



**HAL**  
open science

# Méthodes stochastiques dans les problèmes de placement

Frederik Premti

► **To cite this version:**

Frederik Premti. Méthodes stochastiques dans les problèmes de placement. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1983. Français. NNT : . tel-00306962

**HAL Id: tel-00306962**

**<https://theses.hal.science/tel-00306962>**

Submitted on 28 Jul 2008

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# THESE

*présentée à*

**l' Université Scientifique et Médicale de Grenoble**

*pour obtenir le grade de*

**DOCTEUR DE 3<sup>ème</sup> CYCLE**

**"Recherche Opérationnelle"**

*par*

**Frederik PREMTI**



## **METHODES STOCHASTIQUES DANS LES PROBLEMES DE PLACEMENT**



**Thèse soutenue le 7 juillet 1983 devant la Commission d'Examen :**

<b>Monsieur</b>	<b>M. SAKAROVITCH</b>	<b>:</b>	<b>Président</b>
<b>Messieurs</b>	<b>R. MAYNARD</b>	<b>}</b>	<b>Examineurs</b>
	<b>Ch. PAYAN</b>		
	<b>J.P. UHRY</b>		



# UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

## MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

### PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDQ Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

#### PROFESSEURS DE 2<sup>ème</sup> CLASSE

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

.../...

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie



Je tiens à remercier :

Michel SAFAROVITCH de m'avoir accueilli dans son équipe et de l'honneur qu'il me fait de présider le jury de cette thèse.

Jean-Pierre ULPY qui a été toujours disponible pour me conseiller, me guider et m'encourager tout au long de ce travail.

Monsieur Roger MAYNARD et Monsieur Charles PAYAN qui ont bien voulu faire partie du jury.

Annie UHRY pour tous les conseils compétents pendant l'élaboration du quatrième chapitre.

Claudie FURET pour l'aide précieuse dans la rédaction de cette thèse.

Monsieur IGLESIAS et toute l'équipe de reprographie pour la promptitude et le soin qu'ils ont apportés à la reproduction de cette thèse.

Je profite de l'occasion pour exprimer toute ma sympathie à tous les membres de l'équipe de Recherche Opérationnelle pour l'ambiance de travail qu'ils m'ont créée.

Frederik PREMTI.





## TABIE DES MATIERES

### INTRODUCTION

### CHAPITRE 1 PROBLEME DE DECOUPE DES PIECES UNIDIMENSIONNELLES ET RECTANGULAIRES

- 1.1 - Formulation mathématique du problème de découpe.
- 1.2 - Classification des problèmes de découpe.
- 1.3 - Résolution du problème de découpe.
- 1.4 - Problème de placement.
- 1.5 - Résolution du problème de placement.
  - 1.5.1 - Découpe unidimensionnelle.
  - 1.5.2 - Découpe bidimensionnelle.
    - 1.5.2.a - Découpe guillotine à grande demande.
    - 1.5.2.b - Découpe guillotine à faible demande.
    - 1.5.2.c - Découpe non-guillotine.
- 1.6 - Conclusions.

### CHAPITRE II LA METHODE DE SIMULATION DE RECUIT

- 2.0 - Introduction.
- 2.1 - La méthode de recuit.
- 2.2 - La méthode de Metropolis.
- 2.3 - La méthode de Kirkpatrick.
- 2.4 - Discussions sur la méthode de Kirkpatrick.

2.5 - Applications pratiques de la méthode de Kirkpatrick.

2.6 - Conclusions.

### CHAPITRE III - UN NOUVEL ALGORITHME POUR LE PLACEMENT DES PIÈCES IRREGULIÈRES.

3.0 - Introduction.

3.1 - Le problème à résoudre.

3.2 - L'idée principale de la méthode "thermodynamique".

3.3 - La mise en oeuvre de la méthode "thermodynamique".

3.3.1 - Descriptions d'une configuration du système.

3.3.2 - La fonction d'énergie.

3.3.3 - Configurations voisines.

3.3.3.a - Transformations locales.

3.3.3.b - Transformations globales.

### CHAPITRE IV - UNE NOUVELLE APPROCHE POUR L'UTILISATION DES METHODES STOCHASTIQUES DANS LES PROBLEMES DE PLACEMENT.

4.0 - Introduction.

4.1 - Programme automatique de placement.

4.1.1 - Stratégie globale.

4.1.2 - Stratégie locale.

4.2 - La nouvelle approche.

4.2.1 - Expérimentations en vêtements.

4.2.2 - Expérimentations dans l'industrie de tricot.

- 4.3 - Traitement statistique des résultats obtenus.
  - 4.3.1 - Ajustement par une loi gamma.
  - 4.3.2 - Ajustement par une loi de Weibull
  - 4.3.3 - Ajustement par une loi de répartition des valeurs extrêmes.
- 4.4 - Estimation du nombre d'essais à effectuer.
- 4.5 - Utilisation pratique.
- 4.6 - Conclusions.

BIBLIOGRAPHIE.



## INTRODUCTION

Sous le nom de "problème de découpe" en Recherche Opérationnelle, on désigne la formulation mathématique d'un problème rencontré assez souvent dans la pratique de nombreuses entreprises : comment, à partir d'un stock de matière première, sous forme de plusieurs unités de formes et de dimensions différentes, produire des pièces suivant une demande connue ou non à l'avance, avec le minimum de matière première.

Ce problème se pose dans l'industrie du métal (découpe des barres, des plaques de tôle, etc), du verre, du bois, du papier, du cuir, du plastique, des vêtements, etc. Il se rencontre aussi dans plusieurs autres activités, très lointaines du processus proprement dit de découpe. Ci-dessous suit une liste de telles activités :

- répartition des tâches entre plusieurs processeurs
- répartition du temps de calcul entre plusieurs tâches
- répartition de la mémoire d'un ordinateur entre plusieurs programmes
- remplissage d'un four de recuit en sidérurgie
- remplissage d'un conteneur (camion, wagon etc;)
- emballage des marchandises
- placement des circuits intégrés dans une puce.

Cette liste est loin d'être exhaustive.

Afin de simplifier le langage, on précise ici quelques termes que l'on va utiliser tout au long de cette thèse :

"Plaque" - une unité de matière première. Selon le problème concret la "plaque" sera une barre unidimensionnelle, une plaque rectangulaire, un objet tridimensionnel, un objet de forme irrégulière, le temps, la mémoire d'un ordinateur etc.

"Patron" - une manière de "découper" "la plaque". Ici le

terme "découper" est mis entre guillemets parce que, selon le cas, il s'agit de découper, de remplir etc.

"Problème de placement"(PP) - trouver le "meilleur" patron compte tenu des pièces à "découper".

"Problème de découpe"(PD) - trouver un ensemble de patrons pour satisfaire à la demande des pièces.

Le problème de placement est la partie la plus importante du problème de découpe pour plusieurs raisons:

- Pour la résolution du problème de découpe, il existe différentes méthodes et heuristiques. Indépendamment de la méthode ou heuristique choisie, le problème de placement est toujours présent. Autrement dit, on ne peut résoudre "efficacement" le problème de découpe que si on sait résoudre efficacement le problème du placement.

- Dans la découpe en continu, il n'existe pas vraiment "un problème de découpe", mais une suite de problèmes de placement. Cette situation se rencontre aussi dans tous les problèmes de découpe où la demande n'est pas connue à l'avance. Dans ces cas, le problème de placement est encore plus difficile parce que il y a souvent d'autres contraintes (surtout en temps de calcul) introduites par le "cycle" de ligne.

- La flexibilité du modèle de découpe est due principalement au problème de placement. Les contraintes supplémentaires, en général, sont prises en compte pendant la résolution de celui-ci.

Au début de l'utilisation du modèle de découpe, le problème de placement était résolu manuellement. Avec l'introduction de plus en plus large des ordinateurs dans la pratique quotidienne des entreprises, la nécessité d'une méthode automatique ou semi-automatique pour le résoudre devenait pressante et c'est la source d'un intérêt toujours croissant pour la recherche en ce domaine, intérêt qui a donné un grand nombre de méthodes et heuristiques utilisées aujourd'hui.

Cependant, ce problème n'est résolu de manière satisfaisante que pour quelques classes de découpe : découpe uni-

dimensionnelle, découpe bi ou tridimensionnelle en multistades de rectangles dans des rectangles, etc (cf. Chapitre 1). Pour les autres classes et surtout pour des pièces de formes irrégulières, ce problème est loin d'être résolu définitivement. Les meilleurs résultats sont obtenus par des méthodes manuelles, interactives ou semi-automatiques.

Dans les premières, un placeur expérimenté essaie différents placements en dialogue avec un (micro-)(mini-)ordinateur. L'ordinateur est commandé par des instructions simples, un crayon magnétique, etc. Il sert à dessiner les pièces sur l'écran, les faire bouger selon les ordres du placeur et mémoriser les placements acceptés.

Dans les deuxièmes, l'ordinateur, à l'aide de programmes plus ou moins sophistiqués, produit des placements qui sont examinés par un placeur expérimenté. Ce dernier, peut les accepter tels-quels ou les retoucher comme dans les méthodes manuelles interactives.

Aujourd'hui, les résultats obtenus par les méthodes automatiques ne sont pas tout à fait satisfaisants et ils ne peuvent concurrencer ceux obtenus manuellement que dans certains cas spéciaux. L'utilisation pratique de ces méthodes n'est justifiée, pour le moment, que pour les procédés automatiques de découpe où l'ordinateur commande directement la machine à découper et dans les cas où la perte supplémentaire, introduite par ces méthodes, est plus faible que le coût des autres procédés.

Obtenir automatiquement des placements d'aussi bonne qualité que ceux obtenus manuellement, constitue un véritable défi pour l'optimisation combinatoire (un peu analogue à celui des machines à jouer aux échecs). Il n'est d'ailleurs pas sûr que ce but puisse être atteint sans faire appel aux techniques de l'intelligence artificielle permettant la prise en compte de l'expérience humaine et de la spécificité du problème.

\* \* \*

\*



La présente étude est consacrée au problème de placements de pièces de formes irrégulières et plus précisément à l'application des méthodes stochastiques pour le résoudre.

Dans le premier chapitre on donne une formulation mathématique du problème de découpe comme un programme linéaire en nombres entiers, suivi d'une classification de problèmes de découpe selon les dimensions des "plaques" et des pièces et selon le procédé de découpe. Ensuite, en 1.3, est présentée la méthode de Gilmore-Gomory dont l'idée principale est de diviser le problème initial en un programme linéaire et un problème de placement. Cette idée originale a ouvert, pour la première fois, de grandes possibilités théoriques et pratiques pour la résolution automatique du problème de découpe et sert comme point de départ pour plusieurs développements possibles. En 1.4 et 1.5 est discuté le problème de placement, son rôle clé dans le problème de découpe, et on présente différents algorithmes et heuristiques pour le placement des pièces unidimensionnelles et rectangulaires. On discute en détail certaines de ces méthodes et heuristiques et on donne un aperçu général des autres, afin de mieux comprendre les difficultés pratiques qu'on rencontre même dans le cas le plus simple, et aussi pour présenter quelques heuristiques récentes très efficaces.

Dans le deuxième chapitre on présente une heuristique générale pour résoudre les problèmes d'optimisation combinatoire. Elle peut être adaptée facilement à une vaste classe de ces problèmes. Cette heuristique est due à Kirkpatrick et al. et est issue d'une analogie qui existe entre les problèmes d'Optimisation Combinatoire et le procédé expérimental de recuit, utilisé en thermodynamique pour trouver des états stables. Le chapitre commence par une description rapide du procédé de recuit et d'une méthode de Monte Carlo modifiée, introduite par Metropolis et al., pour trouver la configuration d'équilibre d'une collection d'atomes à une tempé-

rature donnée. Ensuite, on donne une description assez détaillée de la méthode de Kirkpatrick, en tenant compte du fait que cette méthode récente n'est pas bien connue pour le moment et qu'elle peut présenter de l'intérêt pour beaucoup de chercheurs et praticiens en Optimisation Combinatoire. A la fin du chapitre on donne quelques applications pratiques de cette méthode. Les résultats obtenus confirment son efficacité.

Dans le troisième chapitre, on présente un algorithme, basé sur la méthode de Kirkpatrick, pour le placement des pièces irrégulières. La présentation de l'algorithme est assez détaillée pour faciliter la mise en oeuvre par les personnes intéressées. Ensuite on en présente des expérimentations pratiques, pour le placement de rectangles et de pièces irrégulières, puis on étudie ses avantages et ses défauts. On donne aussi quelques réflexions pour améliorer la performance de celui-ci.

Dans le dernier chapitre, on présente une autre approche de l'utilisation des méthodes stochastiques dans le problème de placement. L'équipe de Recherche Opérationnelle de l'IMAG a mis en oeuvre un logiciel très efficace pour le placement des pièces irrégulières. Plusieurs versions de ce logiciel (PAMPA, pour le placement dans l'industrie du tricot, RIMBAUD, pour le placement des pièces de tôle etc.) sont utilisés dans l'industrie avec des résultats encourageants. Après une description générale de ce logiciel, on présente de quelle façon, en introduisant l'aléa à différents niveaux du programme, on obtient de meilleurs placements, en échange d'un plus grand temps de calcul.

Les résultats de quelques centaines de placements effectués à l'aide cette méthode, nous permettent de déduire que ceux-ci sont distribués selon des lois connues en statistique. Cela nous donne un moyen de prévoir statistiquement le nombre d'essais nécessaires pour obtenir un "bon" placement et, en même temps, d'envisager l'utilisation pratique suivante:

On effectue un certain nombre de placements. On estime

les paramètres de la loi respective et le nombre d'essais nécessaires pour trouver un meilleur placement. Ce nombre nous permet de décider si l'on continue les essais ou si on se contente du meilleur placement obtenu jusque là.

Ces résultats présentent de l'intérêt pour les procédés de découpe, où le temps de calcul n'est pas critique.

En dernière partie, on donne une bibliographie assez complète concernant le problème de découpe. Certains articles de cette bibliographie ne sont pas cités directement dans cette thèse, mais nous avons considéré opportun de les faire figurer pour une information plus complète.

## CHAPITRE I

### 1.1 Formulation mathématique du problème de découpe

On dispose de plusieurs "plaques" de matière première. Les "plaques" sont toutes de la même forme et de la même taille.

L'hypothèse que toutes les "plaques" sont identiques est faite uniquement pour simplifier les notations. Si les "plaques" sont de formes différentes et de dimensions différentes, la formulation suivante change très peu. C'est surtout dans la méthode de résolution que ces différences jouent un rôle plus important.

On suppose qu'on a à découper  $m$  pièces différentes et que la pièce  $i$  est demandée en quantité  $d_i$ .

Il existe plusieurs manières de découper une "plaque". Chacune de ces manières est appelée un patron. Chaque patron  $j$  est caractérisé par le vecteur :

$$a_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix}$$

où  $a_{ij}$  ( $i = 1, 2, \dots, m$ ) représente le nombre des pièces de type  $i$  produites par l'utilisation du patron  $j$ .

Soit  $A = [a_1, a_2, \dots, a_n]$ , la matrice de tous les patrons.

Soit  $X$  le vecteur  $\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$  où  $x_j$  ( $j = 1, 2, \dots, n$ ), représente le nombre d'utilisations du patron  $j$ .

En utilisant les notations ci-dessus, le problème de découpe :

"Produire les quantités demandées pour chaque pièce à coût minimum" peut être modélisé par le programme linéaire :

$$\begin{array}{rcl} \min Z & = & \sum c_j x_j \quad (1) \\ \text{sous les contraintes :} & & \\ & & Ax = b \quad (2) \\ & & x \geq 0 \quad (3) \\ & & x \text{ entier} \quad (4) \end{array} \quad \left. \vphantom{\begin{array}{rcl} \min Z & = & \sum c_j x_j \\ \text{sous les contraintes :} \\ & & Ax = b \\ & & x \geq 0 \\ & & x \text{ entier} \end{array}} \right\} \text{(PD)}$$

où  $c_j$  ( $j = 1, 2, \dots, n$ ) prend différentes valeurs suivant l'objectif poursuivi :

Par exemple,

$c_j = 1$  pour tout  $j$  est équivalent à : avec le nombre minimum de "plaques", etc.

Cette formulation est donnée pour la première fois par Kantorovitch en 1939 et reformulée indépendamment par Eismann \*) puisque le papier de Kantorovitch, publié en russe, est resté longtemps inconnu.

Les contraintes (4) sont souvent négligées dans les problèmes de découpe à grandes demandes ou dans les problèmes où le coût de la "plaque" n'est pas assez élevé pour justifier les dépenses supplémentaires en temps de calcul. Dans ces cas, une solution arrondie du problème (1) - (3), suivie par découpe des quelques "plaques" supplémentaires, pour satisfaire à la demande, est une heuristique presque toujours utilisée.

## 1.2 Classification des problèmes de découpe

Les problèmes de découpe sont classés selon les dimensions significatives de la "plaque" et selon le procédé

\*) Suivant Hinxman (1980), cette formulation est donnée pour la première fois par Paull et Walter dans :  
"The trim problem : an application of linear programming to the manufacture of newsprint paper."  
Presented at Annual Meeting of Econometric Society, Montreal, Sept. 10-13 (1954)

de découpe.

Selon les dimensions significatives, on distingue :

- le problème unidimensionnel où il y a une seule dimension significative pour la "plaque".

Ce problème apparaît dans la découpe des barres d'acier, la "découpe" de temps, etc.

- le problème à une dimension et demi où la "plaque" a une dimension fixe et une dimension variable. C'est le problème rencontré dans la découpe de verre, de papier, de tricot, etc.

Dans ce cas, le problème d'optimisation consiste à minimiser la longueur de la matière première utilisée.

- le problème bidimensionnel où la "plaque" est rectangulaire et de dimensions fixes. C'est le problème le plus étudié parce qu'il se rencontre assez souvent en pratique (découpe de plaques de bois, de verre, etc.) et aussi parce que la découpe des pièces de forme quelconque dans des rectangles se ramène souvent dans ce cas.

- etc.

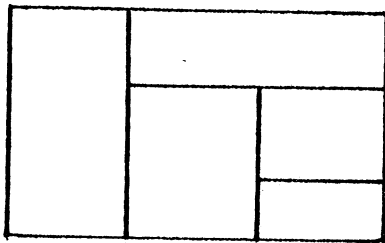
Selon le procédé de découpe, on distingue :

- découpe guillotine et non-guillotine.

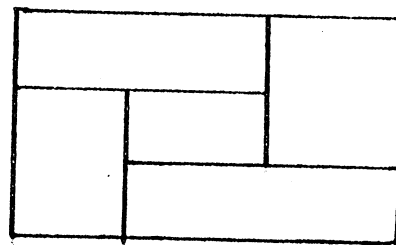
On appelle découpe guillotine, le procédé de découpe qui va d'un bout de la plaque à l'autre.

C'est une forme de découpe qui se présente dans la plupart des procédés mécaniques de découpage. Avec l'apparition des nouveaux procédés de découpage (oxycoupage, découpage avec des rayons laser, découpage avec fil d'eau, etc.), la découpe non-guillotine se présente de plus en plus.

La différence entre la découpe guillotine et non-guillotine est illustrée par la fig.1 :



a) découpe guillotine

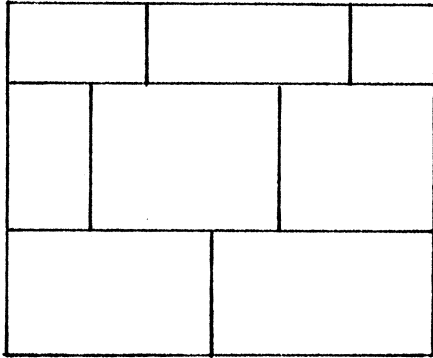


b) découpe non-guillotine

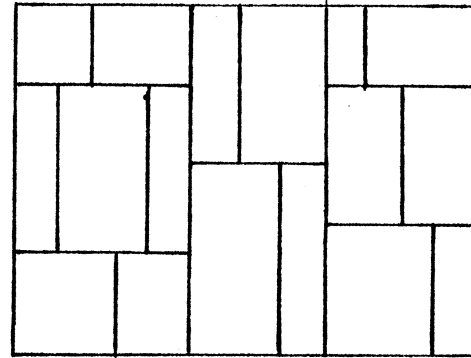
fig. 1

- découpe en 2-stades (en 3-stades et en général en n-stades)

C'est un cas spécial de la découpe guillotine. La "plaque" est d'abord coupée en plusieurs bandes et chaque bande est ensuite coupée séparément comme le montre la figure 2.



a) Découpe en 2-stades  
premier stade horizontal  
deuxième stade vertical



b) Découpe en 3-stades  
premier stade vertical  
deuxième stade horizontal  
troisième stade vertical

fig. 2 - Découpe en 2- et 3-stades.

La découpe en n-stades se ramène facilement à n problèmes de découpe unidimensionnelle.

etc.

Cette classification est valable pour des pièces de formes quelconques (bien sûr pour des pièces "non régulières", la découpe guillotine ou unidimensionnelle n'existe pas), mais dans la littérature, faute d'autres précisions, elle est implicitement appliquée pour des pièces de formes rectangulaires.

### 1.3 Résolution du problème de découpe

Les difficultés pour résoudre directement le problème (PD) sont dues :

(i) au nombre très élevé de patrons différents, c.a.d. au nombre très élevé des colonnes de la matrice A. Même pour des problèmes de taille modeste, ce nombre est très grand. Gilmore et Gomory [1961] présentent un exemple de découpe unidimensionnelle de rouleaux de papier de longueur 200 inches et avec 40 pièces de longueur comprise entre 20 inches et 80 inches (des problèmes de cette taille se rencontrent souvent en pratique). Pour ce problème, le nombre de patrons différents peut dépasser 10 millions et parfois 100 millions.

(ii) à la condition d'intégrité (4), car on sait que la résolution d'un programme linéaire en nombres entiers peut être très difficile même pour des problèmes de taille assez petite.

Plusieurs méthodes et heuristiques ont été élaborées pour surmonter ces difficultés.

a) Dans les premières années d'utilisation de ce modèle, Eisemann a suggéré de se limiter à un sous-ensemble de "bons patrons", sélectionnés d'après certains critères dans un ensemble de patrons construits manuellement. Bien que cette méthode donne de bonnes approximations, elle est très laborieuse. De plus, la sélection à priori de "bons patrons" est très subjective.

Gilmore et Gomory [1961] ont proposé une méthode qui résout le problème (1)-(3) sans construire d'avance tous les patrons possibles mais, si besoin est, au fur et à mesure qu'on progresse dans la solution.

Cette technique s'appelle la "génération de colonnes".

On va décrire brièvement cette technique pour le problème de découpe unidimensionnelle.

On suppose que "la plaque" est de longueur  $L$  et les pièces de longueurs  $l_1, l_2, \dots, l_m$ .



Chaque colonne  $a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \in A$ , satisfait les contraintes :

$$\begin{matrix} a_i \geq 0 \\ a_i \text{ entier} \end{matrix} \quad i = 1, 2, \dots, m$$

$$\sum_{i=1}^m a_i l_i \leq L,$$

Pour résoudre (1)-(3), on utilise la méthode révisée du simplexe. Dans cette méthode, à chaque itération, on connaît les multiplicateurs  $V_i (i = 1, 2, \dots, m)$ , et on cherche la colonne pivot, c.a.d. la colonne  $p$  à laquelle correspond le coût réduit minimum  $\tilde{c}_p$ . Si  $\tilde{c}_p > 0$ , la colonne  $p$  est la colonne pivot, sinon on a trouvé la solution optimale.

Sachant que :

$$\tilde{c}_p = \sum_{i=1}^m V_i a_{ip} - c_p$$

celà revient à calculer :

$$\max \sum_{i=1}^m V_i a_{ip}$$

et à le comparer avec  $c_p$ .

En somme, pour trouver la colonne pivot (ou pour prouver l'optimalité) il suffit de résoudre le problème auxiliaire :

$$\max \sum_{i=1}^m V_i a_i \quad (5)$$

sous les contraintes :

$$\sum_{i=1}^m a_i l_i \leq L \quad (6)$$

$$a_i \geq 0 \quad i = 1, 2, \dots, m \quad (7)$$

$$a_i \text{ entier} \quad i = 1, 2, \dots, m \quad (8)$$

(PP)

Le problème (PP) est connu dans la littérature sous le nom "problème du sac-à-dos". La technique de génération de colonnes peut être appliquée dans n'importe quel problème de découpe. Pour cela, il suffit de changer le problème auxiliaire. Plus précisément, la contrainte (6) est remplacée par une (plusieurs) autre(s) contrainte(s) qui exprime(nt) mathématiquement les différentes contraintes du processus physique de la découpe.

b) Pour tenir compte de la contrainte (4), on distingue : problème de découpe à grande demande et à faible demande.

