 25, 35]→ 210+291+253+384 +475+505=2,118 k$	[0, 0, 37, 36, 30, 29, 47]→ 169+247+459+633 +635+649=2,792 k$

Table C.4: Optimal initial inventory for ordering scheme D for different ordering and shipping delays and for the Step pattern (all initial inventories may be different).

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[0, 0, 0, 0, 2, 1, 1]→ 49+100+39+85 +65+86=424 k$	[0, 0, 0, 0, 6, 0, 0]→ 120+248+100+218 +172+179=1,037 k$	[0, 0, 0, 18, 18, 24, 24]→ 152+161+118+394 +442+500=1,766 k$	[0, 0, 0, 36, 24, 30, 30]→ 247+142+196+598 +629+653=2,465 k$
2 week ordering delay	[0, 0, 0, 0, 0, 12, 12]→ 66+147+46+117 +70+218=664 k$	[0, 0, 0, 0, 12, 24, 24]→ 109+208+75+159 +265+416=1,232 k$	[0, 0, 0, 12, 23, 30, 30]→ 191+238+141+349 +484+551=1,953 k$	[0, 0, 24, 37, 29, 35, 35]→ 167+178+359+616 +661+686=2,668 k$
3 week ordering delay	[0, 0, 0, 0, 6, 9, 9]→ 127+210+93+162 +172+189=953 k$	[0, 0, 5, 10, 18, 24, 24]→ 144+195+148+280 +352+411=1,530 k$	[0, 0, 4, 16, 24, 32, 32]→ 226+273+197+410 +480+549=2,135 k$	[0, 5, 30, 36, 30, 36, 36]→ 188+258+409+633 +661+671=2,820 k$

Table C.5: Optimal initial inventory for ordering scheme D for different ordering and shipping delays and for the Step pattern (all initial inventories may be different, except by the Sawmill).

	1 week shipping delay	2 week shipping delay	3 week shipping delay	4 week shipping delay
1 week ordering delay	[1, 1, 1, 1, 1, 1, 1]→ 54+109+48+86 +98+86=483 k$	[3, 3, 3, 3, 3, 3, 3]→ 121+248+118+250 +171+189=1,097 k$	[6, 6, 6, 6, 6, 6, 6]→ 198+408+209+439 +354+313=1,920 k$	[4, 4, 4, 4, 4, 4, 4]→ 248+363+346+570 +641+546=2,714 k$
2 week ordering delay	[2, 2, 2, 2, 2, 2, 2]→ 95+176+88+168 +121+110=759 k$	[8, 8, 8, 8, 8, 8, 8]→ 155+269+181+316 +292+239=1,452 k$	[16, 16, 16, 16, 16, 16, 16]→ 216+327+281+457 +483+410=2,173 k$	[22, 22, 22, 22, 22, 22, 22]→ 267+381+380+616 +685+569=2,898 k$
3 week ordering delay	[6, 6, 6, 6, 6, 6, 6]→ 132+201+141+219 +186+161=1,040 k$	[12, 12, 12, 12, 12, 12, 12]→ 184+279+229+361 +339+290=1,682 k$	[18, 18, 18, 18, 18, 18, 18]→ 237+347+314+500 +520+435=2,353 k$	[26, 26, 26, 26, 26, 26, 26]→ 293+396+417+640 +714+601=3,060 k$

Table C.6: Optimal initial inventory for ordering scheme D for different ordering and shipping delays and for the Step pattern (all initial inventories are identical).

third optimization method assumes all initial inventories are equal, therefore only 100 values have to be tested.

- *Second investigation:* We could next do the second investigation in which we would check if the results presented in Section 8.4 still hold with local optima near of the global optimum. This second investigation would give us an idea of the robustness of our results.

For the specific case of our second series of experiments, we let each company in our second series of experiments choose its initial inventory level and the Sawmill has the constraint that $I_1^{6-lumber} = I^{6-paper}$, which corresponds to the second optimization method in Subsections B.3.2 and B.4.2. In this context, we have to extend the entry "Ordering 2" and "Shipping 2" in Tables C.2 and C.5 for the eight other market consumption patterns, because Tables C.2 and C.5 only hold for the Step pattern. This extension is presented in Table C.7. Therefore, the initial inventory levels in Scheme B and D used in our second series of experiments in Section 8.4 have to be set up according to Table C.7. The format of data is the same as in Tables C.1 to C.6, except the addition of two numbers between parenthesis after C. The meaning of these numbers is clarified in the next subsection. Since we also use Scheme A" with empty initial inventory in our experiments, the costs it induces are also presented in Table C.7.