Dans les premiers, la demande de chaque pièce dépasse considérablement le nombre d'exemplaires de cette pièce qui peuvent être placés dans une seule "plaque" et la contrainte ;

$$x_i \leq d_i \text{ pour tout } i$$

est ignorée pendant la résolution du problème de placement (cf. 1.4) respectif. La solution de (1)-(3), arrondie et modifiée pour satisfaire les demandes marche très bien en pratique.

Dans les deuxièmes, la demande de chaque pièce peut dépasser le nombre d'exemplaires de cette pièce qui peuvent être placés dans une seule "plaque"

La contrainte ;

$$x_i \leq d_i \text{ pour tout } i$$

est impérative pendant la solution du problème de placement respectif. Dans ces cas, si la solution directe de (PD) s'avère très coûteuse, on utilise différentes heuristiques. Une heuristique souvent utilisée est celle qu'on nomme "réduction exhaustive répétitive" (cf. 1.4).

Golden [1976] propose une autre approche pour la résolution du (PD) basée sur la méthode de sous-gradient. Cette approche est stimulée par le succès de la méthode de sous-

gradient dans la résolution du problème de "voyageur de commerce". A notre connaissance, il n'existe pas de publication sur l'efficacité pratique de cette approche pour la résolution du (PD).

Les deux méthodes présentées ci-dessus (la génération de colonnes et réduction exhaustive répétitive) donnent, en général, une solution optimale de  $m$  patrons différents. Or, dans la pratique, vu que chaque patron différent exige un nouveau réglage de la machine de découpe, on souhaiterait un nombre plus petit de tels patrons en échange d'une hausse limitée de la perte. Cette contrainte est très difficile à prendre en compte dans les deux méthodes précédentes. Haessler [1975] propose une heuristique pour ce cas et Johnston [1979] donne une extension de cette heuristique.

#### 1.4 Problème de placement

Le problème auxiliaire (PP) pour la découpe unidimensionnelle ou l'analogie pour tous les autres problèmes de découpe s'appelle problème de placement.

En d'autres termes, le problème de placement s'exprime ainsi :

"Connaissant la valeur de chaque pièce, comment remplir "au mieux" une seule plaque tout en respectant les contraintes de la découpe ou, encore, trouver le patron de valeur maximale où la valeur d'un patron est définie par la somme des valeurs des pièces qui le constituent".

Dans les problèmes de découpe :

-unidimensionnelle, le problème de placement est un problème de sac-à-dos

-bidimensionnelle, le problème de placement est un problème de sac-à-dos bidimensionnel

-en 2-stades, le problème de placement se ramène en 2 problèmes de sac-à-dos unidimensionnels

etc.

En résumé, le problème de découpe est un binôme "programme linéaire-problème de placement".

Le problème de placement est la partie principale d'un problème de découpe pour deux raisons :

(i) c'est la partie la plus difficile du point de vue algorithmique et la plus coûteuse en temps de calcul.

(ii) en pratique, pour résoudre le problème (PD), on utilise assez souvent l'heuristique de "réduction exhaustive répétitive" qui consiste à trouver le meilleur patron initial et à l'utiliser jusqu'à ce que la demande pour au moins une des pièces qui le constituent soit satisfaite; trouver le meilleur patron pour les pièces qui restent et ainsi de suite.

Dans ce cas, le programme linéaire disparaît et le problème de découpe se réduit à plusieurs problèmes de placement.

### 1.5 Méthodes pour résoudre le problème de placement

Les difficultés pour résoudre le problème de placement sont évidentes puisque même dans le cas le plus simple de la découpe unidimensionnelle, il se ramène au problème de sac-à-dos, qui est NP-complet. Lorsqu'on ajoute des contraintes supplémentaires (de production, de gestion, etc...), on a une idée des difficultés que l'on peut rencontrer.

Dans ce paragraphe, on va donner un aperçu général des méthodes algorithmiques et heuristiques les plus utilisées.

#### 1.5.1 Découpe unidimensionnelle:

Les méthodes algorithmiques les plus utilisées en pratique pour résoudre le problème (5)- (8), appartiennent à deux classes : programmation dynamique et "branch and bound".

Ici, on va traiter uniquement des méthodes de programmation dynamique.

$$\text{Soit } F_k(x) = \max_{i=1}^k v_i a_i \quad (9)$$

sous les contraintes :

$$\sum_{i=1}^k l_i a_i \leq x \quad x \in N \quad (10)$$

$$a_i \in N \quad (11)$$

$$\text{Alors, } F_0(x) = 0 \quad \text{pour tout } x \quad (12)$$

$$F_k(0) = 0 \quad \text{pour tout } k \quad (13)$$

$$\text{et } F_k(x) = \max_{r_k} [r_k \cdot v_k + F_{k-1}(x - r_k l_k)] \quad (14)$$

$$\text{où } r_k \in [0, E(x/l_k)], r_k \text{ entier} \quad (15)$$

et  $E(y)$  est la partie entière de  $y$ .

Cette approche est coûteuse en temps de calcul et encombrement mémoire : pour calculer la solution optimale, on a besoin de  $F_{k-1}(x)$  et  $F_k(x)$  - deux vecteurs de dimensions  $(L + 1)$  et d'un tableau  $a_k(x)$  de dimensions  $(L + 1) \times m$  qui nous donne la valeur de  $a_k, k = 1, 2, \dots, m$ , pour chaque valeur entière de  $x \in [0, L]$ .

Gilmore et Gomory ont utilisé d'autres relations de récurrence et ont obtenu une méthode beaucoup plus efficace en temps de calcul et en encombrement mémoire.

La nouvelle forme se résume ainsi :

$$\begin{aligned} F_k(0) &= 0 \quad \text{pour tout } k \\ F_1(x) &= V_1 E(x/l_1) \quad \text{pour tout } x \text{ entier dans } [0, L] \\ F_k(x) &= \max \{F_{k-1}(x), V_k + F_k(x - l_k)\} \quad (16) \end{aligned}$$

Cette forme nous permet de calculer  $F_k(x)$  à partir des valeurs de  $F_{k-1}(x)$  et de  $F_k(y)$  pour  $y < x$ .

Pour calculer la solution optimale, à chaque  $F_k(x)$ , on associe un indicateur  $I_k(x)$  qui est le plus grand indice  $i$  pour lequel  $a_i$  est positif dans la découpe de  $F_k(x)$ .

L'indicateur  $I_k(x)$  est calculé suivant les formules :

$$I(x) = \begin{cases} 0 & \text{si } x < l_1 \\ 1 & \text{si } x \geq l_1 \end{cases} \quad (17)$$

$$I_k(x) = \begin{cases} k & \text{si } F_k(x) = V_k + F_k(x-l_k) \\ I_{k-1}(x) & \text{si } F_k(x) = F_{k-1}(x) \end{cases} \quad (18)$$

Pour calculer  $I_k(x)$ ,  $k=2, \dots, m$ , on a besoin uniquement de  $I_{k-1}(x)$ .

Donc, au lieu de mémoriser le tableau  $a_k(x)$  de dimension  $(L+1) \times m$ , on mémorise seulement deux vecteurs  $I_{k-1}(x)$  et

$I_k(x)$ , tous deux de dimension  $(L+1)$ . Puisque dans (16), on effectue une seule comparaison, cette méthode est donc beaucoup plus rapide que la précédente.

Herz [1972] et Moreau [1973] ont encore amélioré cette méthode :

au lieu d'effectuer les calculs (16) pour tout  $x$  entier,  $x \in [0, L]$ , on se limite à un ensemble  $P$  dont les éléments s'expriment comme combinaisons linéaires à coefficients entiers des longueurs  $l_i$ .

Donc,

$$P = \{ p_k / (a_1, a_2, \dots, a_m), a_i \in \mathbb{N} \text{ pour tout } i; p_k = \sum_{i=1}^m a_i l_i \leq L \}$$

Cette dernière approche a aussi l'avantage de ne pas exiger l'intégrité des  $l_i$ ,  $i = 1, 2, \dots, m$ , comme le fait l'approche précédente.

Le problème de placement de la découpe unidimensionnelle s'applique aussi bien aux problèmes à grande demande qu'à ceux à faible demande. Pour ces derniers, il suffit de remplacer la contrainte (15) par ;

$$r_k \in [0, D] \text{ où } D = \min \{d_k, E(x/l_k)\} \quad (15')$$

et  $d_k$  - la demande pour la kème pièce,

ou de faire des modifications analogues dans les autres formulations.

On voit qu'une demande faible peut faciliter le problème de placement dans ce cas.

Une autre caractéristique intéressante de ce modèle est la facilité avec laquelle on peut modéliser plusieurs autres contraintes du processus physique de découpe comme le nombre limité des couteaux de la machine de découpe, les problèmes de la balance des machines, etc. Par exemple, pour exprimer que le nombre des couteaux est limité par  $S$ , il suffit de tenir compte de la contrainte :

$$\sum_{j=1}^k a_j \leq S$$

chaque fois qu'on calcule une nouvelle valeur  $a_k$ . Cela change très peu les schémas précédents de calcul.

Les méthodes heuristiques pour résoudre ce problème sont celles connues sous le nom de "bin-packing". Elles représentent une alternative, non seulement pour le problème de placement, mais aussi pour le problème (PD). Quelques unes de ces méthodes sont assez performantes en temps de calcul et en pourcentage de perte de matière première (cf. Hu T.C. "Combinatorial Algorithms" Addison Wesley 1981 et aussi Golden [1976]).

### 1.5.2. Découpe bidimensionnelle

Ce problème est beaucoup plus difficile que le précédent. Jusqu'à maintenant, il n'existe pas d'algorithme performant pour le résoudre dans sa forme générale, même pour les cas les plus faciles de découpe des rectangles dans des plaques rectangulaires.

Dans ce paragraphe, on va décrire quelques méthodes pour le placement de rectangles puis, aux chapitres 3 et 4, on étudiera le problème de placement des pièces de forme quelconque.

Le problème de placement de rectangles est résolu efficacement seulement dans des cas particuliers (découpe guillotine à grande demande, découpe en 2 stades, etc.) Pour les autres classes, il existe soit des algorithmes peu efficaces, soit des heuristiques plus ou moins performantes du point de vue informatique (temps de calcul, encombrement mémoire) et de la perte en matière première.

Ce problème est de loin le plus étudié dans la littérature spécialisée, parce qu'il est le plus souvent rencontré en pratique (industrie du verre, du papier, du métal, du plastique, etc.) et aussi parce qu'il représente une bonne approximation pour la découpe d'autres formes dans des rectangles.

Les problèmes de découpe en 2 stades (3 stades, etc.) se ramènent à 2 (3, etc.) problèmes de placement unidimensionnels analogues à celui représenté en 1.5.1. On va donc regarder les méthodes utilisées pour les découpes guillotine générale et non guillotine.

#### 1.5.2.a --- Découpe guillotine à grande demande

Soit :  $R$  de dimensions  $L * W$  la "plaque"

$r_i$  de dimensions  $l_i * w_i$ ,  $i = 1, 2, \dots, m$ , la  $i$ ème pièce (rectangle)

et  $v_i$  sa valeur.

Alors, le problème de placement est le suivant :

trouver  $a_i$ ,  $i = 1, 2, \dots, m$



tels que :

$$\sum_{i=1}^m v_i a_i = Z \text{ (max)} \quad (19)$$

$$a_i > 0 \quad (20)$$

$$a_i \text{ entier} \quad (21)$$

$(a_1, a_2, \dots, a_m)$ , soit obtenue par une découpe guillotine de R (22)

Pour résoudre ce problème, Gilmore et Gomory ont introduit le concept de fonction de knapsack :

Soit  $F(x, y)$ , la valeur de (19) pour la solution optimale du problème (19) - (22) pour une "plaque" de dimension  $x * y$ .

$F(x, y)$  est appelé fonction de knapsack et peut être calculée selon les relations de récurrence suivantes :

$$F(x, y) = \max \{F_0(x, y), F_1(x, y), F_2(x, y)\} \quad (23)$$

où :

$$F_0(x, y) = \max_{\substack{x_i \leq x \\ y_i \leq Y}} v_i \quad (24)$$

$$F_1(x, 0) = 0 \quad (25)$$

$$F_1(x, y) = \max_{x_1, x_2} \{F(x_1, y) + F(x_2, y) / x_1 + x_2 \leq x, 0 \leq x_1 \leq x_2\} \quad (26)$$

$$F_2(0, y) = 0 \quad (27)$$

$$F_2(x, y) = \max_{y_1, y_2} \{F(x, y_1) + F(x, y_2) / y_1 + y_2 \leq y, 0 \leq y_1 \leq y_2\} \quad (28)$$

L'interprétation de (23) est évidente :

- si  $F(x, y) = F_0(x, y)$ , le rectangle  $x * y$  n'est pas coupé.

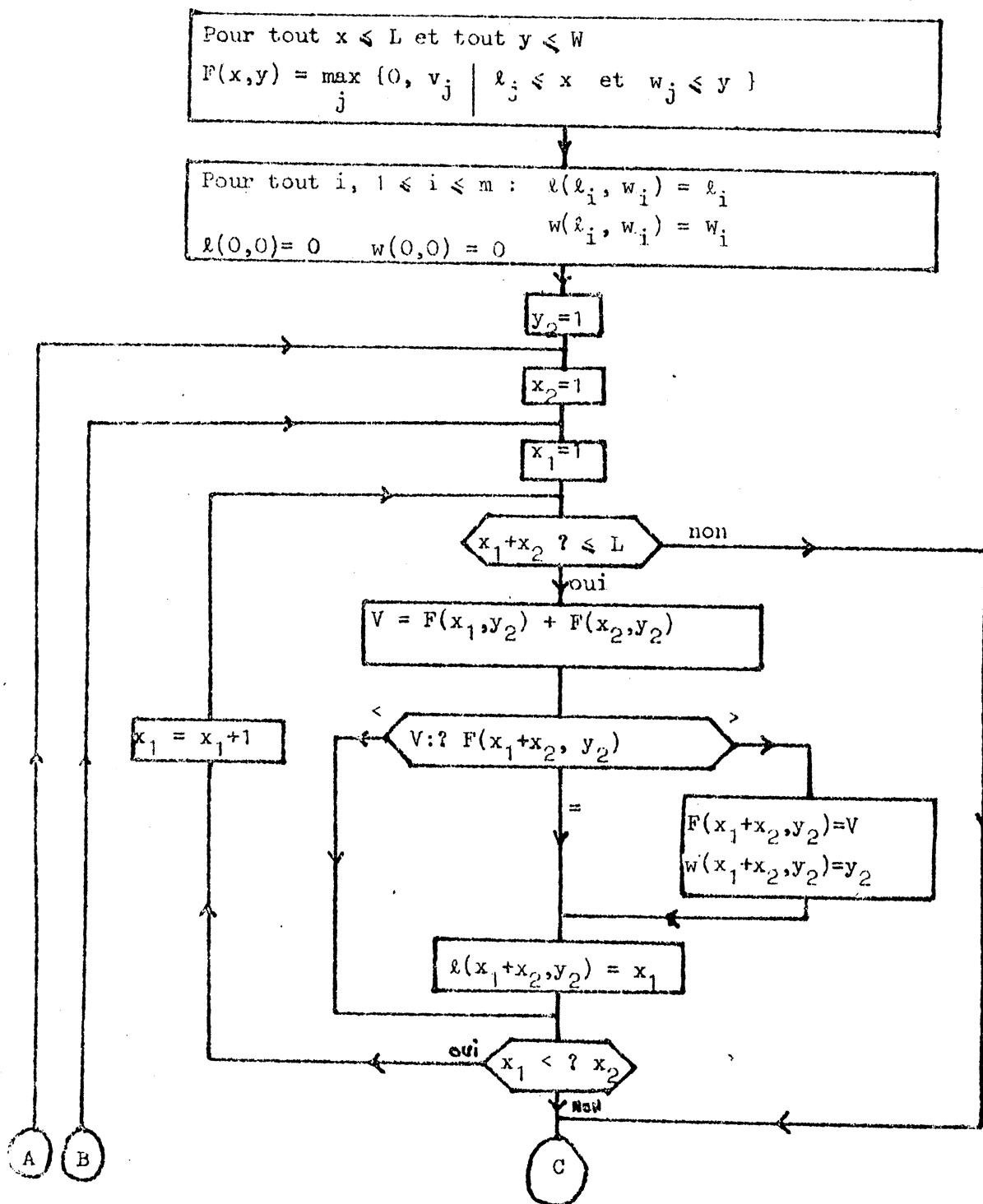
- si  $F(x, y) = F_1(x, y)$ , le rectangle  $x * y$  est divisé en deux rectangles de dimensions  $x_1 * y$  et  $x_2 * y$  par une découpe guillotine parallèle à  $Oy$  en un point d'abscisse  $x_1$ .

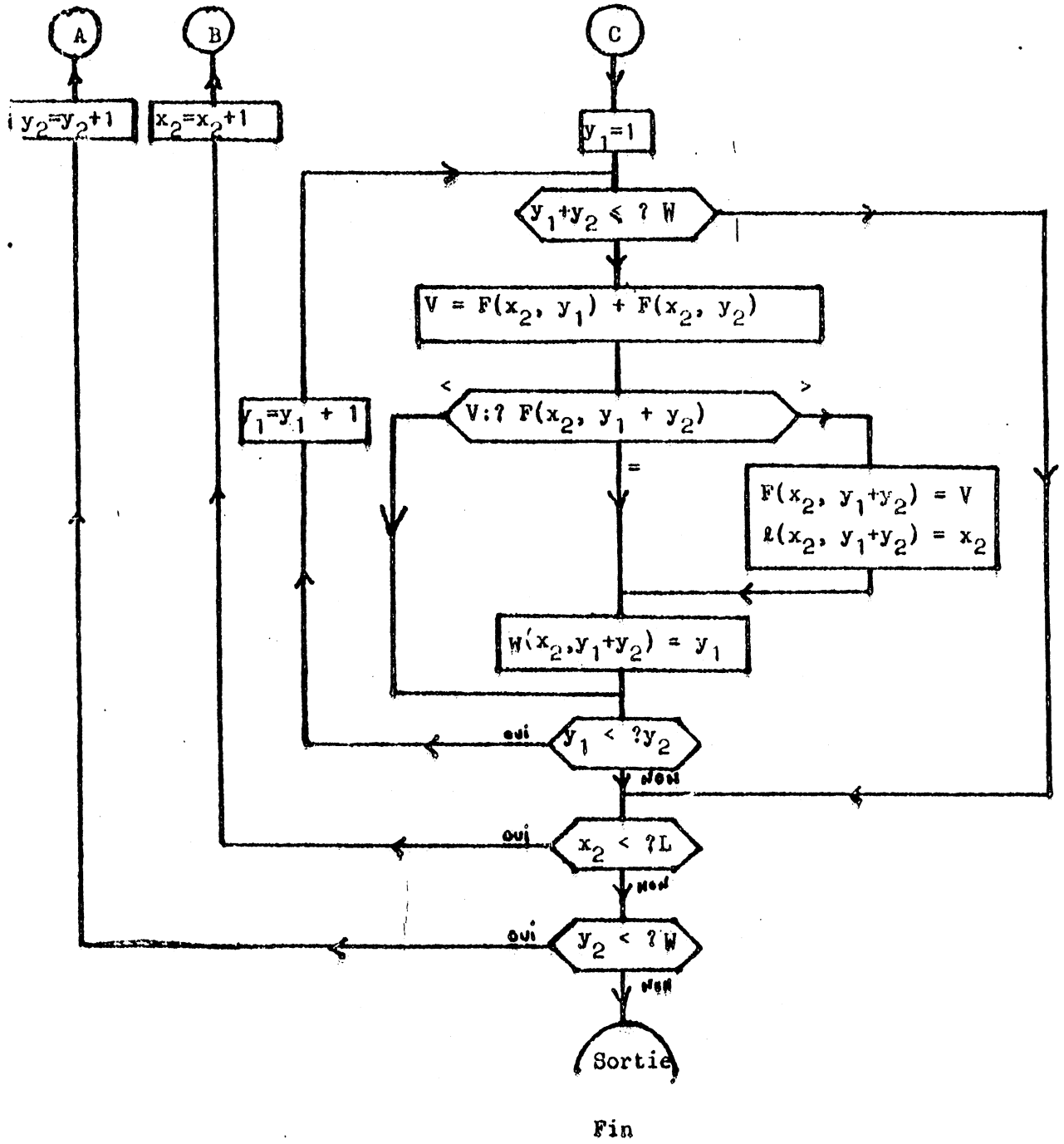
- si  $F(x, y) = F_2(x, y)$ , le rectangle  $x * y$  est

divisé en deux rectangles de dimensions  $x_1 * y$  et  $x * y_2$  par une découpe guillotine parallèle à  $Ox$  en un point d'ordonnée  $y_1$ .

A partir des relations récurrentes (23)-(28), on obtient un algorithme pour résoudre le problème (19)-(22).

L'organigramme de cet algorithme est donné ci-dessous :





Cet algorithme donne la solution optimale du problème de découpe guillotine à grande demande. Mais il demande un temps de calcul et un encombrement mémoire considérable :

- on entre dans la boucle la plus interne environ :  
 $L.W (L + W)/8$  fois, soit, si  $L = W = 2\ 000$ , deux milliards de fois.
- on mémorise au moins deux tableaux  $F(x,y)$  et  $lw(x,y)$  de dimensions  $L \times W$  (les deux tableaux  $l(x,y)$  et  $w(x,y)$  peuvent être stockés en un seul tableau  $lw$  ).

On peut améliorer cet algorithme en temps de calcul et en encombrement mémoire si  $x$  et  $y$  (et donc  $x_1, x_2$  et  $y_1, y_2$  ) prennent leurs valeurs dans des ensembles discrets dont les éléments sont des combinaisons linéaires à coefficients entiers des longueurs (pour  $x$ ) et des largeurs (pour  $y$ ) des rectangles  $r_i$  , comme nous l'avons vu dans 1.5.1. On peut , en plus, améliorer le temps de calcul de  $F(x,y)$  si on utilise une méthode récursive proposée par Herz. Ces améliorations sont présentées en détail dans Herz [1972] et Moreau [1973].

Avantages de l'algorithme Gilmore-Gomory :

- il trouve la solution optimale pour la découpe guillotine,
- si le stock contient plusieurs plaques de dimensions différentes, une fois le problème résolu pour une plaque de dimensions  $L \times W$  où  $L = \max \{L_i\}$  et  $W = \max \{W_i\}$  ( $L_i * W_i$  les dimensions de la  $i$ ème "plaque" du stock), il nous donne la solution optimale pour n'importe quelle "plaque" de dimension  $L' * W'$ ,  $L' \leq L$ ,  $W' \leq W$ .

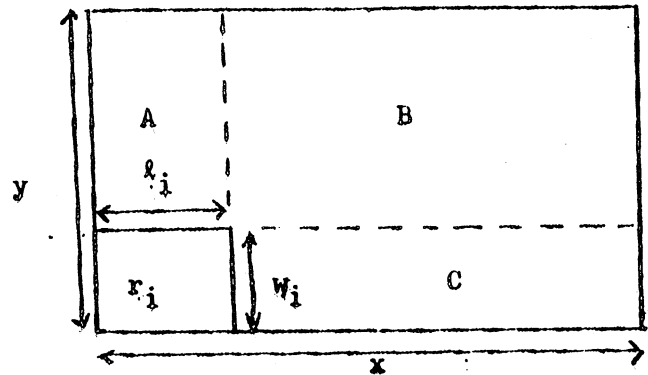
Pour réduire le temps de calcul et la place mémoire utilisée, plusieurs heuristiques ont été inventées. Parmi celles-ci, rappelons une heuristique due à Haims et Freeman [1970] et reprise par Moreau [1973] où pour chaque rectangle  $x * y$ , on cherche à placer une pièce  $r_i$  dans le coin bas-gauche. Alors, par une découpe guillotine fictive, on obtient les rectangles :

$$r_i = l_i * w_i$$

$$A = l_i * (y - w_i)$$

$$B = (x - l_i) * (y - w_i)$$

$$C = (x - l_i) * w_i$$



Pour chaque rectangle A, B, C ainsi que pour les rectangles (A + B) et (B + C), on essaie de placer un rectangle  $r_i$  dans le coin bas-gauche et ainsi de suite.

Soit  $F_k(x,y)$  la valeur de (19) pour la solution optimale du problème (19)-(22) pour la découpe des pièces  $r_1, r_2, \dots, r_k$ , dans une "plaque" de dimensions  $x * y$ .  
Alors :

$F_m(L,W)$  est la solution optimale du problème initial  
et

$$F_i(0,0) = 0 \quad \text{pour tout } i$$

$$F_1(x,y) = \begin{cases} 0 & \text{si } l_1 > x \text{ ou } w_1 > y \\ \max_{a_1} \{v_1 a_1\} & \text{sinon} \end{cases}$$

- si  $l_i > x$  ou  $w_i > y$   
 $F_i(x,y) = F_{i-1}(x,y)$

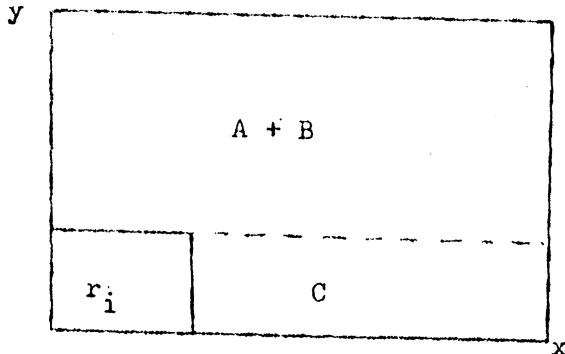
- si  $l_i < x$  et  $w_i < y$

$$F_i(x,y) = \max \begin{cases} F_{i-1}(x,y) \\ v_i + F_i(x-l_i, y) + F_i(l_i, y-w_i) \\ v_i + F_i(x-l_i, w_i) + F_i(x, y-w_i) \end{cases} \quad (29)$$

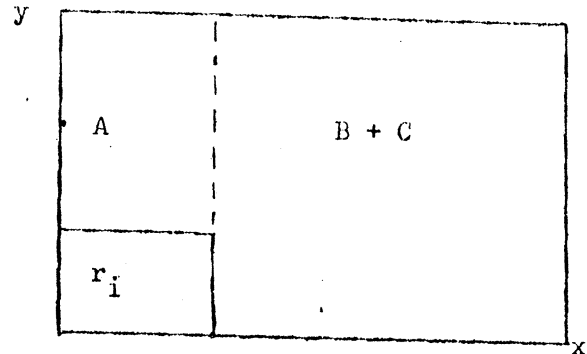
L'interprétation de (29) :

- si  $F_i(x,y) = F_{i-1}(x,y)$ , le rectangle  $x * y$  n'est pas coupé
- si  $F_i(x,y) = v_i + F_i(x-l_i, y) + F_i(l_i, y-w_i)$ , le rectangle  $x * y$  est découpé en  $r_i, A, B + C$ .

-si  $F_i(x,y) = V_i + F_i(x - l_i, w_i) + F_i(x, y - w_i)$ , le rectangle  $x * y$  est découpé en  $r_i, A + B; C$ .



$$F_i(x,y) = V_i + F_i(x-l_i, y) + F_i(l_i, y-w_i)$$



$$F_i(x,y) = V_i + F_i(x-l_i, w_i) + F_i(x, y-w_i)$$

Cette heuristique est en moyenne dix fois plus rapide que celle de Gilmore-Gomory. L'encombrement mémoire est, lui, du même ordre. Les résultats obtenus sont, du point de vue de perte, analogues à ceux obtenus par la précédente, avec un écart de 2 % dans le plus mauvais cas.

Si le rectangle  $r_i$  est placé en un point quelconque à l'intérieur du rectangle  $x * y$ , on obtient une heuristique pour la découpe non guillotine. Dans ce cas, dans les relations analogues à (29), on doit considérer 16 alternatives possibles. A notre connaissance, il n'y a pas de publication sur l'efficacité de cette heuristique pour la découpe non guillotine.

#### 1.5.2.b Découpe guillotine à faible demande

En 1.3, on a présenté l'heuristique de réduction exhaustive répétitive comme une alternative pour les problèmes à faible demande. Pour que cette heuristique soit applicable, le nombre d'exemplaires de chaque pièce placés dans la plaque doit être inférieur à la demande respective (initiale, pour la première plaque et résiduelle pour les autres).

En 1.5.1, on a vu que les problèmes à faible demande

jouent un rôle plutôt positif, en restreignant le domaine à explorer, dans la découpe unidimensionnelle. Ce n'est pas le cas pour les problèmes bidimensionnels, où cette contrainte supplémentaire influence considérablement les algorithmes précédents.

Christofides et Whitlock [1977] proposent un algorithme qui donne la solution optimale dans ce cas.

Leur approche est divisée en deux phases :

- phase 1 :

On ignore momentanément les contraintes de la demande et on résoud le problème en utilisant l'algorithme de Gilmore-Gomory (ou une version améliorée). Si la solution obtenue satisfait aux contraintes de la demande, celle-ci est la solution de problème à faible demande. Sinon, on passe à la seconde phase.

- phase 2 :

Cette phase est une exploration arborescente :

- au rectangle initial  $L * W$  correspond la racine de l'arbre.
- chaque sommet représente l'état du rectangle "père" après la découpe.
- aux rectangles qui ne sont pas coupés ultérieurement, correspondent des sommets pendants.

Une borne supérieure pour un sommet non pendant est donnée par la solution "Gilmore-Gomory" du rectangle respectif (on a vu que l'algorithme de Gilmore-Gomory donne la solution optimale pour tous les sous-rectangles de  $L * W$ ).

Pour les sommets pendants, cette borne est donnée par la solution d'un problème de transport de forme particulière.

Cet algorithme pose des problèmes d'encombrement mémoire et de temps de calcul considérables. Par conséquent, il ne peut être appliqué en pratique que pour des problèmes de petite taille (plaque de petite dimensions et un nombre modéré des pièces). A titre indicatif, pour une plaque de dimensions  $40 * 70$  et 20 pièces, le temps de calcul est 66.14 secondes en un CDC-7600 et le nombre des sommets de l'arbre est 22 184.



Pour les problèmes de grande taille, le seul recours reste les méthodes heuristiques.

Une des heuristiques les plus performantes, tant en temps de calcul qu'en encombrement mémoire, est due à Adamovicz et Albano [1976] et améliorée par Albano et Orsini [1980]. Dans leur approche, ils tiennent compte de l'expérience des placeurs qui, avant de commencer le placement, forment des bandes de plusieurs pièces de largeurs identiques (ou presque). Ensuite, ils les placent dans la plaque.

L'heuristique d'Adamovicz-Albano procède en deux phases :

- phase 1

on construit un nombre  $r$  des bandes candidates au placement. Dans la construction de chaque bande, on tient compte de la demande pour les pièces qui la constituent.

- phase 2

on place les bandes dans la plaque. Pour cela, on résout un problème du sac-à-dos dans la largeur de la plaque. Les relations récursives sont de la forme :

$$G_{i+1}(x) = \max_k \{G_i(x - kw_i + 1) + k \cdot s_{i+1}\}, \quad i = 1, 2, \dots, r \quad (29)$$
$$0 \leq x \leq W, \quad 0 \leq k \leq \min \{b_{i+1}, E(w/w_{i+1})\}$$

où :

$r$  - le nombre de bandes construites

$W$  - la largeur de la plaque

$w_i$  (resp.  $s_i$ ) - la largeur (resp. la valeur) de la  $i$ ème bande.

$b_i$  - la demande pour la  $i$ ème bande (dans le calcul de  $b_i$ , on tient compte de la demande de chaque pièce dans cette bande).

Ce problème est résolu pour la plaque entière ou pour des "sous-plaques". La solution finale est construite à partir des solutions de chaque partie, en utilisant différentes stratégies pour remplir "au-mieux" les portions non occupées de la plaque.

La solution du problème (29) demande l'utilisation d'un tableau de dimension  $W * (r + 3)$ .

Albano et Orsini [1980] ont la même approche, mais pour la résolution du problème (29), ils utilisent un algorithme d'exploration arborescente qui demande un encombrement mémoire de l'ordre de  $O(r)$ .

Les résultats obtenus par ces heuristiques sont très proches des solutions optimales du point de vue perte. En ce qui concerne le temps de calcul et encombrement mémoire, elles sont bien plus performantes que les algorithmes de Gilmore-Gomory (et ses dérivées) et de Christofides-Whitlock. Par exemple, un problème de placement des 398 pièces dans une plaque de dimensions  $12030 * 2550$  est résolu en 0.5 secondes avec 1 % de perte sur un IBM 370/168. Les autres résultats sont du même ordre.

#### 1.5.2.c Découpe non guillotine

A notre connaissance, il n'existe pas d'algorithme pratique pour la solution optimale de ce problème. En 1.5.2, on a vu une heuristique qui peut s'adapter à ce cas, mais elle est très chère en temps de calcul et en encombrement mémoire

Bengtsson [1982] donne une heuristique pour ce problème qui est applicable lorsque les pièces sont relativement larges par rapport à la plaque et que l'on place, donc, dans celle-ci très peu de pièces.

A titre indicatif, pour un problème de 40 pièces à placer dans une plaque de dimensions  $40 * 25$  une solution avec 5 % de perte est obtenue en 4 secondes et avec 1 % de perte en 56 secondes.

Pour la découpe non guillotine, une heuristique souvent appliquée est l'adaptation d'une des méthodes de découpe guil-

lotine. La perte supplémentaire introduite par cette approximation est largement acceptée devant les dépenses supplémentaires en temps de calcul qu'une méthode spéciale entraînerait.

### 1.6 Conclusions

Le problème de découpe se formule mathématiquement comme un programme linéaire en nombres entiers avec un très grand nombre de variables. Cela rend très difficile la solution directe.

Deux des techniques les plus utilisées pour la résolution sont la génération de colonnes et la réduction exhaustive répétitive. Indépendamment de la méthode de résolution, on rencontre toujours le problème de placement, qui dans les cas des pièces régulières se ramène dans un (plusieurs) problème(s) de sac-à-dos uni-ou-multi-dimensionnel(s).

Le problème de placement est la partie la plus importante, la plus difficile et la plus coûteuse en temps de calcul du problème de découpe.

## CHAPITRE II

### 2.0. Introduction

La mécanique statistique est la discipline qui étudie le comportement de la matière à partir de l'étude des atomes qui la constituent. La matière est considérée comme un système d'atomes qui se déplacent aléatoirement suivant différentes lois internes.

A cause du très grand nombre d'atomes, c'est la conduite la plus probable que l'on observe lors d'un équilibre thermique. Cette conduite est la moyenne de toutes les petites fluctuations des atomes autour de leur position d'équilibre.

Les atomes forment différentes configurations. A chaque configuration  $i$ , correspond une énergie interne  $E_i$ . Les différentes configurations, à une température donnée  $T$ , suivent une distribution de Boltzmann : la probabilité d'observer la configuration  $i$  est proportionnelle à  $\exp(-E_i/k.T)$  où  $k$  est la constante de Boltzmann.

Quelques méthodes de mécanique statistique présentent des analogies avec les méthodes itératives utilisées en Optimisation Combinatoire. C'est justement la simulation d'une de ces méthodes qui va nous donner une heuristique générale et prometteuse pour une vaste classe de problèmes d'Optimisation Combinatoire.

### 2.1. La méthode de recuit

Un des problèmes clés de la mécanique statistique est l'étude du comportement du système à basses températures. On regarde alors si les atomes restent fluides ou se solidifient et, dans ce dernier cas, s'ils forment une structure cristalline ou un verre.

L'énergie interne du système a plusieurs "puits énergétiques", c'est à dire plusieurs minima locaux. La structure

cristalline, état régulier et stable, correspond aux "puits énergétique" le plus profond et représente le minimum absolu de la fonction d'énergie, tandis que les verres, états métastables, correspondent aux "puits énergétiques" quelconques et représentent des minima locaux de la fonction d'énergie.

La méthode de recuit est un procédé expérimental pour trouver l'état stable du système. Au départ, on chauffe le système jusqu'à ce qu'il devienne liquide. Ensuite on baisse doucement la température en restant très longtemps dans les températures voisines du point de gel. Il faut rester assez longtemps à chaque température intermédiaire de façon à ce que le système trouve l'état d'équilibre de la température en question. Sinon, le système risque de sortir de l'état d'équilibre et on trouvera une structure cristalline non régulière ou bien un verre.

## 2.2 La méthode de Metropolis

C'est une méthode relativement simple, présentée par Metropolis et al. [1953], pour simuler une collection d'atomes en équilibre à une température donnée. Ici on va décrire ses idées principales pour faire la liaison avec ce qui suit. Dans cette méthode la matière est supposée constituée de carrés élémentaires, tous identiques, chaque carré contenant un nombre  $N$  d'atomes distribués aléatoirement.

Si on connaît les positions des atomes dans le carré, on peut calculer facilement l'énergie potentielle du système:

$$E = 1/2 \sum_{i=1}^N \sum_{j=1}^N V(d_{ij}) \quad (2.1)$$

où  $V(d_{ij})$  est le potentiel entre les atomes  $i$  et  $j$  qui se trouvent à distance  $d_{ij}$ .

On peut alors calculer la valeur à l'équilibre,  $\bar{F}$ , d'une quantité  $F$  d'après la formule : (2.2)

$$\bar{F} = \frac{\int F \exp(-E/kT) d^{2n}_p d^{2n}_q}{\int \exp(-E/kT) d^{2n}_p d^{2n}_q}$$

où  $d^{2n_p} \cdot d^{2n_q}$  représente le volume élémentaire dans l'espace à  $4N$ -dimensions.

La formule (2.2) représente une intégration dans l'espace à  $4N$ -dimensions \*). Etant donné que le nombre  $N$  d'atomes est de l'ordre de quelques centaines, la seule méthode pratique, pour effectuer les calculs, reste la méthode de Monte-Carlo :

On génère une position aléatoire pour chaque atome dans le carré, on calcule, à l'aide de (2.1), l'énergie de cette configuration et on pondère cette configuration avec le poids  $\exp(-E/kT)$ .

On répète ces calculs et on trouve ;

$$\bar{F} = \frac{\sum_{i=1}^m F_i}{m} \quad (2.3)$$

où ;  $m$  est le nombre des essais effectués et  $F_i$  est la valeur de la fonction  $F$  au  $i$ ème essai.

Cependant, pour des configurations assez denses ( $E$ -grand), on choisit, avec une grande probabilité, une configuration où  $\exp(-E/kT)$  est très petit et cela peut être la source de plusieurs anomalies pendant les calculs.

Pour surmonter ces difficultés, Metropolis et al. [1953] ont utilisé un schéma Monte Carlo modifié : au lieu de choisir les configurations uniformément et de les pondérer, ensuite, avec le poids  $\exp(-E/kT)$ , on les choisit avec une probabilité  $\exp(-E/kT)$ , puis on les pondère toutes avec un poids égal à 1.

Le fonctionnement de ce schéma modifié est très analogue au précédent :

On choisit une configuration initiale et on déplace chaque atome d'après les formules

$$\left. \begin{aligned} x_n &= x_a + d \cdot r_1 \\ y_n &= y_a + d \cdot r_2 \end{aligned} \right\} \quad (2.4)$$

---

\*) En effet, après quelques simplifications, les intégrales en (2.2) peuvent être séparées en intégrales dans l'espace à  $2N$ -dimensions.

où:

- $(x_a, y_a)$  - les anciennes coordonnées de l'atome
- $(x_n, y_n)$  - les nouvelles coordonnées de l'atome
- $d$  - le pas maximal de déplacement
- $r_1, r_2$  - nombres aléatoires uniformément distribués sur  $(-1,1)$

On calcule alors le changement d'énergie du système

$$\Delta E = E_n - E_a$$

où  $E_n$  ( $E_a$ ) - l'énergie du système dans la nouvelle (ancienne) configuration.

Si  $\Delta E \leq 0$ , le déplacement est accepté et l'atome est effectivement déplacé en  $(x_n, y_n)$ .

Sinon, le déplacement est accepté avec la probabilité  $\exp(-\Delta E/kT)$  : on génère un nombre aléatoire  $r_3$  uniformément distribué sur  $(0,1)$  et si  $r_3 < \exp(-\Delta E/kT)$  le déplacement est accepté, sinon, il est rejeté et l'atome reste en  $(x_a, y_a)$ .

D'une façon plus formelle, l'atome est déplacé en  $(x,y)$  avec

$$(x,y) = \begin{cases} (x_a, y_a) & \text{si } \Delta E > 0 \text{ et} \\ & r_3 \geq \exp(-\Delta E/kT) \\ (x_n, y_n) & \text{sinon} \end{cases}$$

Dans les deux cas,  $(x,y)$  est considéré comme une "nouvelle" position pour effectuer le calcul de  $\bar{F}$ .

Metropolis et al. démontrent que cet algorithme choisit les différentes configurations avec la probabilité  $\exp(-E/kT)$ . Cela nous garantit que le système évolue suivant une distribution de Boltzmann.

### 2.3. La méthode de Kirkpatrick

Le problème de trouver les structures cristallines des substances est, en fait, un problème d'Optimisation Combinatoire. On dispose d'un nombre fini d'atomes (molécules, particules, etc.) qui peuvent avoir des configurations différentes et il faut trouver la configuration d'énergie interne minimum.

Le procédé expérimental de recuit arrive à trouver le minimum absolu de la fonction d'énergie interne, tout en évitant un grand nombre des minima locaux. Dans sa réalisation pratique, ce procédé ressemble beaucoup aux méthodes itératives d'Optimisation Combinatoire. C'est cette analogie qui a attiré l'attention de Kirkpatrick et al. pour inventer une méthode heuristique, pour des problèmes d'Optimisation Combinatoire, qui simule le procédé de recuit.

Dans cette méthode, la fonction économique joue le rôle d'énergie. Pourtant dans les problèmes d'Optimisation Combinatoire il n'y a pas de paramètre analogue à la température du procédé de recuit. La température est le paramètre le plus important de ce procédé. Elle permet au système de sortir des "puits locaux", puisque, à chaque  $T > 0$  il existe une probabilité  $p = \exp(-\Delta E/kT)$  d'accepter des "détériorations" de la configuration du système. L'absence d'un tel paramètre dans les méthodes itératives d'Optimisation Combinatoire se manifeste par le fait que celles-ci donnent le "premier" minimum local rencontré dans leur "parcours", même s'il est très éloigné du vrai minimum. De ce point de vue, ces méthodes sont équivalentes au procédé de trempe si d'une température élevée on passe directement à la température  $T = 0$ .

Pour simuler le procédé de recuit Kirkpatrick et al. [1982] ont introduit, dans l'algorithme glouton, un paramètre équivalent à la température. Ce paramètre n'a pas de signification physique. Il est seulement un paramètre de contrôle exprimé en mêmes unités que la fonction économique.

Avec l'introduction de la température cette méthode se résume ainsi :



On amène tout d'abord le système à l'état "liquide" c'est à dire qu'on part d'une très "haute température". Cette "haute température" dépend du problème combinatoire en question, plus précisément, de l'ordre de grandeur de la fonction économique et du "mécanisme" de génération des nouvelles configurations.

A cette température, le système évolue librement : n'importe quel changement dans la configuration du système est accepté.

Ensuite, on baisse doucement la température et à chaque niveau on applique la méthode de Metropolis pour trouver la configuration d'équilibre correspondante.

De manière formelle, l'algorithme est le suivant :

(0) Initialisation :

On part d'une configuration initiale (déterminée aléatoirement ou déjà connue). On calcule l'énergie interne initiale  $E_0$  et la température de départ  $T_0$ .

(1) Itération :

Le système se trouve à la température  $T$  et possède l'énergie  $E$ .

(1a) On génère une nouvelle configuration, on calcule l'énergie  $E_1$  et  $\Delta E = E_1 - E$ .

(1b) Si  $\Delta E > 0$  et  $r > \exp(-\Delta E/T)$ , où  $r$  uniforme sur  $(0,1)$ , on rejette la nouvelle configuration. Aller en (2).

(1c) Sinon, le système passe dans la nouvelle configuration,  $E = E_1$ .  
Aller en (2) .

(2) Equilibre pour la température  $T$  :

Si la position d'équilibre pour la température  $T$  est trouvée aller en (3), sinon, aller en (1).

(3) Solution optimale :

Si on a trouvé la solution optimale STOP. Sinon, passer à la température suivante et aller en (1).

2.4 Discussion sur la méthode de Kirkpatrick

Les méthodes pour résoudre les problèmes d'Optimisation Combinatoire se divisent en deux classes :

1. Méthodes itératives, qui, à partir de quelques règles de transformation, génèrent différentes configurations du système et gardent la "meilleure".

2. Méthodes de séparation qui divisent le problème initial en plusieurs sous-problèmes faciles à résoudre. La solution finale se construit à partir des solutions de ces sous-problèmes.

La méthode de Kirkpatrick, bien que méthode itérative en principe, présente certaines caractéristiques des méthodes de séparation. Dans les premières étapes de l'algorithme on observe la formation du "squelette" de l'état stable, c'est à dire la formation des groupes d'éléments tandis que dans les étapes suivantes, on observe la "finition" de cet état, c'est à dire des arrangements locaux dans ces groupes.

Cela représente la caractéristique la plus importante de cette méthode.

Pour mettre en oeuvre celle-ci, on a besoin :

1. D'une description concise des configurations du système.

Elle dépend du système concret qu'on étudie.

2. D'une fonction économique qui exprime quantitativement l'évolution du système.

3. D'un ensemble de "transformations" pour générer les différentes configurations du système.

Les méthodes itératives existantes peuvent très bien suggérer un ensemble de "bonnes" transformations.

4. D'une liste de températures pour le procédé de recuit. On a vu précédemment les critères qui nous aident à initialiser la température  $T_0$ . Les autres températures sont choisies d'après une formule de la forme

$$T_n = u^{n*} T_0 \quad (0 < u < 1)$$

afin de rester plus longtemps à basses températures, comme le suggère le procédé expérimental de recuit. D'autres formules de transformation ne sont pas exclues.

5. Des critères pour reconnaître l'équilibre à une température donnée.

Kirkpatrick et al. [1982] suggèrent ce critère : à chaque température on reste jusqu'à ce qu'un nombre  $N_a$  des configurations acceptées soit atteint ou bien un nombre  $N_t$  d'itérations soit dépassé. Les nombres  $N_a$  et  $N_t$  dépendent du nombre  $n$  des variables du problème.

6. Des critères d'arrêt

Kirkpatrick et al. proposent ce critère :

Le système est considéré gelé si le nombre  $N_a$  n'est pas atteint pour 3 températures successives. Le système concret peut suggérer des critères plus efficaces.

Un autre critère peut être suggéré par la mécanique statistique. Là, la quantité

$$C(T) = d E(T)/dT = ( \langle E(T)^2 \rangle - \langle E(T) \rangle^2 ) / kT^2$$

est appelée chaleur spécifique. Une grande valeur de  $C(T)$  signale un changement de l'état d'ordre du système, tandis que des valeurs proches de zéro signalent que le système est proche d'un état d'équilibre.

Le procédé expérimental de recuit est appliqué pour des

objets identiques (atomes, particules, etc.). Dans les problèmes d'Optimisation Combinatoire ce n'est pas toujours le cas et cela peut représenter un handicap pour la méthode. Pour diminuer celui-ci, soit on construit dès le départ des "macro-atomes" "identiques", soit on introduit des transformations qui lissent en quelque sorte cette différence.

## 2.5. Applications pratiques de la méthode de Kirkpatrick

Cette méthode est appliquée avec succès dans différents problèmes d'Optimisation Combinatoire, principalement dans ceux que l'on rencontre en "conception d'ordinateurs".

### 2.5.1. Problème de partitionnement des circuits intégrés

Le problème pratique s'exprime ainsi :

Comment partitionner  $N$  circuits intégrés entre deux "puces" de manière à ce que le nombre des signaux qui passent d'une puce à l'autre soit minimum.

Soit  $\{a_{ij}\}$  le nombre des signaux qui passent entre le circuit  $i$  et le circuit  $j$  et  $M_i = \pm 1$  une variable qui indique la position du circuit  $i$  dans l'une ou l'autre puce. Alors le nombre des signaux qu'échangent les deux puces est :

$$N_c = \sum_{i>j} (a_{ij}/4) (M_i - M_j)^2$$

Pendant, en pratique, il est préférable d'avoir à peu près le même nombre de circuits dans les deux puces. La différence entre ces deux nombres est

$$N_d = \sum_{i=1}^N M_i$$

En pondérant  $N_d$  avec un coefficient  $p$ , on obtient la fonction économique à minimiser :

$$f = N_c + pN_d = 4p + \sum_{i>j} a_{ij}/2 + \sum_{i>j} (p - a_{ij}/2)M_iM_j$$

$$\text{ou } f = C + \sum_{i>j} (p - a_{ij}/2) M_i M_j \quad (2.5)$$

Le poids  $p$  doit être choisi inférieur ou égal à  $Z/2$  ou  $Z$  est le nombre moyen de circuits reliés à un circuit type.