Finally, it is interesting to note that all costs in this dissertation are calculated under the assumption that a unit in inventory costs half a backordered unit. Since the only difference between "QWSG costs", our "improved costs" and our "realistic costs" is a multiplicative constant depending on the considered company, and since optimization is carried out to minimize the sum of individual company's costs, optimal initial inventories are different when considering these three methods of cost evaluation. The overall supply chain cost also changes with these three methods of cost calculation. But the Nash equilibria are exactly the same, because they are determined by comparing costs within a single company, and thus, the multiplication by a constant does not change the equilibria.

On the contrary, we can wonder what would happen if we changed the ratio of $1/2$ between inventory holding cost and backorder cost. In particular, if the pricing function considered that backorders are free, the optimal value of initial inventory level would be zero, because companies would only want to avoid to hold products in inventory. It could be interesting to change the relative weights given to inventory holding and backorders in order to check if results of the second series of experiments would be affected. In fact, this will change the optimal value of initial inventories, but results from the analyze of simulation outcomes may also change. In particular, Nash equilibria found in Section 8.4 may change.

Market cons. pattern	Ordering scheme A"	Ordering scheme B	Ordering scheme D
1. Step	[0, 0, 0, 0, 0, 0, 0] →2,658+5,988+14,764 +35,574+98,680+149,945 =307,609 k$ (2; 2)	[0, 0, 0, 30, 0, 39, 39] →173+198+139 +609+282+703 =2,104 k$ (2; 2)	[0, 0, 0, 0, 12, 24, 24] →109+208+75 +159+265+416 =1,232 k$ (2; 2)
2. Inversed step	[0, 0, 0, 0, 0, 0, 0] →1,061+1,853+12,471 +22,331+72,635+107,140 =217,491 k$ (2; 2)	[0, 0, 0, 0, 0, 6, 6] →30+48+93 +140+260+242 =813 k$ (2; 2)	[0, 0, 0, 0, 0, 6, 6] →30+43+51 +74+101+173 =470 k$ (2; 2)
3. Dirac	[0, 0, 0, 0, 0, 0, 0] →692+955+2,796 +6,945+25,737+38,319 =75,444 k$ (2; 2)	[0, 0, 0, 6, 0, 0, 0] →61+28+61 +166+100+119 =535 k$ (2; 2)	[0, 0, 0, 0, 0, 0, 0] →48+95+53+94+110+56 =456 k$ (2; 2)
4. Inversed Dirac	[0, 0, 0, 0, 0, 0, 0] →3,299+6,732 +19,531+41,176 +120,053+176,755 =367,546 k$ (2; 2)	[0, 0, 0, 0, 0, 0, 0] →73+114+73 +114+114+80 =568 k$ (3; 2)	[0, 0, 0, 0, 0, 0, 0] →58+102+55 +97+75+38 =427 k$ (2; 2)
5. Increase	[0, 0, 0, 0, 0, 0, 0] →1,327+2,722+5,553 +18,267+49,087+75,991 =152,947 k$ (2; 2)	[10, 10, 26, 26, 43, 69, 69]→8+17+28 +50+105+308 =516 k$ (2; 2)	[10, 9, 19, 19, 28, 58, 58]→8+32+19 +37+69+364 =529 k$ (3; 4)
6. Decrease	[0, 0, 0, 0, 0, 0, 0] →3,692+7,029 +23,648+48,107 +143,390+215,647 =441,513 k$ (2; 2)	[0, 0, 0, 0, 0, 0, 0] →57+82+141 +204+312+275 =1,071 k$ (2; 2)	[0, 0, 0, 0, 0, 0, 0] →57+82+95 +150+206+144 =734 k$ (1; 1)
7. Weak seasonality	[0, 0, 0, 0, 0, 0, 0] →1,563+1,748+5,583 +9,089+22,739+32,454 =73,176 k$ (1; 2)	[0, 1, 4, 6, 4, 2, 2] →83+154+55 +174+79+179 =724 k$ (1; 1)	[2, 1, 2, 1, 2, 0, 0] →68+90+40 +66+50+41 =354 k$ (1; 1)
8. Medium seasonality	[0, 0, 0, 0, 0, 0, 0] →509+864+1,494 +1,886+10,492+14,645 =29,890 k$ (2; 2)	[5, 4, 14, 9, 7, 4, 4] →129+151+267 +342+519+571 =1,979 k$ (1; 1)	[4, 4, 5, 5, 3, 0, 0] →91+135+191 +271+324+230 =1,242 k$ (2; 2)
9. Strong seasonality	[0, 0, 0, 0, 0, 0, 0] →621+958+2,902 +5,893+29,358+43,320 =83,052 k$ (2; 2)	[8, 8, 16, 16, 12, 16, 16]→160+220+311 +480+868+887 =2,926 k$ (2; 2)	[6, 6, 11, 11, 11, 8, 8]→162+239+294 +415+522+428 =2,058 k$ (2; 2)