La valeur  $p \approx Z/2$  indique que le changement de  $N_d$  d'une unité est équivalent au changement de  $N_c$  avec  $Z/2$ , c'est à dire avec le passage d'une puce à l'autre du circuit moyen représentatif des circuits présents.

La fonction économique (2.5) représente des situations pratiques très simplifiées mais on peut construire des fonctions analogues à (2.5) qui représentent la réalité avec beaucoup plus de fidélité. Néanmoins, cela ne change pas le déroulement de la méthode.

Celle-ci est appliquée par Kirkpatrick et al. pour partitionner en 2 puces le Microprocesseur unipuce 370 d'I.B.M., contenant environ 5 000 portes logiques primitives. La puce originale a 200 sorties. Le changement de configuration du système s'exprime par le passage d'un circuit (choisi au hasard) d'une puce à l'autre.

En prenant  $\lambda = 0.01$ ,  $T_0 = 10$  ("très haute température" pour ce système) et  $T_n = (0.9)^n T_0$ , ils ont trouvé une partition avec 353 sorties dans une puce et 321 dans l'autre.

En utilisant pour  $N_d$  une formule, qui ne pénalise pas les différences inférieurs à 100 circuits ( $p = 0$  si  $N_d < 100$ ), ils ont trouvé une solution avec respectivement 271 et 183 sorties.

L'algorithme glouton donne une partition avec environ 700 sorties en chaque puce.

### 2.5.2. Problème du voyageur de commerce

Kirkpatrick et al. ont utilisé cette méthode pour le problème du voyageur de commerce avec  $N$  villes. Chaque tour

(configuration du système) est représenté par une permutation de nombres de 1 à N. Le système passe d'une configuration à l'autre d'après la transformation suivante :

On enlève une section du tour et on l'introduit entre deux villes consécutives choisies au hasard. La section enlevée peut être inversée.

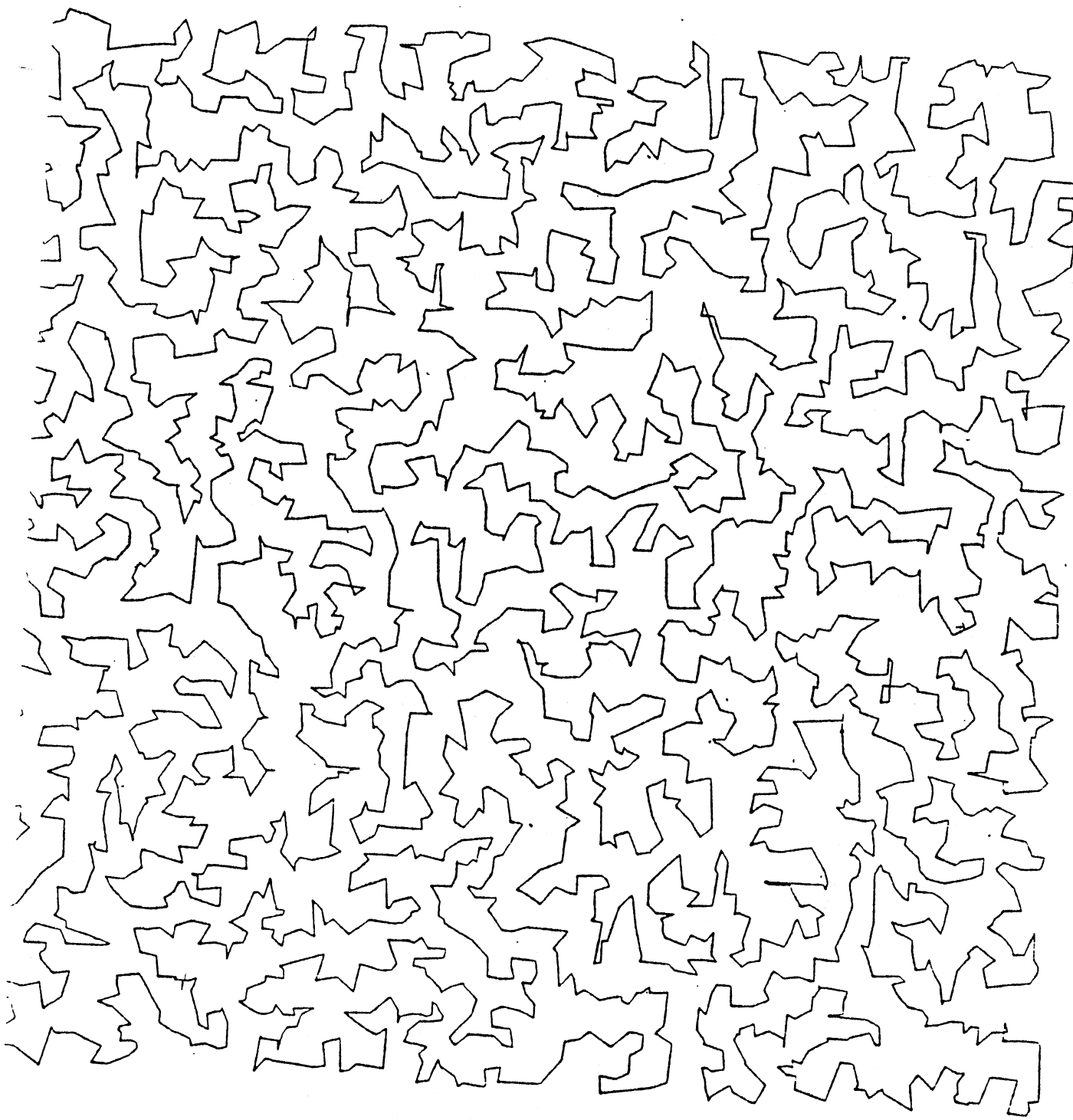
La température initiale est de l'ordre  $N^{1/2}$ .

A l'aide de cette méthode, on trouve une solution optimale locale pour un problème avec 3 200 villes. Le temps de calcul est de l'ordre de  $N \cdot \ln(N)$ . Les méthodes qui donnent une solution optimale exigent un temps de calcul proportionnel à  $\exp(N)$ . Même si le problème avec 3 200 villes est d'une structure particulière (N points choisis au hasard dans un carré) ce résultat paraît très intéressant. La solution de ce problème est présentée fig 1. La fig 2 illustre un problème de voyageur de commerce avec 400 villes à différentes températures.

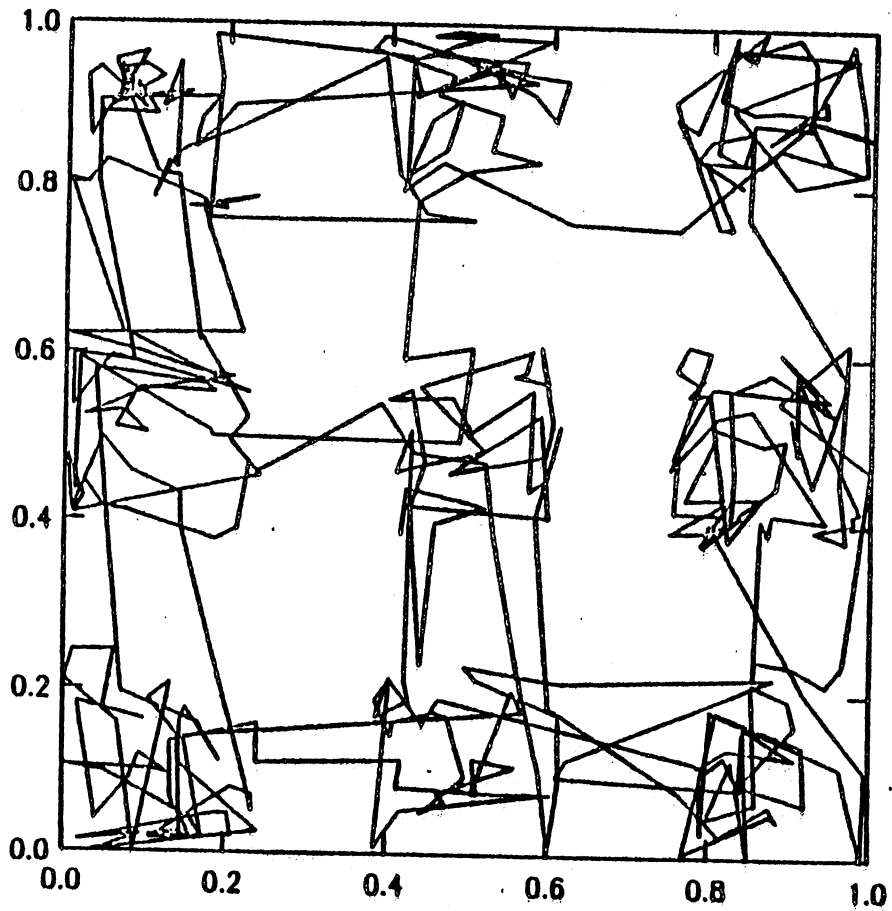
Dans Kirkpatrick et al [1982] on trouve aussi une application pour le placement physique des circuits dans une puce et une autre sur le problème de câblage dans une puce.

## 2.6. Conclusions

L'analogie entre le procédé expérimental de recuit et les méthodes itératives d'Optimisation Combinatoire donne lieu à une méthode heuristique qui est appliquée avec succès dans plusieurs problèmes pratiques. La méthode, un schéma Monte Carlo modifié, est d'une simplicité remarquable et peut être adaptée facilement pour une vaste classe de problèmes d'Optimisation Combinatoire. Pour le moment, l'expérimentation pratique est assez limitée. Néanmoins, le fait que cette méthode soit assez générale (pas de limitations sur la fonction économique, etc.) justifie l'intérêt pour une étude plus approfondie des applications possibles. Les expérimentations pratiques réalisées montrent la supériorité de cette méthode par rapport à l'algorithme glouton.

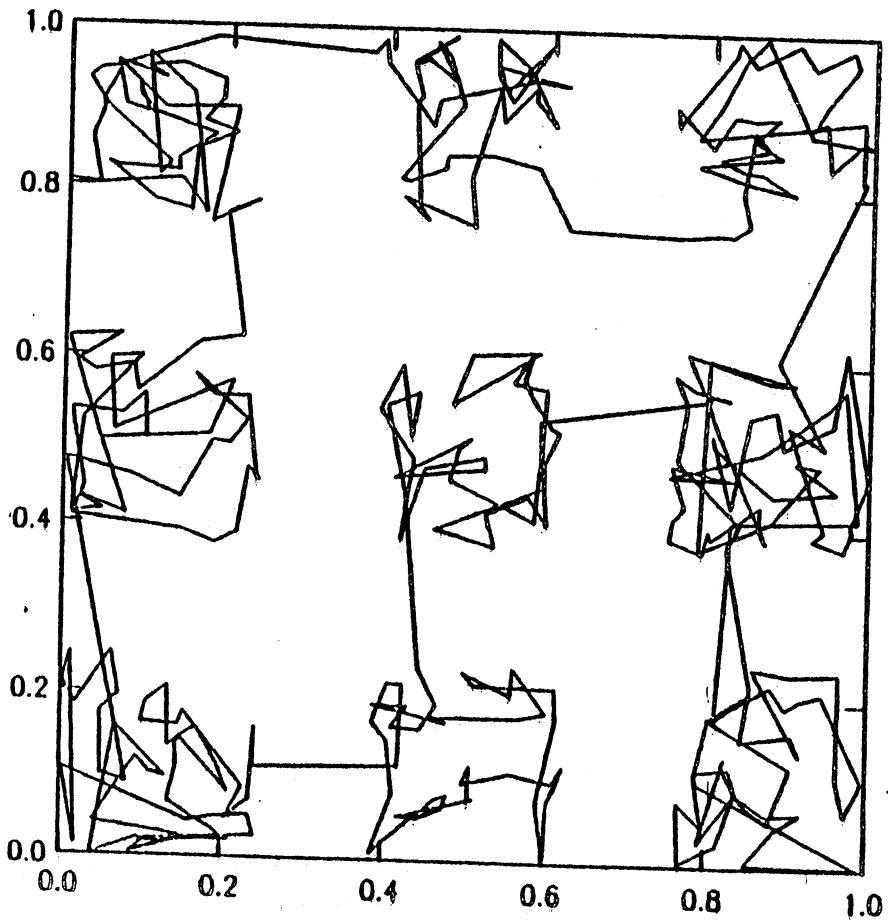


3200 POINT TRAVELLING SALESMAN SOLUTION WITH  $\rho = .93117$ .  
fig. 1 (extrait de Kirkpatrick [1982])



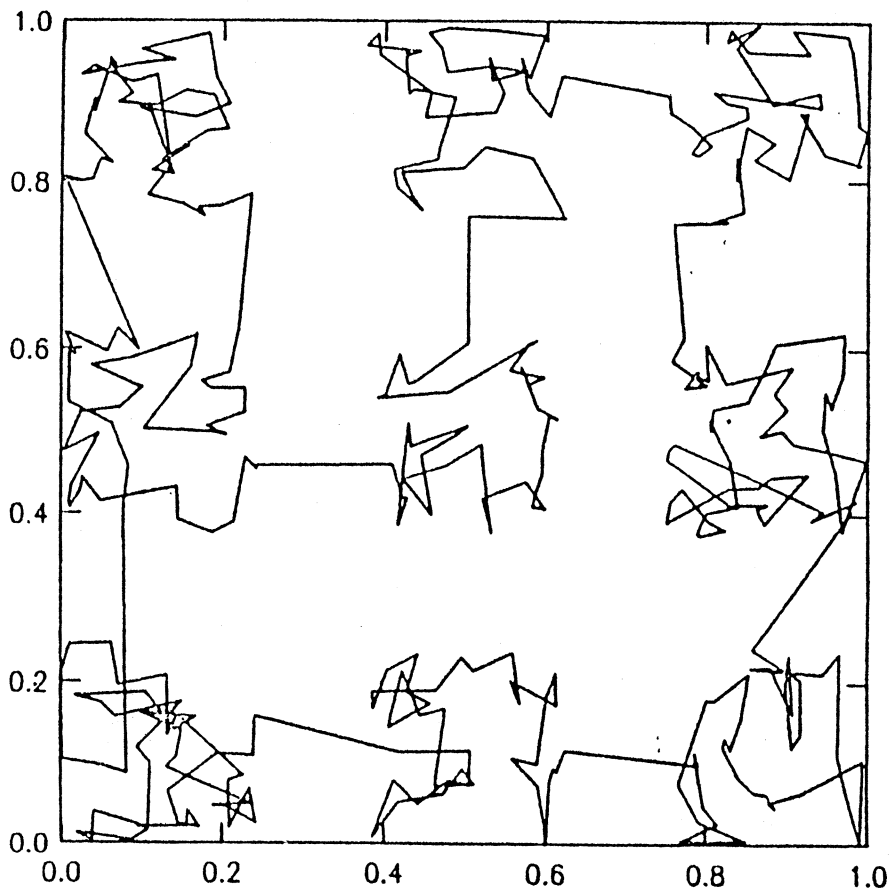
400 sites  
 $T = 1.2$   
 $\alpha = 2.0567$

a



400 sites  
 $T = 0.5$   
 $\alpha = 1.515$



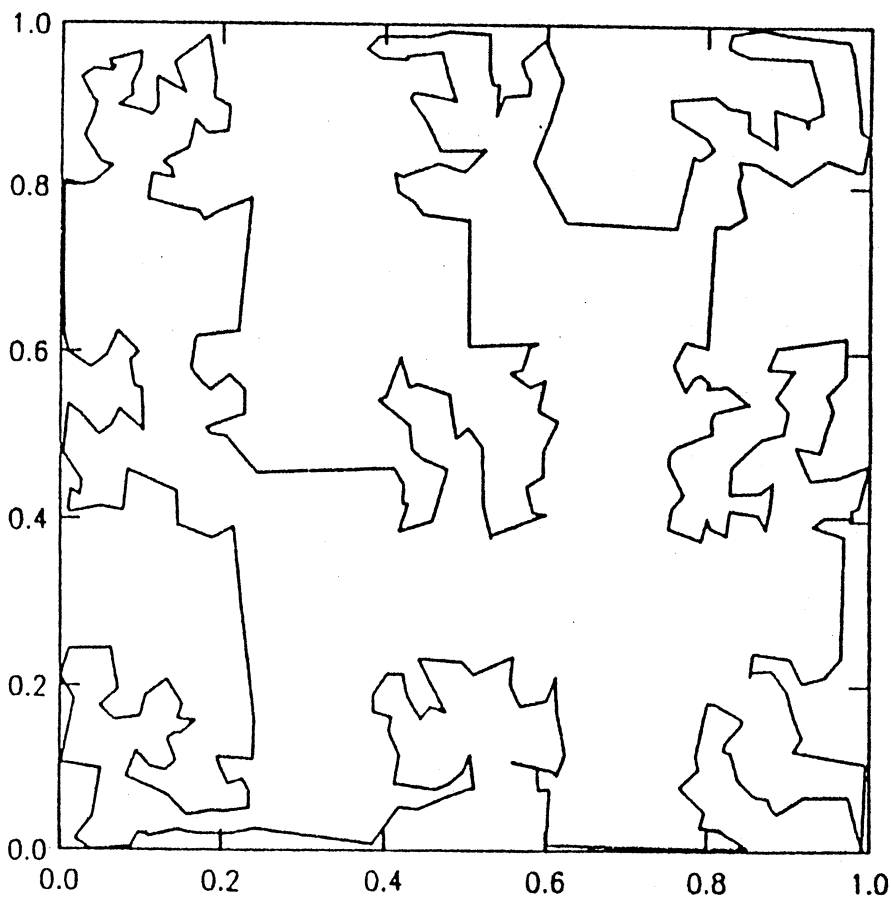


400 sites

$T = 0.4$

$\alpha = 1.055$

c



400 sites

$T = 0$

$\alpha = .7834$

d

Fig 2 a, b, c, d, Différentes phase du problème de voyageur de commerce avec 400 villes (extrait de Kirkpatrick)

### CHAPITRE III

#### 3.0. Introduction

Comme nous l'avons vu au chapitre 1, le problème de placement est la partie centrale du problème de découpe. Ce problème, pour les pièces de forme quelconque, est très peu traité dans la littérature. D'après Hinxman [1980], qui cite Abraham et al \*), "le nombre de publications consacrées au problème de découpe est très limité parce que beaucoup de modèles, procédures et programmation sont développés pour être commercialisés et par conséquent sont confidentiels."

Les méthodes de placement des pièces de formes quelconques se divisent en deux classes :

##### 1. Méthodes stochastiques

Dans ces méthodes, on "jette" les pièces aléatoirement sur la plaque. Pour chaque pièce on effectue un grand nombre d'essais (quelques centaines ou quelques milliers) et à partir de certains critères locaux, on garde la meilleure position de la pièce. Auparavant, les pièces sont réparties en classes de priorité et on choisit en premier les pièces appartenant à la classe de plus forte priorité.

Ces méthodes sont discutées dans Böhme et Graham [1979].

##### 2. Méthodes qui se ramènent au placement de rectangles

Dans ces méthodes, on construit, premièrement, des modules rectangulaires contenant un ou plusieurs exemplaires d'une ou plusieurs pièces. On ne garde alors que les "bons" modules, c'est à dire ceux dont la perte est inférieure à un seuil fixé a priori. Chaque pièce doit figurer au moins une fois dans un de ces modules.

---

\*) Abraham P.M., Kirby S.J. et Ng Y.G.

"Production schedule of a float glass process plant"  
Systems Engineering Science and Industrial Management.  
University of Liverpool, 1976

Ensuite, on applique une des méthodes appropriées pour le placement des rectangles (cf 1.5).

Cette approche est décrite dans Adamovicz et Albano [1976], Albano et Sapuppo [1980], Haims et Freeman [1970], Uhry et Kuntz [1981], Le Borgne et Martin Martin [1982], etc.

On va maintenant proposer une méthode de placement des pièces irrégulières, basée sur la méthode de simulation du processus de recuit.

### 3.1. Le problème à résoudre

On dispose d'un nombre de pièces irrégulières et on veut les placer dans une "plaque" de longueur variable (problème de découpe d'une dimension et demi).

Notre objectif est de les placer sans chevauchement dans une "plaque" de longueur minimum.

En pratique, la "plaque" considérée est toujours de dimension fixe. Sa longueur est estimée à partir de la perte qu'on veut obtenir. Si à la fin, les pièces "n'entrent pas" dans cette "plaque", celle-ci est prolongée. Ceci dit, le seul chevauchement permis est celui avec le bord droit de la "plaque".

Les cas où la "plaque" est fixe où le nombre de pièces à placer n'est pas connu à l'avance sont traités plus loin (cf 3.6.4.).

### 3.2. L'idée principale de la méthode "thermodynamique"

Pour résoudre le problème énoncé ci-dessus on va adapter une méthode (appelée la méthode "thermodynamique") basée sur les idées de la méthode de Kirkpatrick.

Ici, le système est la collection de pièces à placer. Une configuration de ce système est une manière de disposer les pièces à "proximité" de la "plaque", c'est à dire à une distance limitée du bord le plus proche de la "plaque". Un

répertoire de transformations (choisi soigneusement) permet au système de passer d'une configuration à une configuration voisine. On va définir une fonction d'énergie sur l'ensemble des configurations. Cette fonction va être choisie de manière que la configuration d'énergie minimum soit la solution du problème ci-dessus.

### 3.3. La mise en oeuvre de la méthode thermodynamique

#### 3.3.1. Description d'une configuration du système

La "plaque" est rectangulaire et de dimensions  $L * W$  connues.

Pour décrire le système, on introduit un système de coordonnées cartésiennes dont l'origine est le point bas-gauche de la "plaque". Chaque pièce est repérée par le quintuplé  $(x, y, s_x, s_y, \text{angle})$ , où :

$(x, y)$  - les coordonnées du coin bas-gauche de son rectangle circonscrit

$s_x$  (resp.  $s_y$ ) - symétrie de la pièce par rapport à  $Ox$  (resp.  $Oy$ ) dans la configuration courante. D'après le problème concret, certaines symétries sont permises et certaines non.  $s_x$  (resp.  $s_y$ ) indique si la symétrie permise est effectuée ( $s_x = 1$ ) ou non ( $s_x = 0$ ).

angle - angle de la rotation de la pièce par rapport à une direction de base de la pièce.

#### 3.3.2. La fonction d'énergie

Pour tenir compte des objectifs du problème à résoudre (cf 3.1.) on a choisi une fonction d'énergie de la forme :

$$\sum_{i=0}^m \sum_{j=i+1}^m \text{chev}_{ij} + p_0 \sum_{i=1}^m |x_i| w_i \quad (1)$$

où :

$m$  - le nombre de pièces à placer ;

la "plaque" est considérée comme la pièce 0

$chev_{ij}$  - le chevauchement entre les pièces  $i$  et  $j$ . Il est calculé selon les formules :

$$chev_{ij} = \begin{cases} p_1 S_{ij} & \text{pour } i \neq j, i \neq 0, j \neq 0 \\ p_2 s_{ij}^1 + p_3 s_{ij}^2 & \text{pour } i = 0, j \neq 0 \end{cases}$$

$S_{ij}$  - la surface de chevauchement des pièces  $i$  et  $j$

$s_{0j}^2$  - La surface de la pièce  $j$  qui dépasse la plaque en longueur à droite mais qui ne dépasse pas la "plaque" en largeur

$s_{0j}^1$  - la surface de la pièce  $j$  qui dépasse la "plaque" diminuée de  $s_{0j}^2$   
 ( $s_{ij}$ ,  $s_{0j}^1$  et  $s_{0j}^2$  sont illustrés en fig 3.1.)

$p_0, p_1, p_2, p_3$  - le poids de chaque chevauchement  
 ( $p_2 > p_1 > p_3$ ) )

$w_i$  - la largeur du rectangle circonscrit à la pièce  $i$

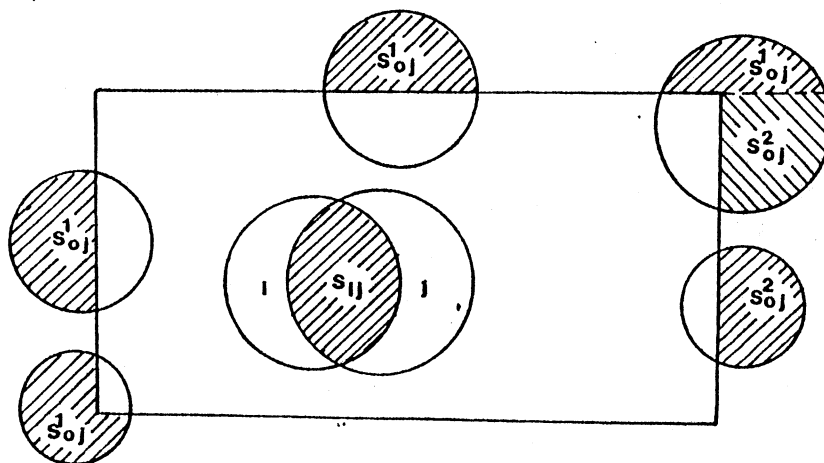


Fig. 3.1.

Le premier terme de la fonction d'énergie représente "l'énergie d'interaction" entre les pièces tandis que le deuxième, "l'énergie de position" de la pièce indépendamment des pièces qui l'entourent.

Le premier favorise l'éloignement réciproque des pièces qui se chevauchent, le deuxième favorise le glissement des grandes pièces vers la gauche (ce qui correspond à l'heuristique "les grandes pièces d'abord", bien connue par les placeurs ). Ce dernier tasse les pièces vers la gauche parce que, grâce à lui, la position la plus favorable d'une pièce "sans chevauchement" est la plus à gauche possible.

Le choix  $p_2 > p_1 > p_3$  exprime notre but de trouver des placements sans chevauchement et de "longueur" minimum. Les pièces qui se chevauchent ont tendance à s'éloigner réciproquement. Lorsqu'elles se trouvent à proximité du bord droit, elles tendent à "sortir" de la "plaque" ( $p_1 > p_3$ ) . Cette sortie est contrôlée par  $p_3$  ; une grande valeur de  $p_3$  favorise la minimisation de la longueur de la "plaque", mais elle peut "bloquer" la "sortie de secours" de ces pièces et inversement.

Pour la minimisation de la longueur de la "plaque", il faut noter aussi le rôle spécifique du bord droit. La minimisation de chevauchement des pièces qui dépassent ce bords est la "force principale" qui "pousse" les pièces vers la gauche. Par conséquent, ce bord doit être toujours choisi "à gauche" de la position calculée à partir de la perte qu'on veut obtenir.

Le poids  $p_0$  doit être choisi de manière qu'il "détecte" la translation en direction  $Ox$  de la plus petite pièce (en largeur) et qu'il ne permette pas de grands chevauchements.

### 3.3.3 Configurations voisines

La position initiale des pièces est, soit choisie au hasard, soit calculée à l'aide d'une heuristique. Toutefois,

il faut noter que la méthode dépend très peu de la position initiale.

Le passage du système d'une configuration à une configuration voisine est réalisé à l'aide de certaines transformations simples choisies au hasard dans un "répertoire" qui contient deux sortes de transformation :

- a) les transformations locales
- b) les transformations globales

### 3.3.3 a Transformations locales

#### a<sub>1</sub> ) Symétrie par rapport à Ox (Oy)

Ici on doit tenir compte des contraintes du problème en question. Par exemple dans l'industrie du vêtement seules certaines symétries sont permises tandis que dans la découpe de tôle toutes les symétries sont permises. Pour les pièces de forme quelconque, cette transformation est très importante. Elle permet d'exploiter les formes spécifiques des pièces pour trouver des imbrications favorables.

#### a<sub>2</sub> ) Translation avec un pas fixe dx (dy) parallèle à Ox (Oy)

Ce pas est fixe à une température donnée. Au fur et à mesure que la température diminue, le pas diminue aussi jusqu'à une valeur minimale. La valeur minimale pour dx est égal au pas de discrétisation (cf 3.4.1). La valeur minimale pour dy dépend du problème concret.

#### a<sub>3</sub> ) Translation parallèle à un des axes jusqu'au "rectangle le plus proche" dans une direction choisie

Cette transformation est illustrée en fig 3.2.

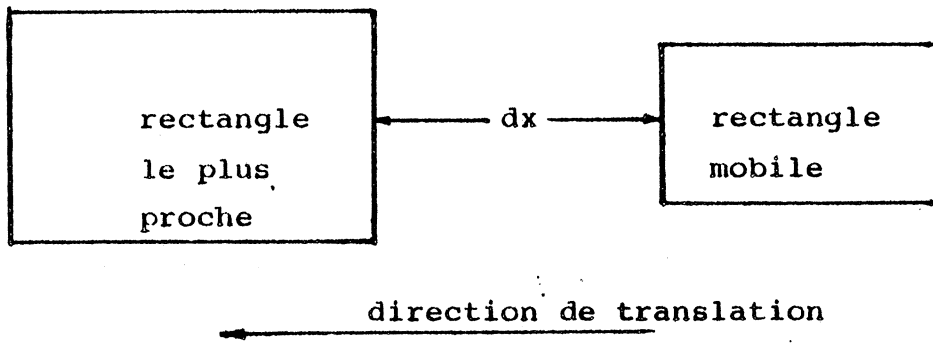


Fig 3.2.a

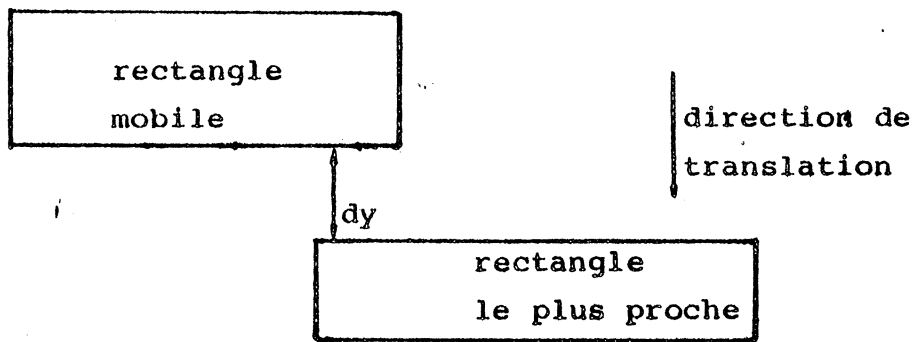


Fig 3.2.b

a<sub>4</sub> ) Echange de deux pièces en contact suivant une direction Ox ou Oy

Cette transformation est appliquée, lorsque l'application de la transformation précédente trouve deux rectangles qui se touchent ( $dx = 0$  ou  $dy = 0$ ) et est illustrée par la fig 3.4.

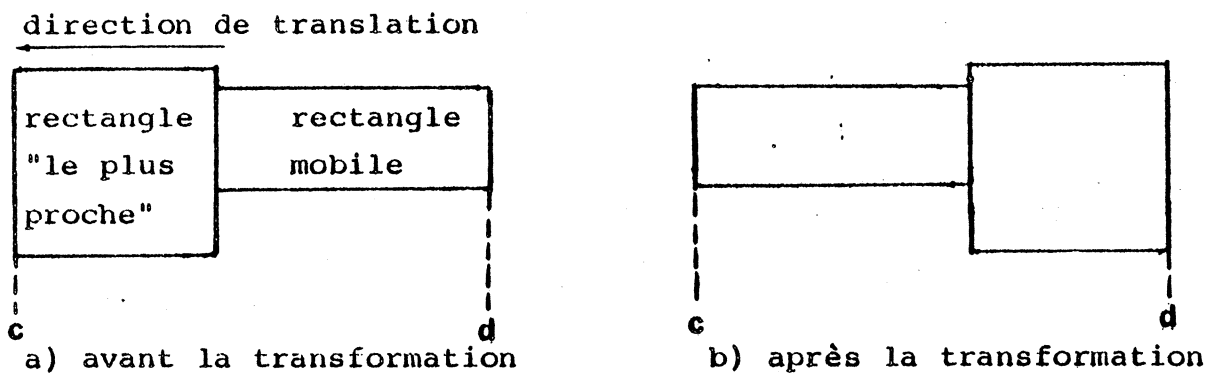


Fig 3.3.



a<sub>5</sub> ) Translation parallèle à 0x ou 0y "par dessus une pièce"

Cette transformation s'applique lorsque l'application de la transformation du point a<sub>3</sub> ) donne dx = 0 ou dy = 0. Dans ce cas, la pièce mobile peut passer de l'autre côté de la pièce en contact avec elle (fig 3.4).

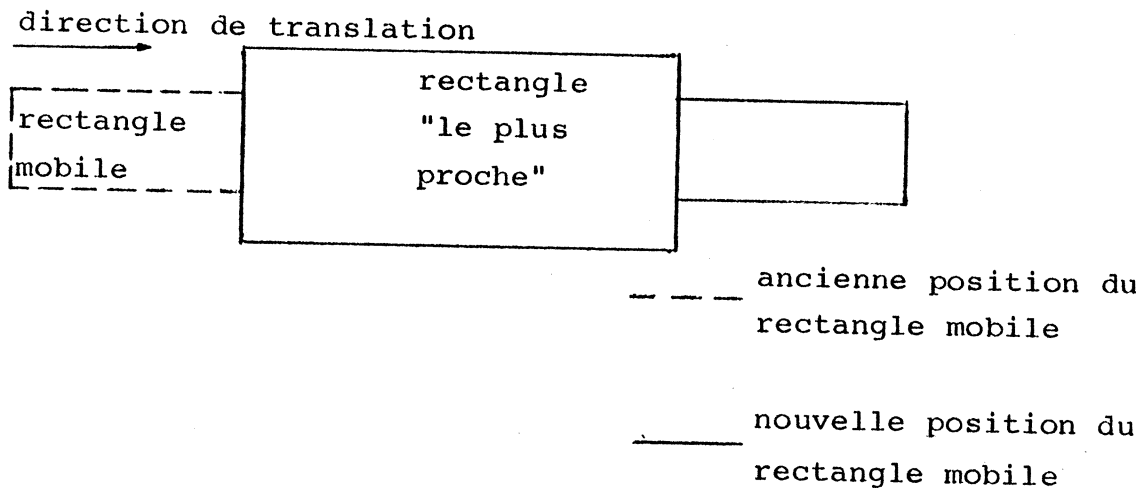


Fig 3.4.

Cette transformation est introduite afin de permettre aux petites pièces de trouver des positions favorables au voisinage de leurs positions courantes (cf. 3.3.3.b).

Le choix entre a<sub>4</sub> et a<sub>5</sub> est fait au hasard.

3.3.3.b. Transformations globales

b<sub>1</sub> ) Echange de deux pièces "équivalentes"

En (2.4.) on a discuté sur la nécessité de construire des "macroatomes équivalents" ou d'introduction des transformations qui simulent en quelque sorte cette équivalence. Dans notre algorithme, deux pièces i et j sont considérées équivalentes si  $\text{abs}(l_i - l_j) / \max(l_i, l_j) < \epsilon$  (équivalence en longueur) ou  $\text{abs}(w_i - w_j) / \max(w_i, w_j) < \epsilon$  (équivalence en

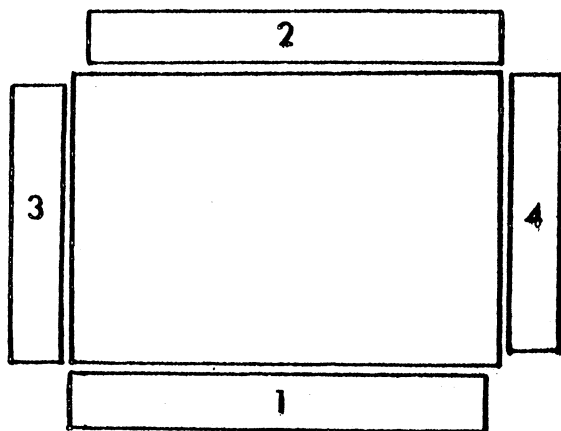
largeur) ou les deux (équivalence totale).

La transformation consiste à échanger deux pièces équivalentes indépendamment de leurs positions respectives dans la "plaque". Les nouvelles positions des pièces sont choisies suivant la sorte d'équivalence entre les pièces en question. Cette transformation donne à l'algorithme une vue générale de la configuration courante.

En général,  $\epsilon \in (0.1, 0.2)$  mais on peut donner à  $\epsilon$  des valeurs plus grandes ou bien on peut considérer "équivalentes" toute paire de pièces. Cela se manifesterait par une hausse considérable du temps de calcul parce qu'on introduirait à priori un grand nombre de transformations non acceptables.

b2) "Parachutage" au voisinage d'une pièce équivalente

C'est le cas, où la pièce la plus petite (en surface) de deux pièces "équivalentes" est "parachutée" au voisinage de l'autre. Ici le terme "au voisinage" diffère suivant la sorte d'équivalence entre ces deux pièces et est illustré par la fig 3.5.



Les positions (1) et (2) ((3) et (4)) sont valables pour les pièces équivalentes en longueur (largeur). Toutes les quatre pour l'équivalence totale.

Fig 3.5.

Cette transformation est en quelque sorte l'échange de la pièce avec le vide. Dans cette méthode on ne considère que les positions des pièces et on ne "connait" pas la "plaque". Plus

précisément on n'identifie pas les positions non remplies de la plaque. Les transformations  $a_5$ ) et  $b_2$ ) servent à "identifier" le vide uniquement lorsqu'il se trouve au voisinage d'une pièce "équivalente" ou d'une pièce en contact avec la pièce mobile.

### 3.4. L'algorithme

#### (0) Initialisation

On initialise les paramètres et on calcule les positions initiales des pièces.

(1) On choisit au hasard la pièce à bouger  $i_b$  et la direction de translation dir.

(2) On choisit au hasard une des transformations du répertoire.

(3) On calcule la nouvelle position de  $i_b$ , l'énergie  $E_n$  dans cette position et

$\Delta E = E_n - E_a$  ( $E_a$  - l'énergie dans l'ancienne position).

(4) Si  $\Delta E > 0$  et  $r > \exp(-\Delta E/T)$  ( $r$  - uniformément distribué sur  $(0,1)$ ), la nouvelle position est rejetée, la pièce reste dans l'ancienne position. Sinon, la nouvelle position est acceptée.

(5) Si l'équilibre pour la température  $T$  est atteint, on passe à la température suivante et on change  $dx$  et  $dy$ . Aller en (6).

Sinon, aller en (1)

(6) Si la solution optimale est atteinte, STOP. Sinon, aller en (1).

### 3.4.1. Calcul de la surface de chevauchement

Initialement les pièces sont représentées par des contours polygonaux. La pièce est définie par plusieurs contours si elle a des trous et par un seul sinon.

Pour le calcul de la surface de chevauchement on a choisi la représentation en bandes des pièces :

chaque pièce est "découpée" en bandes rectangulaires (appelées segments ci-dessous), parallèles à  $Oy$ , qui débordent vers l'extérieur comme le montre la fig 3.6.

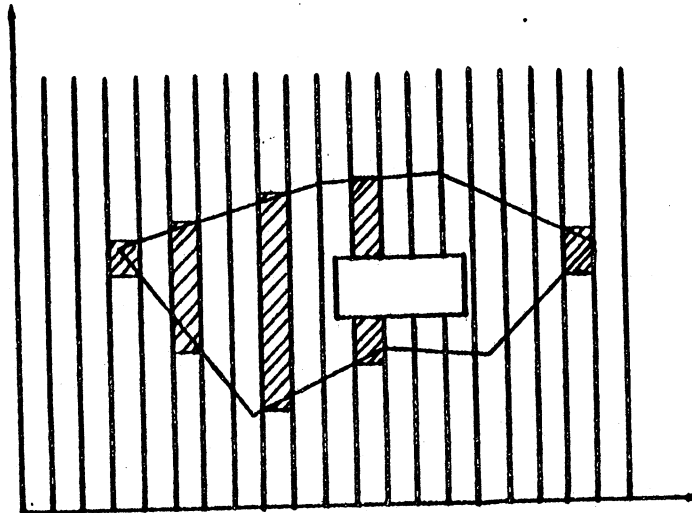


Fig 3.6.

A un niveau (abscisse) donné peut correspondre un ou plusieurs segments. Une pièce est alors un ensemble de segments ou plus exactement un ensemble de listes des segments, chaque liste contenant les segments d'un certain niveau.

La surface de chevauchement est alors calculée comme intersection des segments respectifs.

La qualité d'approximation dépend beaucoup du pas de discrétisation (largeur d'une bande). Ce dernier est un paramètre important qui va jouer beaucoup dans la solution du problème du point de vue temps de calcul, encombrement, mémoire et perte.

Cette représentation est "chère" (en temps de calcul) si

l'on veut considérer les rotations des pièces. Pour beaucoup d'applications (industrie du tricot, industrie des vêtements, etc.) les rotations des pièces ne sont en général pas permises (contrainte droit fil). La mise en bandes des pièces est alors faite une fois pour toute au début du programme. Lorsque les rotations sont permises (découpe de tôle, etc.) la nouvelle mise en bandes est faite, si nécessaire, pendant le déroulement du programme. S'il y a un petit nombre de rotations permises (seulement suivant certains angles), on met en bandes un exemplaire de la pièce pour chaque rotation permise et pendant l'exécution du programme, on choisit "l'exemplaire" adéquat.

Il faut noter que, pour l'approche choisie, la "plaque" n'existe qu'en forme de rectangle (4 coordonnées) et sert uniquement pour calculer le chevauchement. Elle n'est pas mise en bandes. Cela peut présenter des économies en mémoire lorsqu'on place des pièces en plusieurs exemplaires.

### 3.4.2. Le choix des paramètres

Les paramètres de la méthode sont : la température initiale  $T_0$  et les pas des translations  $dx_0$  et  $dy_0$ .

Pour  $dx_0$  ( $dy_0$ ) on a choisi la moyenne des longueurs (largeurs) des pièces.  $dx_0$  est arrondi comme un multiple de pas de discrétisation des pièces.

Pour la température, étant donné que la valeur  $T_0$  doit correspondre avec l'état "liquide du système", pour la calculer on a appliqué la règle suivante :  $T_0$  est la température à laquelle le chevauchement maximal est accepté avec une haute probabilité  $p$ . Autrement dit,  $T_0$  est calculé d'après l'équation  $\exp(-\Delta E_{\max}/T_0) > p$  où

$$\Delta E_{\max} = P_2 * \max(dx_0 * W_{\max}, dy_0 * l_{\max})$$

$$l_{\max} = \max_i \{l_i\}, W_{\max} = \max_i \{W_i\}$$

Pour  $p$  on a choisi une valeur en intervalle  $[0.8, 0.9]$  ce

qui donne  $T_0 = 5 * \Delta E_{\max}$  pour  $p = 0.8$  et  $T_0 = 10 * \Delta E_{\max}$  pour  $p = 0.9$ .

Cette règle a l'avantage de calculer automatiquement la température initiale à partir des données du problème concret.

Le changement de la température est fait d'après la formule  $T_{i+1} = a * T_i$  où  $a \in [0.7, 0.9]$ .

### 3.4.3. Le critère d'équilibre

Le critère pour déterminer l'équilibre à une température  $T$  est celui proposé par Kirkpatrick (cf 2.4.) :

l'équilibre est atteint si on a, en moyenne, 10 transformations acceptées par pièce ou 100 transformations (acceptées ou rejetées) par pièce.

### 3.4.4. Le critère d'optimalité

Ce critère est une combinaison des critères simples suivants :

1. Le premier terme de la fonction d'énergie (1) devient nul.
2. Le nombre des transformations acceptées n'est pas atteint pour trois températures successives.
3. Le nombre maximal d'itérations est dépassé.
4. La chaleur spécifique est très proche de zéro.

Naturellement le critère idéal est le premier. Mais comme on ne sait pas à l'avance si on peut placer ou non toutes les pièces dans la plaque, il faut le combiner avec les autres.

Dans nos expérimentations on n'a gardé que les critères 1 et 3.

### 3.5. Expérimentation

La méthode a été testée sur plusieurs problèmes de placement. L'expérimentation a été effectuée en deux phases.

Dans la première phase, la méthode a été programmée en PASCAL et testée sur les microordinateurs APPLE - II et PASCALINE (32K, mot de 16 bits) pour le placement des rectangles.

Dans la deuxième phase on a testé la méthode pour le placement des pièces irrégulières. Le programme est écrit en FORTRAN et les tests ont été effectués sur l'ordinateur CII<sup>®</sup> - HB 68 de l'Université de Grenoble.

#### 3.5.1. Première phase

Notre but dans cette phase n'était pas de comparer cette méthode avec les méthodes précédentes pour le placement des rectangles (cf 1.5.), mais de la comparer avec l'algorithme glouton, de tester différentes stratégies d'évolution du système et de fixer les valeurs de différents paramètres.

C'est pendant cette phase que nous avons établi le répertoire des transformations (cf.3.3.3).

##### 3.5.1.a Le répertoire

La première version du programme ne comprenait que les translations simples a2. L'évolution du système était très lente, même pour des problèmes avec un petit nombre de pièces. L'introduction de la transformation a3 (translation jusqu'au rectangle le plus proche), a donné beaucoup de "dynamisme" à la méthode. Elle est devenue ainsi au moins 10 fois plus rapide.

Ces deux translations sont complémentaires : à hautes températures la translation a3 est plus efficace alors que à basses températures domine la translation a2. La translation

a2 est surtout importante pour les pièces irrégulières, pour "remplir" les "trous" entre des pièces en contact.

La stratégie que nous avons adoptée ici pour l'utilisation de la translation a2 (le pas diminue avec la température) est suggérée par le fait que à hautes températures l'amplitude de déplacement des atomes est grande et qu'elle diminue avec la température. Un pas constant à toutes températures est possible compte tenu de la complémentarité a2-a3 (autrement non), mais cela introduirait un grand nombre de configurations non acceptables.

Cette approche diffère de celle adoptée par Metropolis (cf (2.2) ) parce que les tests faits pour le placement des rectangles l'ont révélée "plus rapide".

Pendant les essais, les petites pièces, une fois sorties de la plaque, ne pouvaient pas rentrer de nouveau. L'introduction de la transformation a5 (translation par dessus une pièce) leur donne une possibilité de rentrer. Cette transformation s'est révélée aussi très utile pour les changements à l'intérieur de la plaque, pour remplir une portion vide qui se trouve à l'opposé d'une pièce en contact avec la pièce mobile.

### Transformations globales

Lorsqu'on part d'une configuration initiale choisie aléatoirement, il est très peu probable que les pièces "partent" dans de "bonnes" directions. On est alors souvent bloqué dans des configurations éloignées de la solution optimale, dans lesquelles les transformations locales deviennent impuissantes. Les transformations globales remédient à ce handicap. Cette situation est analogue à la méthode expérimentale de recuit, lorsque le système "tombe" dans un état métastable. Dans ce cas, pour le faire sortir, il faut, en général, une "force extérieure" ; hausse de la température, addition d'un réactif, etc.

Dans notre méthode, les transformations globales simulent en quelque sorte cette "force extérieure". De plus, grâce à elles la méthode a une vue générale du problème.



Les transformations locales dépendant fortement des dimensions des pièces à placer, du pas de discrétion, etc.

Les transformations globales, elles manifestent une certaine indépendance. La transformation à effectuer est choisie aléatoirement. Pour les tests on a gardé le rapport :

$$\frac{\text{nombre de transformations locales}}{\text{nombre de transformations globales}} = 2$$

A l'intérieur de chaque classe on a essayé différentes probabilités pour chaque transformation. Les résultats les plus stables sont obtenus en donnant une probabilité uniforme à chaque transformation, même si les statistiques faites sur le taux d'utilisation de chaque transformation révèlent certaines "préférences".

Le tableau ci-dessous représente le taux d'utilisation (observé en pratique) de chaque transformation :

Type de transformation.	pourcentage de transformations acceptées.
a <sub>1</sub>	40-70
a <sub>2</sub>	40-70
a <sub>3</sub>	20-30
a <sub>4</sub>	10-30
a <sub>5</sub>	3-15
b <sub>1</sub>	5-18
b <sub>2</sub>	10-40

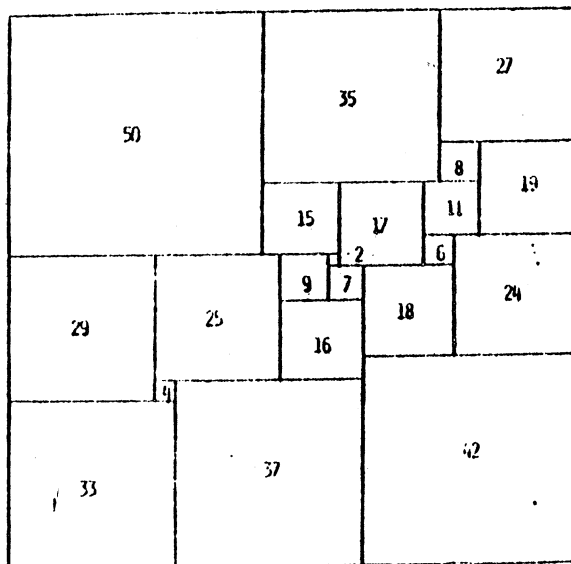
Ces chiffres sont relatifs aux problèmes de placement de pièces irrégulières où l'on part d'une configuration initiale choisie aléatoirement. Quant on part d'une configuration initiale connue (par conséquent d'une température plus basse), tous ces pourcentages sont légèrement diminués sauf na<sub>1</sub>, na<sub>2</sub> et na<sub>3</sub> (na<sub>i</sub> représente le pourcentage des transformations de type a<sub>i</sub> acceptées).