Table C.7: Optimal initial inventory for two-week ordering and shipping delays when all initial inventories may be different except by the Sawmill (recalled in Table 8.1).

C.4 The particular case of the Sawmill

We now study how the Sawmill should take into account both its lumber and paper requirements when this company places orders. An illustration of the importance of this question is the fact that the PaperRetailer, the PaperWholesaler and the PaperMill may incur backorders if the Sawmill ignores them because it only focuses on delivering lumbers. Precisely, the Sawmill is modelled as two subcompanies sharing incoming transports and placing a common order (Op_w^6, Θp_w^6). Therefore, the Sawmill can either base its orders on lumber incoming orders ($Oi_w^{\text{6-lumber}}$, $\Theta i_w^{\text{6-lumber}}$) or on paper incoming orders ($Oi_w^{\text{6-paper}}$, $\Theta i_w^{\text{6-paper}}$). To fulfill these two incoming orders, each subcompany would like to place either the order ($Op_w^{\text{6-lumber}}$, $\Theta p_w^{\text{6-lumber}}$) or the order ($Op_w^{\text{6-paper}}$, $\Theta p_w^{\text{6-paper}}$). In general, $(Op_w^6, \Theta p_w^6) = f(Op_w^{6-lumber}, Op_w^{\text{6-paper}}, \Theta p_w^{6-lumber}, \Theta p_w^{6-paper})$. In all this thesis, the aggregation method is a mean, that is, the function f calculates two means, as shown by Equation C.1. The cost incurred by this first method has been presented in Table C.7.

$$(Op_w^6, \Theta p_w^6) = (\frac{Op_w^{\text{6-lumber}} + Op_w^{\text{6-paper}}}{2}, \frac{\Theta p_w^{\text{6-lumber}} + \Theta p_w^{\text{6-paper}}}{2}) \tag{C.1}$$

In this section, we study the supply chain behaviour when f is different. First, we set f as a maximum, according to Equation C.2. The costs incurred are presented in Table C.8.

$$(Op_w^6, Op_w^6) = (\max(Op_w^{6-lumber}, Op_w^{\text{6-paper}}), \max(\Theta p_w^{\text{6-lumber}}, \Theta p_w^{\text{6-paper}})) \tag{C.2}$$

This is the aggregation method used in our previous papers [Moyaux et al., 2004c, 2003a,b]. Next, we assume in Table C.9 that the Sawmill only considers its paper requirements, as illustrated by Equation C.3:

$$(Op_w^6, Op_w^6) = (Op_w^{\text{6-paper}}, \Theta p_w^{\text{6-paper}}) \tag{C.3}$$

Finally, we assume in Table C.10 that the Sawmill only considers its lumber requirements, as reflected by Equation C.4:

$$(Op_w^6, Op_w^6) = (Op_w^{\text{6-lumber}}, \Theta p_w^{\text{6-lumber}}) \tag{C.4}$$

We can now explain the meaning of the numbers between parenthesis in Tables C.7, C.8, C.9 and C.10. It is the ordering of incurred C and C^6 under the four previous scenarios. For example, let us consider Scheme A" under the Step demand. In Table C.10, i.e., when the Sawmill only consider its lumber requirements to place orders, the whole supply chain has an overall cost $C = 120,656$ k\$. On the other hand, in Table C.7, i.e., when the Sawmill orders the mean of its lumber and paper needs, $C = 307,609$ k\$, which