Les essais montrent la stabilité du rapport

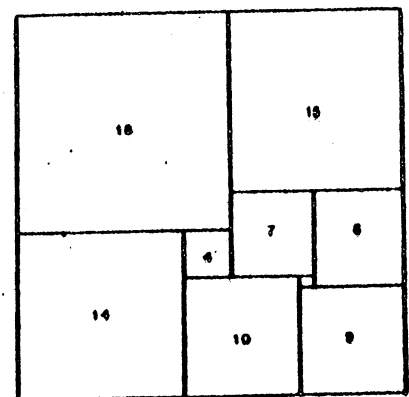
$$na_2 / na_3 \quad (=2).$$

### 3.5.1.b Comparaison avec l'algorithme glouton

Pour faire cette comparaison on a traité plusieurs problèmes de puzzle, guillotine et non guillotine avec des nombres de pièces à placer différents. Les problèmes non guillotine retenus étaient : un rectangle parfait simple de l'ordre 9 (fig.3.8.b)\*) et le seul carré parfait simple de l'ordre 21 connu (fig.3.8.a)\*\*).



a) Carré parfait simple de l'ordre 21.



b) Rectangle parfait simple de l'ordre 9.

Fig.3.8.

Pour tous les problèmes testés, hormis les problèmes guillotine avec un nombre de pièces  $n \leq 10$ , la suprématie

\*) Extrait de Federico P.J. "Squaring rectangles and squares" dans "Graph theory and related topics". Bondy J.A., Murty U.S.R. (eds.) Academic Press 1979.

\*\*\*) Extrait de Duijvestijn A.J.W. "Simple perfect squared square of lowest order", J. of Combinatorial Theory, series B(25), 1978, 260-263.

de cette méthode par rapport à l'algorithme glouton est nette. L'allure générale des solutions est la même. Dans les premières itérations l'algorithme glouton descend plus vite que la méthode "thermodynamique" tandis que dans les itérations suivantes cette dernière continue à descendre pendant que le premier reste constant. La fig 3.9. représente les évolutions des solutions du problème de la fig.3.8.b par les deux méthodes et la fig.3.10, l'évolution de la chaleur spécifique en fonction de la température pour le même problème résolu par la méthode "thermodynamique".

Dans la plupart des cas (environ 95 %), les résultats obtenus sont analogues à ceux de la fig.3.9.

La méthode "thermodynamique" donne des configurations dont l'énergie est de 30 à 40 % plus petite que celle donnée par l'algorithme glouton (parfois même jusqu'à 90 % plus petite).

Le nombre d'itérations est de 1.8 - 3.4 plus élevé (en moyenne 2.5 fois). De plus, la solution obtenue par la méthode "thermodynamique", pour la même valeur de la fonction d'énergie, est plus "régulière" : la configuration obtenue est beaucoup plus proche de la configuration finale. Cette caractéristique est très visible dans les cas de la découpe non guillotine.

Le nombre d'itérations de la méthode "thermodynamique" est de l'ordre  $aN^3$  avec  $a \in (2.5, 5)$ .

La solution du problème de la fig.3.8.b a été aussi comparée avec celle obtenue par une méthode heuristique "non-guillotine" pour le placement des rectangles \*). La méthode "thermodynamique" donne des meilleurs résultats (du point de vue perte) mais avec des petits chevauchements résiduels (cf 3.6.2).

---

\*) Andrieux A., Rapport de stage, ENSIMAG 1982.

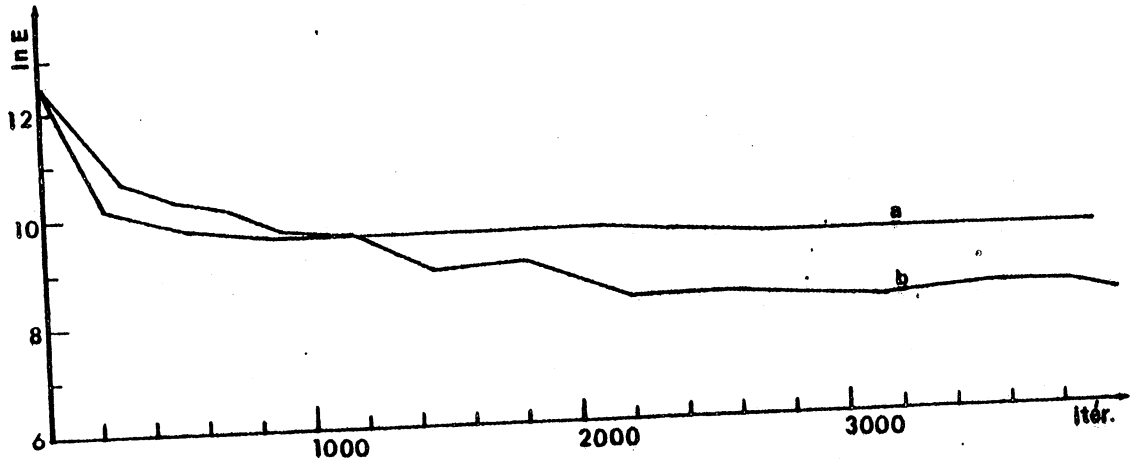


fig. 3.9.  
a - algorithme glouton  
b - méthode thermodynamique

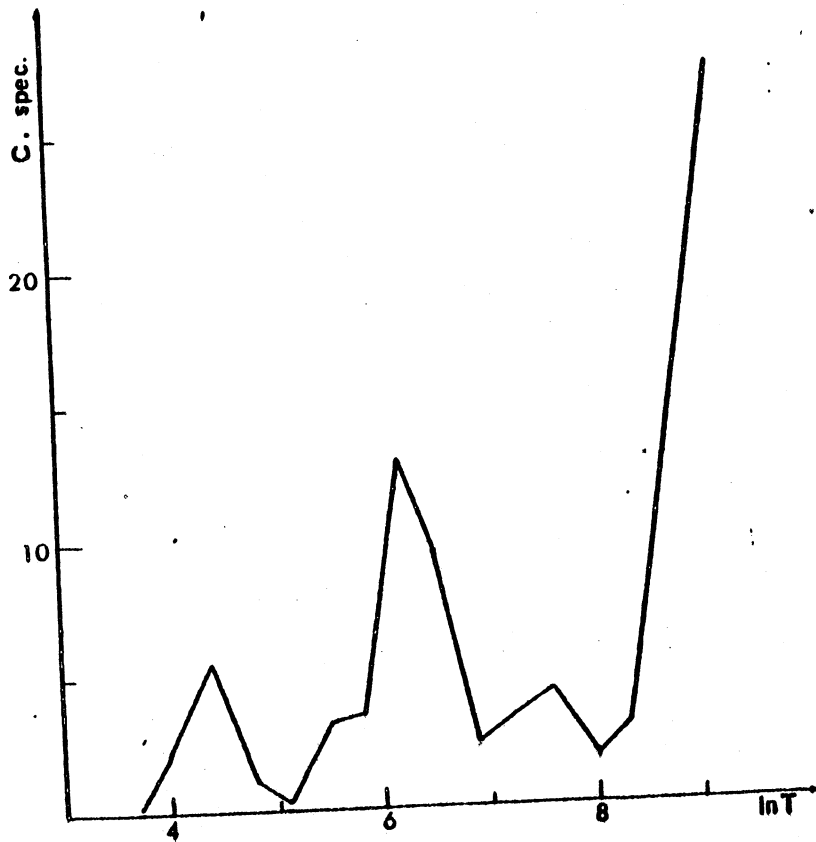


fig 3. 10

### 3.5.2. Deuxième phase : placement des pièces irrégulières

Etant donné que dans l'état actuel la méthode "thermodynamique" est très "chère" en temps de calcul, le nombre d'essais pour le placement des pièces irrégulières est très limité. Les tests effectués concernent le placement des pièces de costumes (pantalon, veste, gilet, total de 23 pièces). Les résultats obtenus sont comparés (du point de vue perte) avec ceux donnés par le logiciel RIMBAUD.

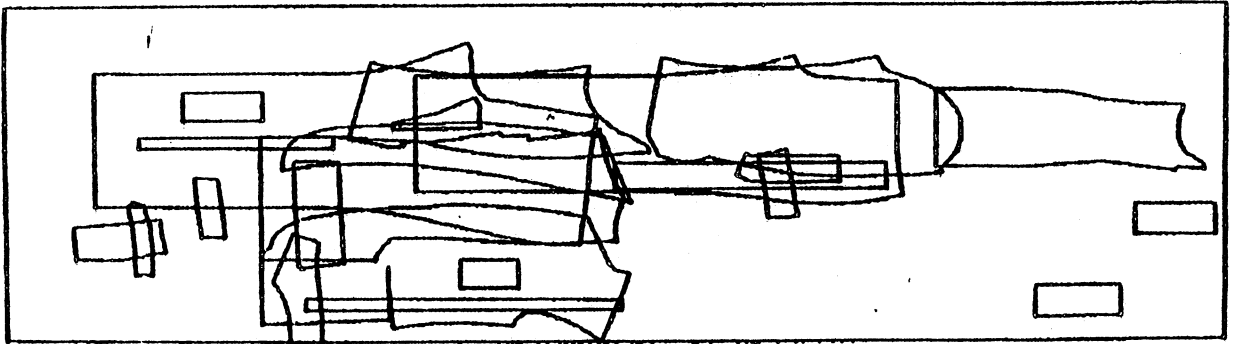
Pour le placement des dix pièces (les pièces constituant la veste) dans tous les cas, la méthode "thermodynamique" donne de meilleurs placements même pour un pas de discrétisation assez grand (= 100). La perte diminue de 2 - 3 % pour un temps de calcul 10 mn. Pour 23 pièces, le programme RIMBAUD donne des placements avec 18.6 % de perte (pas de discrétisation égal à 40 %) en 40 secondes et avec 21.7 % de perte (pas de discrétisation égal à 100) en 30 secondes. La méthode "thermodynamique" donne un placement avec 20 % de perte (pas de discrétisation égal à 100) en environ 25 minutes. On n'a pas essayé avec un pas plus petit. Les différentes phases de ce placement à différentes températures sont présentées à la fig.3.11. La figure 3.12 présente l'évolution de la chaleur spécifique en fonction de la température. La chaleur spécifique n'est pas proche de zéro parce que la sortie du programme s'est faite (à cause de la limite en temps de calcul) alors que le système était encore en "activité". Le nombre d'itérations est de l'ordre a.N avec a 2 000 N - le nombre de pièces)

### 3.6. Conclusions sur la méthode "thermodynamique"

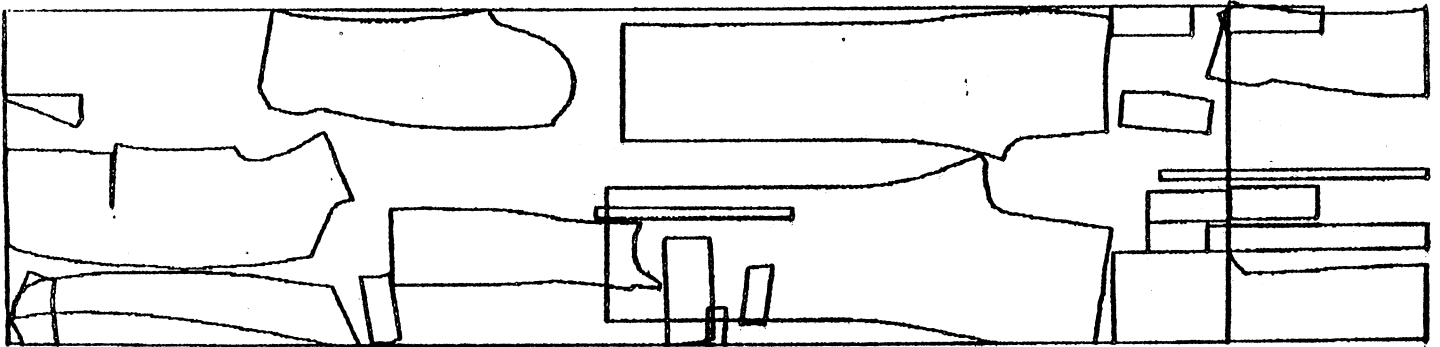
Nous n'avons trouvé qu'une référence consacrée aux méthodes statistiques (Böhme et Graham [1979]) bien qu'à notre avis elles soient très souvent utilisées en pratique.

Contrairement à la méthode "thermodynamique", elles présentent un caractère glouton et local.

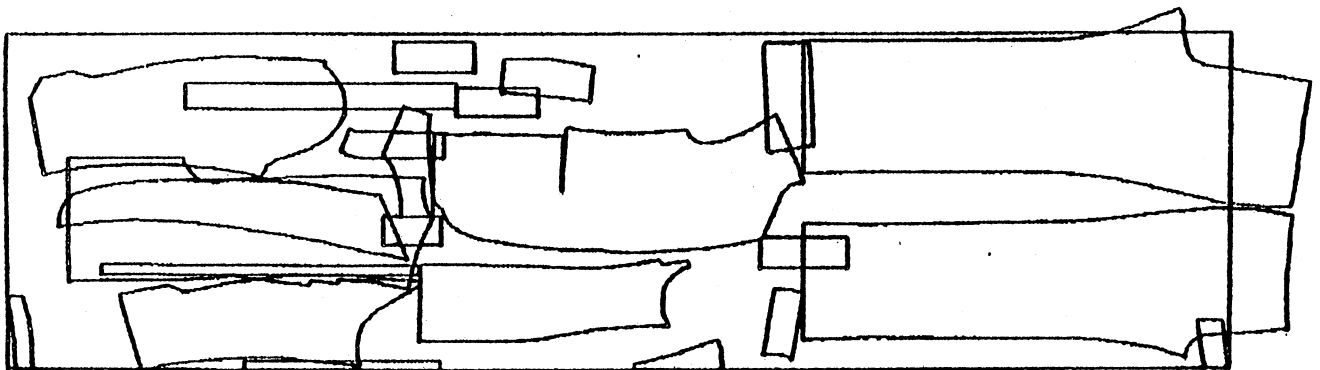
Cependant, un certain nombre de défauts rendent la méthode "thermodynamique" peu opérationnelle pour l'instant



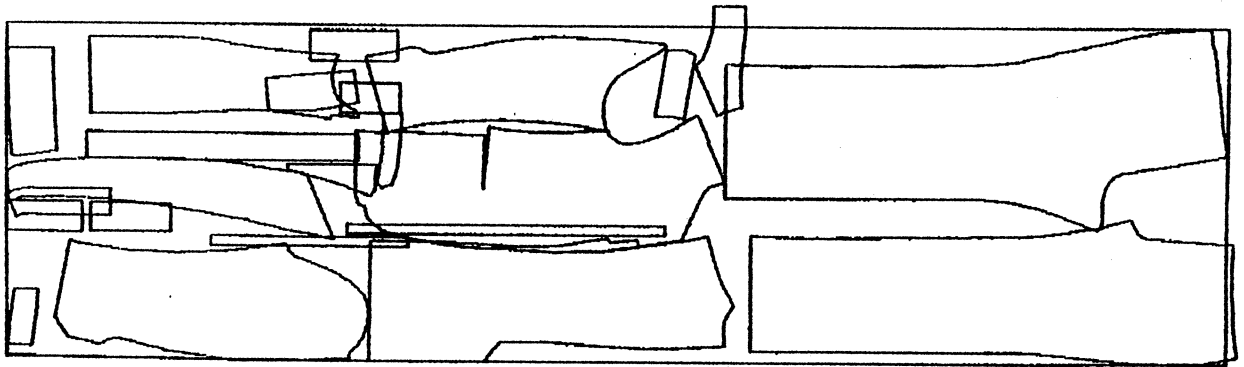
a) T=242865



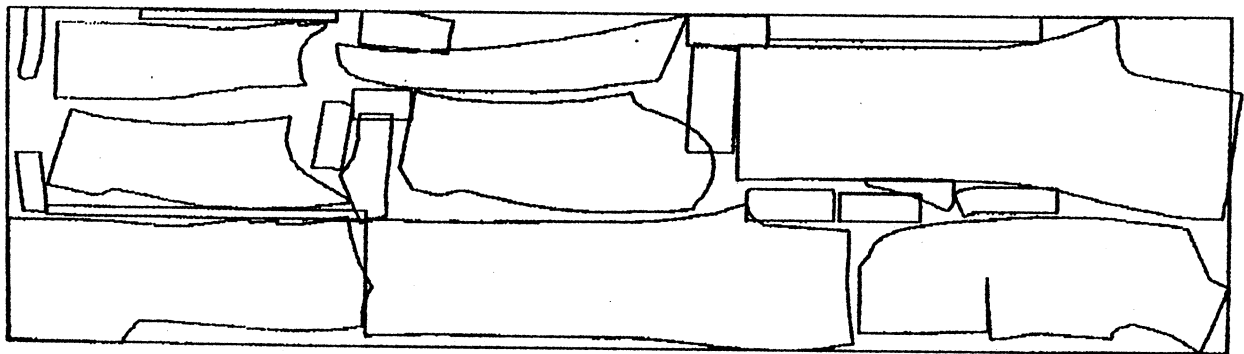
b) T=47819



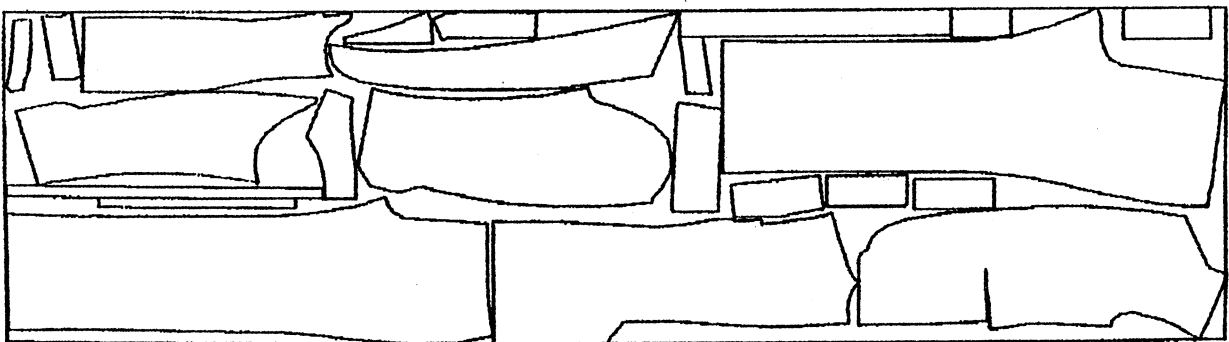
c) T=68001



c) T=4176



c) T=310



f) T=4.5

fig. 3.11.

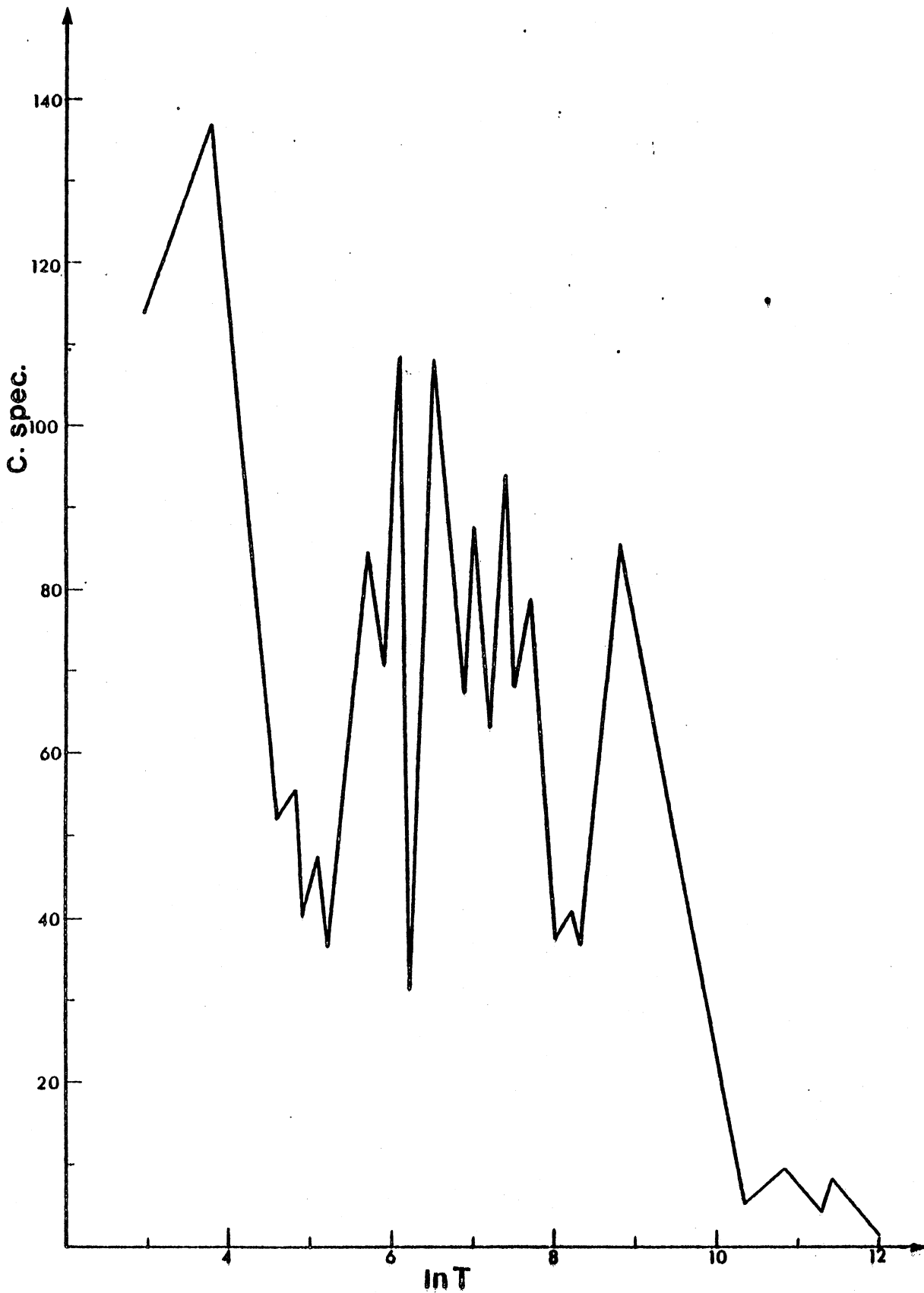


fig 3. 12



(mais l'objectif de cette thèse était plus d'explorer de nouvelles approches que d'élaborer un algorithme industriel).

### 3.6.1. Le temps de calcul

Dans 3.5.2., on a vu que la méthode "thermodynamique" est très "chère" en temps de calcul (25-30 minutes pour le placement de 23 pièces comparées à 30-35 secondes pour le logiciel RIMBAUD (cf 4.1.)).

On peut faire une comparaison indirecte avec le programme ASCO décrit par Böhme et Graham [1979]. Ce dernier est un algorithme stochastique glouton utilisé dans la découpe de tôle par plusieurs entreprises (dont Krupp). Les meilleurs résultats sont obtenus pour 2 000 essais par pièce. Dans la méthode "thermodynamique" on est dans les mêmes chiffres.

ASCO est implanté sur un miniordinateur GA 220 ou GA 460 (32K, 16 bits). Sur GA 220 il utilise une minute par pièce, sur GA 460 environ 30 secondes et sur GA 460 avec point flottant, 5 - 10 secondes. Le cycle mémoire de GA 460 est 500 ns. De ce point de vue, les résultats de la méthode "thermodynamique" sur MULTICS sont comparables avec ceux de ASCO. Il faut noter que, d'après Böhme et Graham, ASCO ne donne de bonnes imbrications que lorsqu'on place un grand nombre de petites pièces.

La plupart du temps de calcul est utilisé dans l'évaluation de la fonction d'énergie, et on peut attendre des améliorations importantes si on trouve une méthode plus économique pour le calcul de la surface de chevauchement entre deux pièces. A notre avis, un algorithme efficace, qui calcule cette surface à partir des représentations en contours polygonaux des pièces, diminuera considérablement le temps de calcul et l'encombrement mémoire.

Une autre possibilité d'amélioration consiste dans le choix d'un répertoire plus sophistiqué.

### 3.6.2. Le chevauchement résiduel

Cette méthode progresse vers la solution optimale à partir de l'ensemble des solutions non réalisables. Chaque solution intermédiaire est non réalisable à cause des chevauchement entre les pièces. La première solution réalisable trouvée est la solution optimale. A cet égard elle est analogue à l'approche du point extérieur dans les méthodes de pénalités en optimisation non linéaire.

Par conséquent, lorsqu'on s'arrête à cause de la limitation en temps de calcul, le placement obtenu peut présenter des chevauchements résiduels entre les pièces. Pour l'instant, on n'a pas pu trouver un critère qui garantit un placement sans chevauchement. Le choix des poids  $p_0$ ,  $p_1$ ,  $p_2$  et  $p_3$  favorise beaucoup de tels placements (comme nous l'avons vu dans 3.3.2) mais il n'est pas suffisant pour éliminer de petits chevauchements (qui peuvent très facilement être éliminés par un opérateur humain).

Actuellement ce problème est résolu en utilisant, si nécessaire, un algorithme glouton à la fin. Cet algorithme considère les pièces qui se chevauchent et les translate suivant la direction la plus "favorable" jusqu'à ce qu'on obtienne un placement sans chevauchement ou bien jusqu'à ce qu'aucune pièce ne bouge. Cet algorithme ne garantit pas non plus de placements sans chevauchement mais, pratiquement, dans tous les cas on a obtenu de tels placements.

Il faut noter que les chevauchements résiduels étaient présents dans les cas où l'arrêt trouvait le système à une température relativement haute. Pratiquement, si on laisse le système évoluer assez longtemps aux basses températures, ces chevauchements sont absents. D'où l'importance de bien estimer la limite du temps de calcul (le nombre maximal d'itérations).

### 3.6.3. Les avantages

L'avantage principal de cette méthode c'est sa simplicité. Elle peut être très facilement mise en oeuvre. Telle quelle est présentée ici, la partie la plus difficile est la mise en bandes des pièces.

Deuxièmement, cette méthode peut donner de "bons" placements du point de vue perte en échange d'un important temps de calcul.

Troisièmement, cette méthode n'a pas besoin de la représentation de la "plaque" en bandes. Cela peut réaliser une bonne économie de mémoire quand on place des pièces en plusieurs exemplaires. Cette caractéristique peut présenter de l'intérêt pour l'implantation sur micros.

Enfin, la possibilité ouverte pour des améliorations possibles (cf 3.6.1).

### 3.6.4. Considérations pratiques

Cette méthode est conçue pour les problèmes où le nombre de pièces à placer est connu à l'avance (cf 3.1.). Ce cas se rencontre dans l'industrie du vêtement où il faut placer toutes les pièces d'un complet (costume, pyjama, etc.).

Dans les autres cas, ce nombre peut être estimé de différentes manières, par exemple en résolvant le problème du sac à dos unidimensionnel sur la surface.

Dans ce cas, la solution du problème

$$\max \sum_{i=1}^t S_i x_i$$

sous la contrainte

$$\sum_{i=1}^t S_i x_i \leq S$$

$$x_i \geq 0 \text{ et entier, } i = 1, 2, \dots, t$$

où  $S_i$  - la surface de la pièce  $i$   
 $S$  - la surface de la plaque

- t - le nombre total de pièces différentes
- $x_i$  - le nombre des exemplaires de la pièce  $i$   
placée dans la plaque

donne un sous ensemble de pièces à placer. A la fin de l'algorithme, les pièces qui ne "rentrent" pas dans la plaque, sont placées dans les plaques ultérieures. Les portions vides de la plaque qui en résultent, peuvent être "remplies" avec des pièces plus "petites" par un algorithme glouton.

Dans l'état actuel cette méthode peut être utilisée uniquement dans des problèmes où il n'y a pas de contraintes spéciales en temps de calcul (placements effectués à l'avance et mis en bibliothèque). Pour cela, la version actuelle en FORTRAN peut être améliorée. La programmation en assembleur peut beaucoup accélérer le programme existant.

Cette méthode peut être utilisée combinée avec une méthode déterministe telle que RIMBAUD, PAMPA, etc. (cf chapitre IV). La méthode "thermodynamique" est alors utilisée pour le placement des grandes pièces tandis que la méthode déterministe remplit de reste de la "plaque" avec des petites pièces non encore placées.

La figure 3.13 présente un exemple de placement des grandes pièces du problème présenté à la figure 3.11.

Cette méthode peut être aussi utilisée pour la construction des modules rectangulaires à partir d'un petit nombre de pièces irrégulières. La plupart des méthodes pour le placement des pièces irrégulières utilise l'approche suivante : premièrement on construit de "bons" modules rectangulaires et ensuite on les "place" dans la "plaque" (cf 4.1.2.). L'efficacité de ces méthodes du point de vue perte dépend largement de la qualité des modules. La méthode "thermodynamique" représente une alternative efficace (du point de vue perte) pour la construction des modules.

4.0 - Introduction

L'équipe de Recherche Opérationnelle de l'IFAC a mis en oeuvre un logiciel très efficace pour le placement automatique des pièces irrégulières. Plusieurs versions de ce logiciel (PAMPA \*) pour le placement dans l'industrie du tricot, RIFAUF pour le placement des pièces de tôle, etc) sont utilisées en industrie avec des résultats encourageants. La méthode est déterministe. Premièrement, on construit des modules rectangulaires. Ensuite on place ces "pièces" rectangulaires dans la "plaque" tout en tenant compte du contour concret du module. Ce programme donne en moyenne 2 à 3 % de perte supplémentaire par rapport aux placements manuels.

Ici, on présente une nouvelle approche pour l'utilisation des méthodes stochastiques dans le placement. L'aléa est introduite à différents niveaux du programme existant. Le placement automatique s'améliore en échange d'une augmentation en temps de calcul.

La méthode consiste à effectuer un certain nombre de placements automatiques où dans chaque placement on change aléatoirement l'ordre des traitements des pièces. Ensuite, à partir des résultats, elle nous aide à décider de continuer les essais afin de trouver un meilleur placement dans la limite de temps CPU accordé, ou bien de se contenter du meilleur placement obtenu jusqu'au moment de la décision. Cette approche peut présenter de l'intérêt dans les problèmes où on n'a pas de contraintes spéciales en temps de calcul.

4.1 - Programme automatique de placement

Ce programme est décrit dans Kuntz et Uhry [1981], Le Borgne et Martin Martin [1982]. Dans ce programme, on distingue la stratégie locale et la stratégie globale.

\*) PAMPA - Placement Automatique de Modules à Pertes Admissibles.

#### 4.1.1 - Stratégie globale

A un instant donné du placement, on applique cette stratégie pour trouver les modules rectangulaires les "plus favorables" à placer. Le terme "plus favorable" est relatif à l'état actuel de la "plaque" et aux pièces qui restent à placer.

La stratégie globale procède en deux phases :

##### a) - Choix d'un trou

On appelle front, la frontière qui sépare la zone occupée de la zone "libre" de toute pièce. Un examen du front détecte les trous. Un trou est une configuration du front comme celui présenté en fig. 4.1. Il est caractérisé par plusieurs paramètres : largeurs, profondeur maximale, profondeur moyenne, surface, etc. A partir de ces paramètres, on choisit le "meilleur" trou à remplir.

##### b) - Détermination des modules susceptibles à placer

On suppose que le trou choisi précédemment a une largeur L. Alors, les modules susceptibles à placer sont obtenus comme solution du problème de sac-à-dos suivant :

$$\max Z = \sum_{i=1}^n c_i x_i \quad (1)$$

sous les contraintes :

$$\sum_{i=1}^n x_i l_i < L \quad (2)$$

$$\sum_{i=1}^n x_i a_{ij} < d_j \quad j=1,2,\dots,m \quad (3)$$

où :

n - le nombre de modules

m - le nombre de pièces

$c_i(l_i)$  - le coût (la largeur) du module numéro i

$a_{ij}$  - le nombre de pièces j qui entrent dans la constitution du module i

$d_j$  - la demande résiduelle pour la pièce j

et :

$x_i$  - représente le nombre de modules numéro  $i$  dans une solution réalisable.

Le coût d'un module mesure en quelque sorte l'utilité à le placer. Il est calculé à partir de la surface, la largeur, la longueur, etc. du module. Ces coûts jouent un rôle très important : un coût élevé donne une priorité au module respectif et ce module va être placé avant les autres. Le changement des coûts peut modifier considérablement la qualité du placement.

#### 4.1.2 - Stratégie locale

C'est la partie originale de cet algorithme et, en même temps, la plus importante. Elle est absente dans toutes les autres approches déterministes (cf. Adamovicz et Albano[1976a], Albano et Sapuppo [1980], Albano [1977], Haims et Freeman [1970]). Dans ces approches, l'effort principal est effectué pour construire de "bons" modules rectangulaires c.à.d. des modules avec une perte acceptable. Ensuite, le module est confondu avec son rectangle circonscrit et on utilise une méthode pour le placement des rectangles.

Cette approche introduit a priori une perte considérable pour plusieurs raisons :

- premièrement, la construction du module est en elle-même un mini problème de placement et les techniques actuelles sont encore loin de donner une solution optimale.
- deuxièmement, la juxtaposition de modules rectangulaires peut créer des surfaces exploitables pour le placement de petites pièces.

La stratégie locale de cette méthode tient compte du contour concret du module et de l'état actuel de la plaque pour trouver, à partir de quelques critères locaux, la position où ce module s'imbrique le mieux. Pour cette stratégie, le module n'est pas un rectangle mais un contour composé, construit à l'aide des bouts de contours (extérieurs, pour être plus précis) des pièces le constituant.

En résumant :

- la stratégie globale est une méthode très rapide pour sélectionner un petit nombre de "bons" modules ou bien, pour introduire un ordre dans le placement de modules. Bien que cette stratégie opère avec les rectangles circonscrits aux modules, elle dépend en quelque sorte de formes particulières des pièces. Cette dépendance s'exprime dans la partie d'examen du front.
  
- la stratégie locale est une méthode "fine" qui utilise les formes particulières des pièces (modules) pour trouver leurs positions dans la "plaque". Si on délivre les pièces dans le "bon" ordre, cette stratégie donne de très "bons" placements. Le programme offre une telle possibilité : on peut donner une priorité à chaque pièce (module) et on peut utiliser uniquement la stratégie locale qui les place selon leur priorité. Cela représente une autre forme de placement automatique dans les problèmes où l'expérience humaine est très vaste (placement des costumes, etc.)

#### 4.2 - La nouvelle approche

L'idée de base de cette approche est de générer aléatoirement de différents ordres de "livraisons" des pièces à la stratégie locale afin d'améliorer le placement automatique. Les expérimentations sont faites dans des problèmes de placement en industrie de vêtement et de tricot.

##### 4.2.1 - Expérimentations en vêtements

Le problème considéré ici est celui traité en 3.5.2 (costume 3 pièces, 23 morceaux à placer). Dans la première série d'expérimentations, on a généré aléatoirement la priorité de chaque pièce proportionnellement à sa longueur. Au bout de 100 essais, on a trouvé un placement avec une perte d'environ 1 % plus petite que la perte de placement automatique. La variation de la perte en fonction du nombre



d'essais est présentée fig. 4.2a.

Dans la deuxième série d'expérimentations, on a changé aléatoirement les coûts des pièces à placer dans le problème de sac-à-dos (cf. 4.1.1.b). Dans le programme de placement automatique, on a  $c_p = s_p$  où  $c_p(s_p)$  est le coût (la surface) de la pièce p.

Dans cette approche  $c_p = r * s_p$  où r est un nombre aléatoire uniformément distribué sur (0.7, 1.2). Le placement automatique est amélioré d'environ 0.8 % après 100 essais. La fig. 4.2b montre la variation de la perte en fonction du nombre d'essais. La fig. 4.3 montre le placement automatique et le meilleur placement obtenu par la première approche .

#### 4.2.2 - Expérimentations dans l'industrie du tricot

Dans ces expérimentations, on a utilisé uniquement la deuxième approche : génération aléatoire des coûts du problèmes de sac-à-dos. On a effectué plusieurs centaines de placements pour pouvoir étudier statistiquement les résultats obtenus. Les expérimentations sont faites pour différents modèles, différentes largeurs du matelas et différents nombres de complets à placer.

Le placement automatique est amélioré de 0,75 à 9,72 % du point de vue perte, avec une moyenne de 3 % en environ 200 essais. En excluant quelques cas "pathologiques", où le placement automatique donnait de mauvais résultats, cette moyenne est de 1,46 %.

La diminution de perte en fonction du nombre d'essais pour 12 échantillons est présentée dans le tableau suivant :

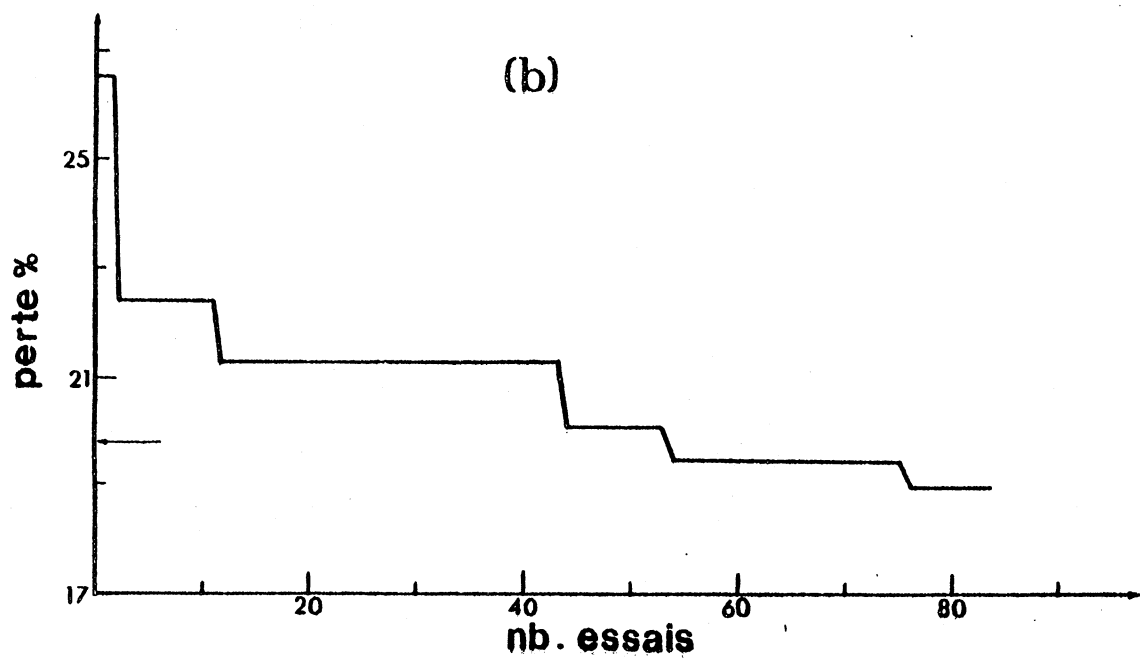
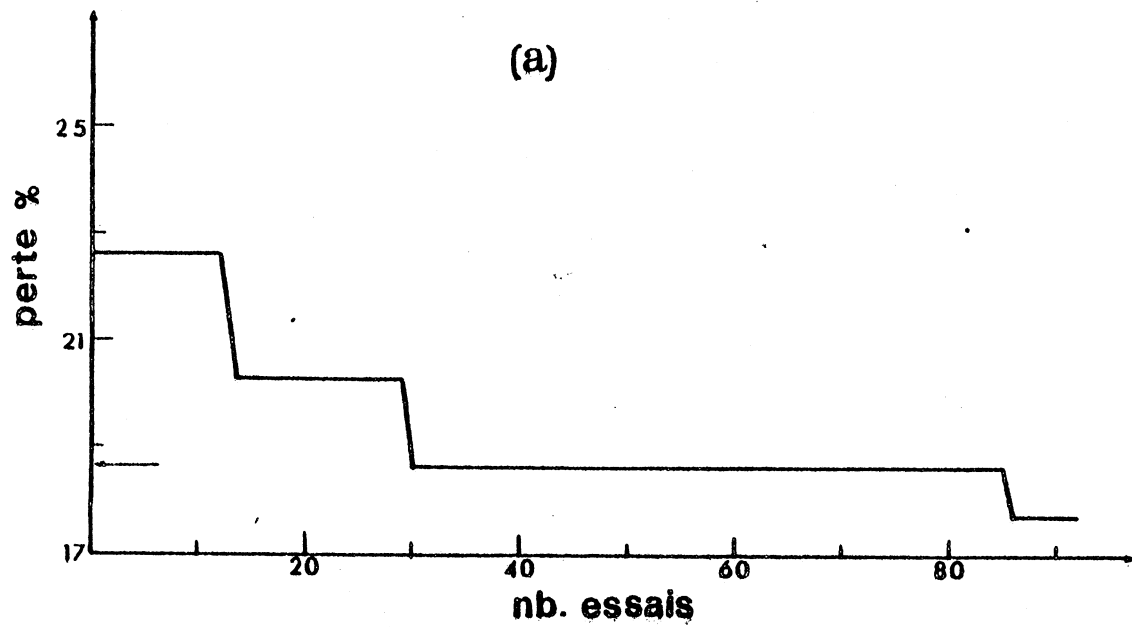


FIGURE 4. 2.

La flèche indique la perte du placement automatique

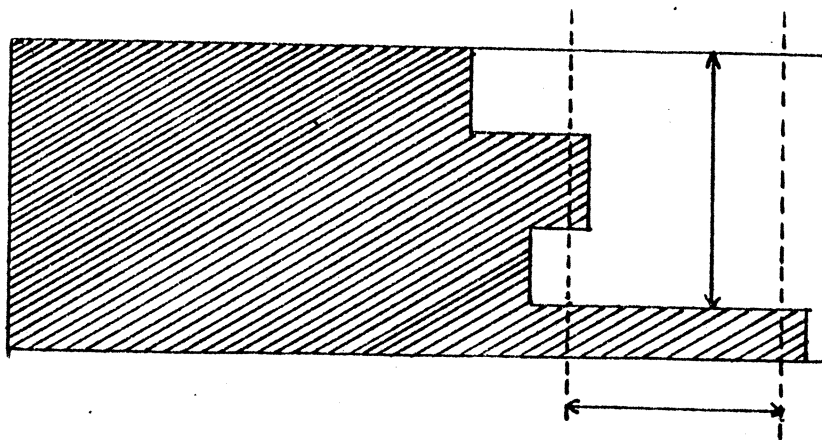
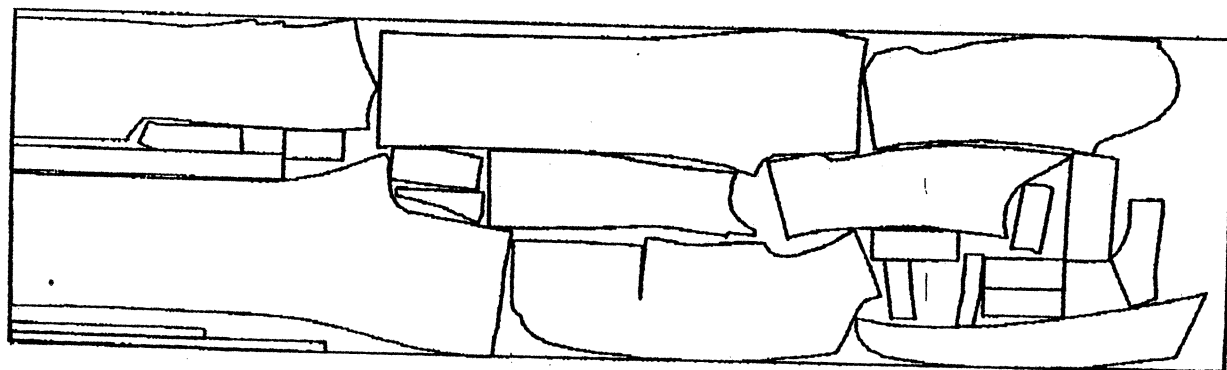
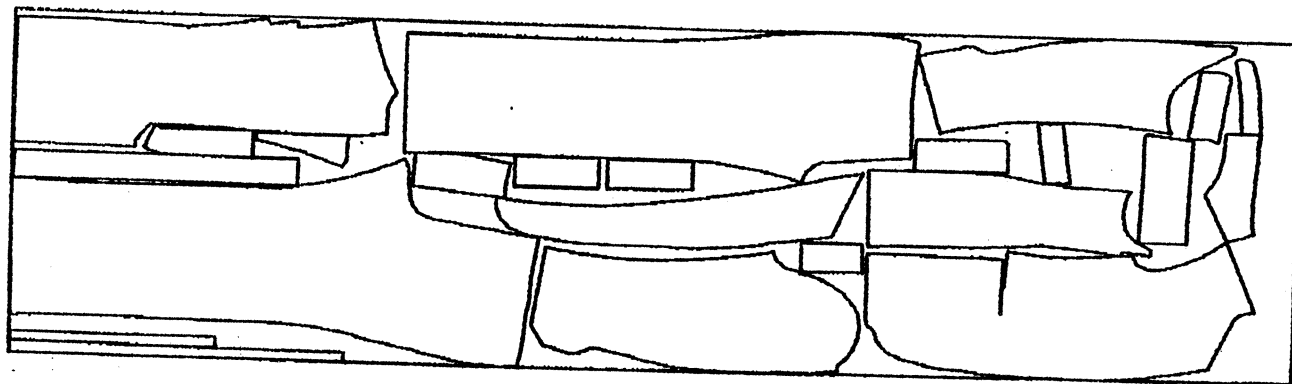


FIGURE 4.1.



a



b

FIGURE 4.3.

- a) placement automatique, perte 18,6 %
- b) placement par la nouvelle approche, perte 17,7 %

N.F	M	Échantillon											
		1	2	3	4	5	6	7	8	9	10	11	12
10	0.9	0.0	0.0	0.2	0.0	4.3	0.4	0.0	1.3	0.0	0.9	0.8	3.2
20	1.3	0.0	0.0	0.2	0.0	4.3	2.4	0.0	1.3	0.3	0.9	3.1	3.2
30	1.8	0.0	0.0	2.9	0.0	4.3	2.4	1.2	1.3	2.1	0.9	3.1	3.2
40	2.0	0.3	0.0	2.9	0.0	6.1	2.4	1.2	1.3	2.1	1.1	3.1	3.2
50	2.0	0.3	0.0	2.9	0.0	6.1	2.4	1.2	1.3	2.1	1.1	3.2	3.2
60	2.0	0.3	0.0	2.9	0.0	6.1	2.4	1.2	1.3	2.1	1.1	3.2	3.2
70	2.0	0.8	0.0	2.9	0.0	6.1	2.4	1.2	1.3	2.1	1.1	3.2	3.2
80	2.0	0.8	0.0	2.9	0.0	6.1	2.4	1.2	1.3	2.1	1.1	3.2	3.2
90	2.1	0.8	0.8	2.9	0.0	6.1	2.4	1.6	1.3	2.3	1.1	3.2	3.2
100	2.2	0.8	0.8	2.9	0.0	6.1	2.4	1.6		2.3	1.4	3.2	3.3
110	2.8	1.1	0.8	3.8	0.0	9.7	2.4	1.6		2.3	1.4	5.2	
170	2.9	1.1	0.8	3.8	0.7	9.7	2.5	1.7		2.3	1.4	5.2	
200	3.0	1.1	0.8	3.8	0.7	9.7	2.9	1.7		3.0	1.4	5.2	

N.F = nombres d'essais

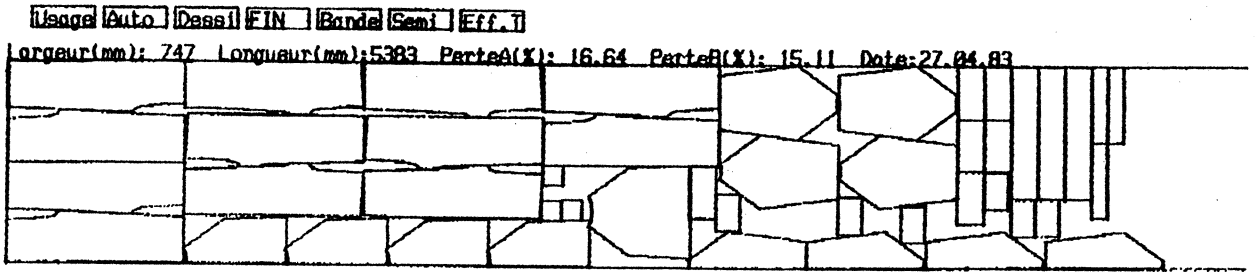
M = moyenne

La figure 4.4 montre quelques placements automatiques et les meilleurs placements respectifs obtenus par cette approche.