Market cons. pattern	Ordering scheme A"	Ordering scheme B	Ordering scheme D
1. Step	[0, 0, 0, 0, 0, 0, 0] →2,393+5,527+12,944 +30,845+87,224+184,786 =323,719 k$ (3; 3)	[0, 0, 0, 30, 0, 0, 0] →257+260+222 +671+340+1,310 =3,060 k$ (4; 4)	[0, 0, 0, 0, 13, 23, 23] →111+206+78 +157+279+629 =1,460 k$ (4; 4)
2. Inversed step	[0, 0, 0, 0, 0, 0, 0] →1,014+1,921+11,293 +20,852+65,194+126,389 =226,663 k$ (3; 3)	[0, 0, 0, 0, 0, 0, 0] →30+43+93 +135+254+1,722 =2,277 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →30+43+51 +74+101+349 =646 k$ (4; 4)
3. Dirac	[0, 0, 0, 0, 0, 0, 0] →704+980+2,282 +6,001+23,080+46,401 =79,448 k$ (3; 3)	[0, 0, 0, 6, 0, 0, 0] →59+28+58 +167+97+3,314 =3,723 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →48+81+53 +80+96+714 =1,072 k$ (4; 4)
4. Inversed Dirac	[0, 0, 0, 0, 0, 0, 0] →2,867+6,581+16,901 +37,137+102,015+215,589 =381,090 k$ (3; 3)	[0, 0, 0, 0, 0, 0, 0] →18+35+19 +35+36+2,110 =2,253 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →40+75+37 +71+49+455 =728 k$ (4; 4)
5. Increase	[0, 0, 0, 0, 0, 0, 0] →1,154+2,307+4,719 +16,647+41,186+89,044 =155,057 k$ (3; 3)	[10, 10, 26, 26, 42, 29, 29]→24+13+43 +41+81+410 =612 k$ (3; 4)	[9, 10, 19, 20, 33, 24, 24]→22+12+22 +31+76+224 =389 k$ (2; 2)
6. Decrease	[0, 0, 0, 0, 0, 0, 0] →3,079+6,570+22,088 +44,117+127,175+255,475 =458,504 k$ (3; 3)	[0, 0, 0, 0, 0, 0, 0] →57+82+141 +204+312+677 =1,473 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →57+82+104 +150+206+253 =851 k$ (4; 4)
7. Weak seasonality	[0, 0, 0, 0, 0, 0, 0] →1,976+2,036+8,025 +13,337+20,673+39,131 =85,178 k$ (3; 3)	[0, 0, 4, 7, 0, 0, 0] →81+131+79 +181+67+5,179 =5,718 k$ (4; 4)	[2, 2, 1, 1, 1, 0, 0] →50+108+49 +53+62+1,067 =1,388 k$ (4; 4)
8. Medium seasonality	[0, 0, 0, 0, 0, 0, 0] →481+743+1,986 +2,272+8,204+18,844 =32,530 k$ (3; 3)	[5, 4, 4, 10, 4, 0, 0] →112+163+311 +305+677+3,404 =4,972 k$ (4; 4)	[4, 4, 5, 5, 2, 0, 0] →91+131+242 +276+351+764 =1,856 k$ (4; 4)
9. Strong seasonality	[0, 0, 0, 0, 0, 0, 0] →630+997+2,701 +5,603+26,684+58,517 =95,132 k$ (3; 3)	[8, 8, 11, 16, 12, 0, 0] →183+220+392 +509+857+2,728 =4,889 k$ (4; 4)	[7, 7, 10, 11, 10, 0, 0] →171+235+305 +430+524+707 =2,372 k$ (4; 4)

Table C.8: Tables C.7 when the Sawmill bases its order on the maximum of lumber and paper requirements.

Market cons. pattern	Ordering scheme A"	Ordering scheme B	Ordering scheme D
1. Step	[0, 0, 0, 0, 0, 0, 0] →3,499+7,337+18,095 +39,383+98,223+211,855 =378,392 k$ (4; 4)	[0, 0, 0, 30, 0, 54, 54] →168+194+134 +593+298+1,001 =2,388 k$ (3; 3)	[0, 0, 0, 0, 12, 24, 24] →109+208+75 +159+265+485 =1,302 k$ (3; 3)
2. Inversed step	[0, 0, 0, 0, 0, 0, 0] →1,051+1,822+14,212 +26,002+83,653+170,972 =297,712 k$ (4; 4)	[0, 0, 1, 0, 0, 12, 12] →30+48+113 +140+260+322 =913 k$ (3; 3)	[0, 0, 0, 0, 0, 12, 12] →30+43+51 +74+101+220 =517 k$ (3; 3)
3. Dirac	[0, 0, 0, 0, 0, 0, 0] →690+1,022+3,809 +8,207+28,343+60,550 =102,621 k$ (4; 4)	[0, 0, 0, 0, 0, 6, 6] →39+84+39 +84+83+237 =566 k$ (3; 3)	[0, 0, 0, 0, 0, 0, 0] →48+95+53 +94+110+59 =459 k$ (3; 3)
4. Inversed Dirac	[0, 0, 0, 0, 0, 0, 0] →4,076+7,999+22,233 +47,950+129,481+264,112 =475,851 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →36+61+37 +62+62+93 =351 k$ (2; 3)	[0, 0, 0, 0, 0, 0, 0] →58+102+55 +97+75+42 =431 k$ (3; 3)
5. Increase	[0, 0, 0, 0, 0, 0, 0] →1,523+3,158+7,080 +19,872+48,956+96,250 =176,839 k$ (4; 4)	[10, 10, 26, 26, 42, 58, 58]→18+12+37 +40+80+248 =435 k$ (1; 1)	[10, 11, 19, 19, 28, 34, 34]→8+22+19 +28+49+256 =383 k$ (1; 3)
6. Decrease	[0, 0, 0, 0, 0, 0, 0] →3,859+7,019+26,345 +54,854+159,617+340,845 =592,539 k$ (4; 4)	[0, 0, 0, 0, 0, 0, 0] →57+82+139 +204+312+251 =1,045 k$ (1; 1)	[0, 0, 0, 0, 0, 8, 8] →52+82+90 +150+206+213 =793 k$ (2; 2)
7. Weak seasonality	[0, 0, 0, 0, 0, 0, 0] →1,784+1,589+2,650 +6,832+33,264+62,366 =108,485 k$ (4; 4)	[1, 0, 8, 7, 0, 8, 8] →42+136+155 +188+55+388 =964 k$ (3; 3)	[0, 1, 1, 1, 3, 1, 1] →39+95+63 +62+64+58 =380 k$ (2; 2)
8. Medium seasonality	[0, 0, 0, 0, 0, 0, 0] →506+845+1,703 +2,004+11,809+23,449 =40,316 k$ (4; 4)	[5, 4, 11, 11, 18, 7, 7] →130+166+319 +349+646+642 =2,252 k$ (3; 3)	[4, 4, 5, 5, 2, 0, 0] →91+131+203 +276+314+182 =1,197 k$ (1; 1)
9. Strong seasonality	[0, 0, 0, 0, 0, 0, 0] →703+893+3,607 +7,146+32,116+70,095 =114,560 k$ (4; 4)	[8, 8, 22, 18, 16, 22, 22] →197+240+396 +539+846+1,041 =3,259 k$ (3; 3)	[6, 6, 11, 11, 11, 6, 6] →160+239+276 +404+518+457 =2,054 k$ (1; 3)

Table C.9: Table C.7 when the Sawmill bases its order only on paper requirements.

Market cons. pattern	Ordering scheme A″	Ordering scheme B	Ordering scheme D
1. Step	[0, 0, 0, 0, 0, 0, 0] →2,393+5,527+12,931 +30,454+15,007+54,344 =120,656 k$ (1; 1)	[0, 0, 0, 7, 12, 0, 0] →257+384+222 +444+303+223 =1,833 k$ (1; 1)	[0, 0, 0, 0, 12, 24, 24] →109+208+75 +159+265+346 =1,163 k$ (1; 1)
2. Inversed step	[0, 0, 0, 0, 0, 0, 0] →1,451+2,383+12,225 +21,314+22,686+36,438 =96,497 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →33+66+96 +157+277+161 =790 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →30+43+51 +74+101+126 =423 k$ (1; 1)
3. Dirac	[0, 0, 0, 0, 0, 0, 0] →704+980+2,282 +5,976+6,438+13,280 =29,660 k$ (1; 1)	[0, 0, 0, 6, 0, 0, 0] →59+28+58 +167+97+112 =521 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →48+95+53 +94+110+53 =453 k$ (1; 1)
4. Inversed Dirac	[0, 0, 0, 0, 0, 0, 0] →3,896+7,268+17,444 +36,671+24,407+61,788 =151,474 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →36+61+37 +62+62+73 =331 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →52+92+49 +88+66+34 =382 k$ (1; 1)
5. Increase	[0, 0, 0, 0, 0, 0, 0] →1,154+2,307+4,702 +15,264+6,093+34,221 =63,741 k$ (1; 1)	[10, 10, 25, 26, 42, 82,82]→19+22+38 +60+113+398 =650 k$ (4; 3)	[11, 9, 19, 19, 82, 27, 27]→15+38+19 +48+377+147 =645 k$ (4; 1)
6. Decrease	[0, 0, 0, 0, 0, 0, 0] →4,703+7,729+22,827 +44,033+38,117+73,234 =190,643 k$ (1; 1)	[0, 0, 0, 0, 0, 0, 0] →57+82+141 +204+312+301 =1,097 k$ (3; 3)	[0, 0, 0, 0, 0, 0, 0] →57+82+104 +150+206+229 =827 k$ (3; 3)
7. Weak seasonality	[0, 0, 0, 0, 0, 0, 0] →3,296+3,941+12,559 +13,121+20,238+22,332 =75,487 k$ (2; 1)	[0, 0, 4, 6, 2, 8, 8] →95+116+96 +124+163+257 =851 k$ (2; 2)	[2, 2, 1, 1, 3, 3, 3] →62+89+43 +62+56+95 =407 k$ (3; 3)
8. Medium seasonality	[0, 0, 0, 0, 0, 0, 0] →481+743+1,870 +2,269+4,770+5,347 =15,480 k$ (1; 1)	[4, 4, 15, 12, 9, 4, 4] →121+156+265 +330+630+614 =2,116 k$ (2; 2)	[4, 4, 5, 5, 3, 3, 3] →94+135+195 +274+333+267 =1,298 k$ (3; 3)
9. Strong seasonality	[0, 0, 0, 0, 0, 0, 0] →630+997+2,701 +5,603+9,647+14,958 =34,536 k$ (1; 1)	[8, 8, 17, 16, 13, 10, 10] →164+221+388 +464+909+741 =2,887 k$ (1; 1)	[6, 6, 11, 11, 11, 9, 9] →158+239+313 +415+537+420 =2,082 k$ (3; 1)

Table C.10: Table C.7 when the Sawmill bases its order only on lumber requirements.

is much higher than with the previous aggregation method. Next, $C = 323,719$ k\$ in Table C.8, and $C = 378,392$ k\$ in Table C.9. $C = 120,656$ k\$ in Table C.10 is the lowest of these four values of C. This explains why the first number between parenthesis for Scheme A″ and Step demand is a "one" in Table C.10. The second highest cost has a "two" as first number between parenthesis in Table C.7, etc.

When these first numbers between parenthesis are considered, we can note that the aggregation with the max function almost always incurs the highest C with Schemes B and D, and the second highest cost with Scheme A″. Unfortunately, our previous papers [Moyaux et al., 2004c, 2003a,b] use this aggregation method, which makes the supply chain efficiency lower in these papers than in this dissertation! We did so in order to avoid backorders, and thus to reduce costs, because the cost of a backordered unit is twice the cost of a unit in inventory. Unfortunately, experiments show that this was not a good strategy.

Results are less clear for the three other aggregation methods, because we cannot find such regularity in Tables C.7, C.9 and C.10. The first number between parenthesis in Table C.10 is more often "one" than in the two other Tables, but if we focus on values of C, this result is only clear for ordering scheme A″, because values of C for Schemes B and D are very near in Tables C.7 and C.10. In conclusion, the aggregation method used by the Sawmill in this dissertation is not the best one, but its results are not very far from it.

Next, the second number between parenthesis has the same meaning as the first one, but for ordering the Sawmill's costs C^6 instead of the overall supply chain costs C. This second number is very important, because it concerns the company that has to decide which aggregation method to use: the Sawmill will choose the ordering scheme indexed as "one" for itself (i.e., the second number between parenthesis), rather than for the supply chain. In general, the two numbers between parenthesis are identical, and therefore, the above comments still hold. This also indicates that the Sawmill's interest about this question is the same as the whole supply chain's interest.