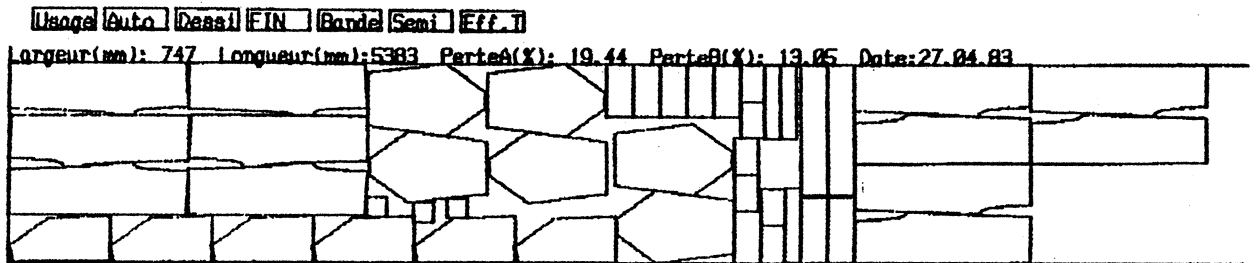
#### 4.3 - Traitement statistique des résultats obtenus

Pour cette étude on a gardé 13 échantillons (de taille entre 92 et 401) de placements de différents modèles, différents nombres de complets à placer dans différentes largeurs de matelas. Notre but est d'estimer statistiquement la perte minimum qu'on puisse obtenir par cette approche et aussi, le nombre d'essais à effectuer pour obtenir un placement de perte proche de ce minimum. Pour cela, il fallait trouver une loi de répartition commune à tous ces échantillons.

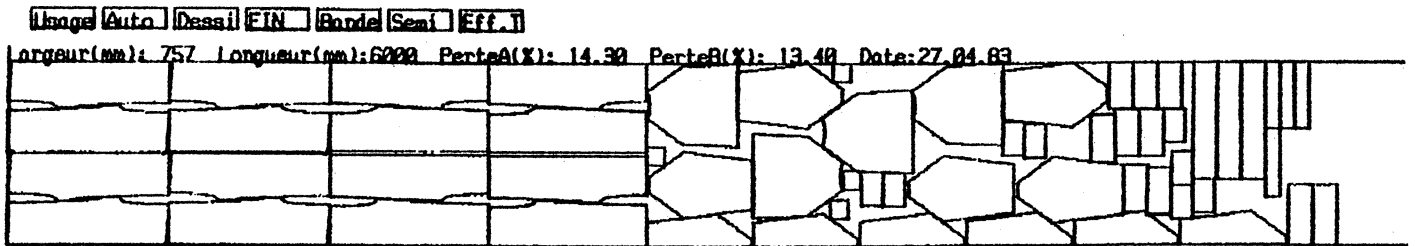
L'étude des histogrammes respectifs nous a suggéré de tester trois lois : une loi gamma translatée, une loi de Weibull translatée et une loi de répartition des valeurs extrêmes. La fig. 4.5 présente un histogramme type et les



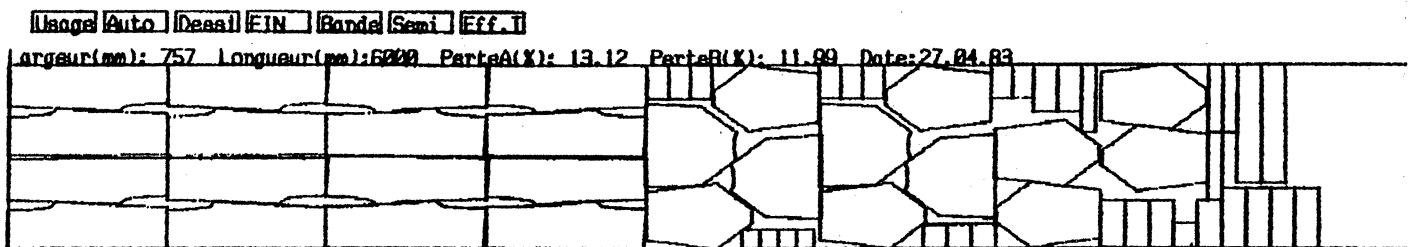
a



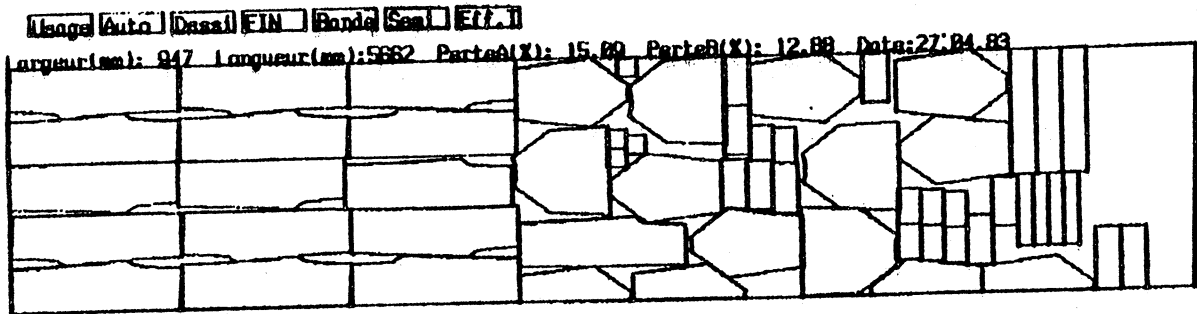
b



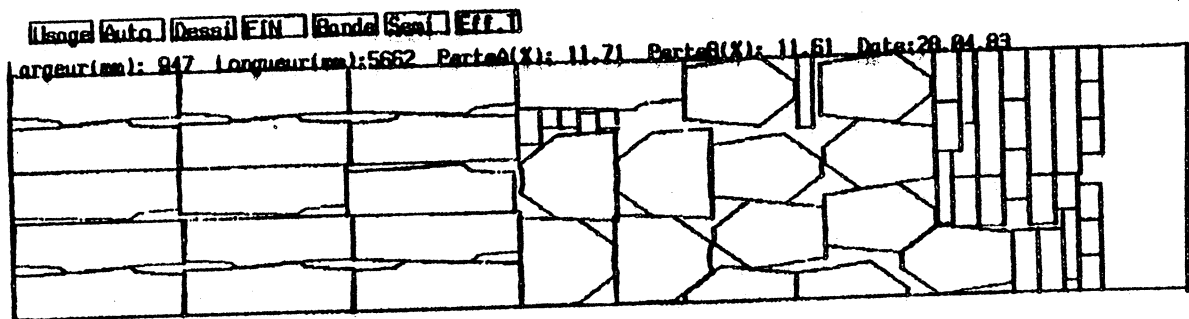
c



d



e

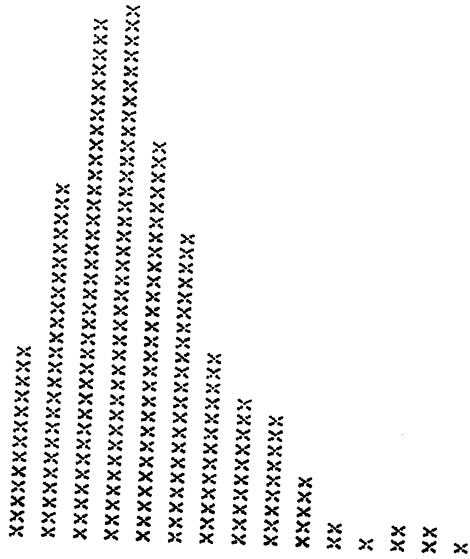


f

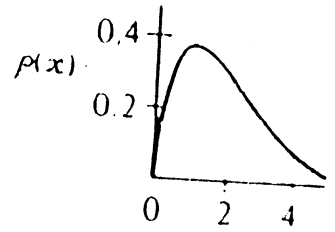
fig. 4.4

a,c,e - placements automatiques

b,d,f - placements par la nouvelle approche

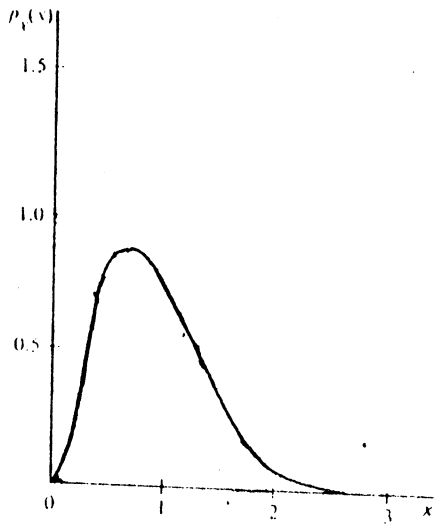


Histogramme res 900

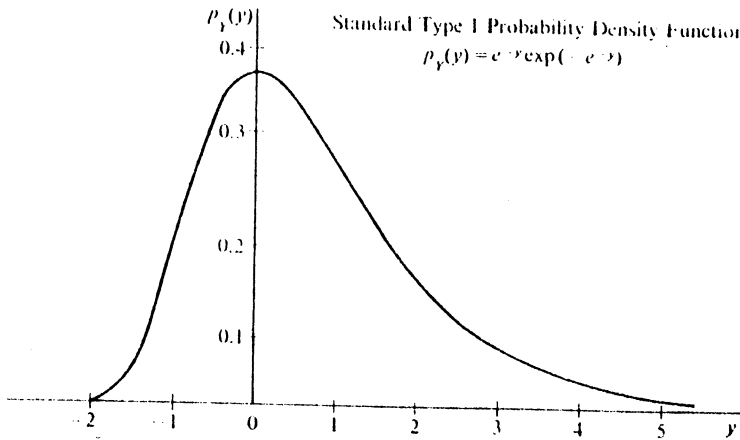


$\beta = 1$

Densite de la loi gamma



Weibull Density Function



Standard Type 1 Probability Density Function  
 $p_v(y) = e^{-y} \exp(-e^{-y})$

FIGURE 4.5.

formes respectives des trois lois considérées.

La référence statistique de base pour ce chapitre est : Johnson et Kotz [1970].

#### 4.3.1 - Ajustement par une loi gamma

La loi gamma testée a une fonction densité de la forme :

$$f_{\lambda}(x) = (x-\gamma)^{\alpha-1} \exp [-(x-\gamma)/\beta] / \beta^{\alpha} \Gamma(\alpha) \quad (4)$$

$$(\alpha > 0, \beta > 0 ; x > \gamma)$$

où :  $\Gamma(\alpha)$  est la fonction gamma.

Bien que nos échantillons soient bornés supérieurement (théoriquement la perte est inférieure de 100 % et pratiquement, on n'a jamais eu une perte supérieure à 40 %), on a pensé qu'une loi gamma avec  $\alpha > 2$  et  $\beta = 1$  pouvait donner des résultats satisfaisants.

Pour estimer les paramètres  $\alpha$ ,  $\beta$ ,  $\gamma$ , on a utilisé la méthode de maximum de vraisemblance et la méthode des moments (bien que cette dernière ne soit pas conseillée pour de telles valeurs de  $\alpha$  et  $\beta$  à cause de sa faible précision).

Les estimateurs de maximum de vraisemblance sont obtenus par la solution de système d'équations non-linéaires :

$$\sum_{j=1}^n \ln (X_j - \gamma) - n \ln \beta - n\psi(\alpha) = 0$$

$$\sum_{j=1}^n (X_j - \gamma) - n \alpha \beta = 0$$

$$-\sum_{j=1}^n (X_j - \gamma)^{-1} + n[\beta(\alpha - 1)]^{-1} = 0$$

où :

$X_1, X_2, \dots, X_n$  est l'échantillon, en question et  $\psi(\alpha) = d\Gamma(\alpha) / d\alpha$ .



Nous n'avons pas pu trouver une solution de ce système avec  $\gamma < X_{\min}$  où  $X_{\min} = \min \{X_j\}$ . Alors, suivant la théorie statistique, on a fixé  $\gamma = X_{\min}$  (bien qu'une valeur pareille pour  $\gamma$  ne présente aucun intérêt pour notre étude (cf. 4.3.2) et on a résolu le système pour  $\alpha$  et  $\beta$ .

Les résultats obtenus par cet ajustement donnaient un écart non acceptable : la différence maximale entre la loi de répartition estimée et la loi empirique était de l'ordre de 0.5 dans les meilleurs cas. La méthode des moments donnait un écart encore plus grand. De plus, la loi gamma est difficile à employer pour la suite de notre étude. Par conséquent, on a exclu cette loi des considérations ultérieures.

#### 4.3.2 - Ajustement par une loi de Weibull

La loi de Weibull  $W(\alpha, \gamma, c)$  a une fonction densité de la forme :

$$P_X(x) = c \cdot \alpha^{-1} [(x - \gamma)/\alpha]^{c-1} \exp[-((x - \gamma)/\alpha)^c] \quad (5)$$

et une fonction de répartition de la forme :

$$F_X(x) = \Pr(X < x) = 1 - \exp[-((x - \gamma)/\alpha)^c] \quad (6)$$

Outre sa ressemblance avec les histogrammes en question, cette loi est intéressante par le fait qu'elle est une des lois de répartition des valeurs extrêmes.

Plus précisément, si  $X_1, X_2, \dots, X_n$  sont des variables aléatoires indépendantes, chacune suivant la même loi de Weibull  $W(\alpha, \gamma, c)$  et

$$Y = \min(X_1, X_2, \dots, X_n) \quad (7)$$

alors,

$$\Pr(Y > y) = [\Pr(X_1 > y)]^n = \exp[-n((y - \gamma)/\alpha)^c] \quad (8)$$

Ainsi  $Y$  suit une loi de Weibull  $W(\alpha n^{-1/c}, \gamma, c)$ . Cette propriété est très importante puisque c'est la loi de

répartition du minimum d'un échantillon qui nous intéresse dans notre étude. On utilisera cette propriété en 4.4.

Les estimateurs de maximum de vraisemblance pour  $\alpha$ ,  $\beta$ , et  $\gamma$  sont obtenus par la solution du système :

$$\alpha = [n^{-1} \sum_{i=1}^n (x_i - \gamma)^c]^{1/c}$$

$$c = [(\sum_{i=1}^n (x_i - \gamma)^c \cdot \ln(x_i - \gamma)) \cdot (\sum_{i=1}^n (x_i - \gamma)^{-1})^{-n^{-1}} \sum_{i=1}^n \ln(x_i - \gamma)]^{-1}$$

$$(c-1) \sum_{i=1}^n (x_i - \gamma)^{-1} = c \cdot \alpha^{-c} \sum_{i=1}^n (x_i - \gamma)^{c-1}$$

Puisque la solution de ce système donne une valeur  $\gamma > X_{\min}$ , on a fixé  $\gamma = X_{\min}$  et on a utilisé la deuxième équation pour trouver  $c$  et la première pour calculer  $\alpha$ .

Les résultats de cet ajustement sont présentés dans le tableau ci-dessous :

Echan.	n	c	$\alpha$	$\gamma$	$d_{\max}$	K <sub>0.2</sub>
res 1	137	1.69	5.36	13.05	0.025	0.091
res 57	401	2.05	3.94	11.99	0.040	0.053
res 67	401	2.06	3.82	13.05	0.026	0.053
res 775	301	1.90	5.74	11.96	0.050	0.062
res 780	301	1.87	4.24	15.10	0.080	0.062
res 795	301	4.01	10.09	13.97	0.040	0.094*
res 800	201	1.52	4.58	12.19	0.030	0.075
res 850	301	1.73	4.89	16.05	0.040	0.062
res 947	92	1.66	4.15	11.60	0.070	0.110
res 950	301	2.23	5.94	14.52	0.080	0.062
res 970	301	1.80	4.31	15.92	0.040	0.062
res 825	301	2.78	9.49	13.30	0.003	0.062
res 855	103	1.62	5.25	15.70	0.080	0.105

La colonne  $d_{\max}$  représente la différence maximale (en valeur absolue) entre la loi de Weibull estimée et la loi de répartition empirique. La colonne  $K_{0.2}$  donne le niveau de signification du test de Kolmogorov - Smirnov pour la valeur  $\alpha = 0.2$ .

L'interprétation de  $K_{\alpha}$  est la suivante :

La différence  $d_{\max}$  est une variable aléatoire qui suit une loi connue (relative à chi-deux à deux degrés de liberté pour les grands échantillons). La probabilité que cette variable prenne une valeur supérieure à  $K_{\alpha}$  est égale à  $\alpha$ .

En d'autres termes,  $K_{\alpha}$  est le  $(1-\alpha)$ -quantile pour la loi de  $d_{\max}$ . Pour  $\alpha$  petit, si on trouve une valeur  $d_{\max} > K_{\alpha}$ , l'adéquation est rejetée. Pour  $\alpha = 0.2$ , si  $d_{\max} < K_{\alpha}$ , l'adéquation est acceptée avec une probabilité d'erreur  $p=\alpha=0.2$ .

Le tableau montre que les adéquations respectives sont confirmées par le test de Kolmogorov-Smirnov.\*)

La figure 4.6 représente la loi de répartition empirique et la loi de Weibull ajustée pour l'échantillon res 67.

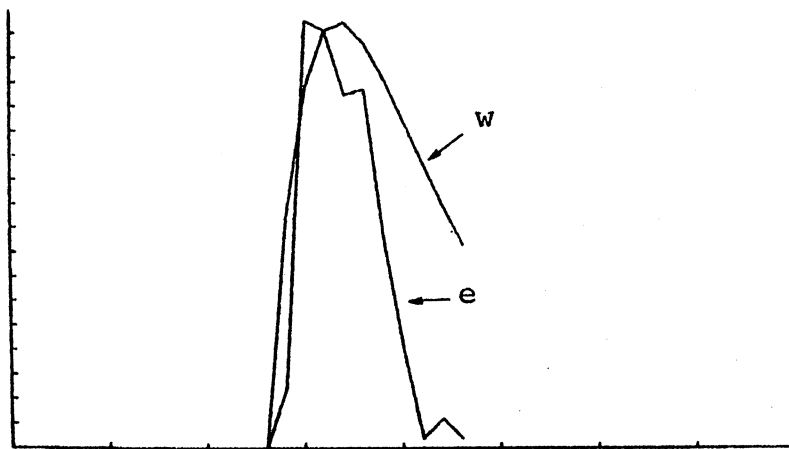


Fig. 4.6 e - empirique  
w - Weibull

Cette approche a le défaut d'estimer la perte minimum qu'on peut obtenir ( $\gamma$ ) par la perte minimale observée ( $X_{\min}$ )

\*) Il faut noter que, dans notre cas où les paramètres de la loi de Weibull sont estimés, le test de Kolmogorov-Smirnov est beaucoup moins puissant et il confirme plus facilement l'adéquation.

Or, en pratique, c'est justement ce paramètre qui nous intéresse le plus et on sait que ce paramètre est toujours plus petit que la valeur minimum observée (n'importe quel placement automatique peut toujours être amélioré).

Etant donné que  $\gamma$  représente le pourcentage de perte du meilleur placement, nous ne sommes intéressés que par la partie de la loi proche de  $\gamma$  (disons entre  $\gamma$  et  $\gamma+2$  ou  $\gamma+3$ ). Pour estimer cette partie de la loi, nous avons estimé  $\gamma$  en fixant les valeurs de  $c$  et  $\alpha$ . Dans ce cas, un estimateur sans biais pour  $\gamma$  est donné par la formule :

$$\gamma = X_{\min} - \alpha \cdot (n+1)^{-1/c}$$

Dans notre approche on a fixé  $c = 2$ . La valeur de  $\alpha$  est déterminée par un balayage de l'intervalle  $[4, 25]$ . Les meilleurs résultats globaux sont obtenus pour  $\alpha = 7$  bien que l'on puisse accepter des valeurs différentes de  $\alpha$  pour différents échantillons.

Les résultats de cette estimation sont présentés dans le tableau suivant :

lchan.	$X_{\min}$	$\gamma$	$\alpha$	$d_{\max}$	$K_{0.2}$
res 1	13.05	12.50	7	0.038	0.091
res 57	11.99	11.66	7	0.052	0.053
res 67	13.05	12.81	5	0.042	0.053
res 775	17.96	17.59	7	0.039	0.062
res 780	15.10	14.73	7	0.027	0.062
res 795	13.98	13.61	7	0.086	$K_{0.01} = 0.094$
res 800	12.19	11.84	5	0.029	0.075
res 850	16.05	15.76	5	0.055	0.062
res 947	11.60	10.93	7	0.034	0.11
res 950	14.52	14.15	7	0.065	$K_{0.15} = 0.065$
res 970	15.92	15.55	7	0.070	$K_{0.1} = 0.07$
res 825	13.31	12.94	7	0.080	$K_{0.01} = 0.094$
res 855	15.70	15.06	7	0.048	0.105

Pour toutes les lois, la portion  $\Pr (X < \gamma + 3)$  varie entre 0.16 et 0.38. Ces ajustements sont acceptables par le test de Kolmogorov-Smirnov comme le montre la dernière colonne.

#### 4.3.3 - Ajustement par une loi de répartition des valeurs extrêmes

La loi en question a une fonction de répartition de la forme :

$$\Pr (X < x) = \exp (A) \quad (9)$$

où

$$A = -\exp[-(x-\gamma) / \theta]$$

C'est la loi la plus utilisée pour l'étude de la répartition des valeurs extrêmes.

Si  $X_1, X_2, \dots, X_n$  sont des variables aléatoires indépendantes, chacune suivant la même loi (9) et

$$Y = \min (X_1, X_2, \dots, X_n) \text{ alors} \\ \Pr (Y > x) = [\Pr (X_i > x)]^n = [1 - \exp(A)]^n \quad (10)$$

Toutes les lois (9) ont la même forme mais sont plus ou moins aplaties suivant la valeur de  $\theta$  et translatées suivant la valeur de  $\gamma$

La fonction de densité de (9) est :

$$P_x(x) = -\theta^{-1} A \exp(A) \quad (11)$$

Elle est unimodale (le maximum est au point  $x = \gamma$ ) et "commence" au point  $x_0 = \gamma - 2.\theta$  .

L'estimateur de maximum de vraisemblance de  $\theta$  est la solution de l'équation :

$$\theta = n^{-1} \sum_{i=1}^n X_i - \left[ \sum_{i=1}^n X_i \exp(-X_i/\theta) \right] \left[ \sum_{i=1}^n \exp(-X_i/\theta) \right]^{-1}$$

Une bonne approximation est donnée par la formule :

$$\theta = n^{-1} \sum_{i=1}^n X_i + n^{-1} \sum_{i=1}^n X'_i \ln[(j - 0.5)/(n + 0.5)]$$

où  $\{X'_i\}$  est l'échantillon  $\{X_i\}$  trié.

L'estimateur de maximum de vraisemblance de  $\gamma$  est calculé selon la formule :

$$\gamma = \theta \cdot \ln[n^{-1} \sum_{i=1}^n \exp(-x_i / \theta)]$$

Les résultats d'ajustement par cette loi sont présentés dans le tableau suivant :

Echan.	n	$\theta$	$\gamma$	$X_{\min}$	$X_0$	$d_{\max}$	$K_{0.2}$
res 1	137	2.22	16.53	13.05	12.08	0.066	0.91
res 57	401	1.37	14.67	11.99	11.93	0.045	0.053
res 67	401	1.39	15.63	13.05	12.85	0.050	0.053
res 775	301	2.06	21.90	17.96	17.79	0.070	$K_{0.1}=0.0703$
res 780	301	1.51	17.97	15.10	14.95	0.060	0.062
res 795	301	2.22	21.62	13.98	17.19	0.220	0.062
res 800	201	2.06	15.11	12.19	11.00	0.021	0.075
res 850	301	2.03	19.24	16.04	15.18	0.048	0.062
res 947	92	1.69	14.33	11.60	10.94	0.040	0.110
res 950	301	1.93	18.68	14.52	14.83	0.040	0.062
res 970	301	1.73	18.74	15.92	15.29	0.030	0.062
res 825	301	2.76	20.11	13.31	14.59	0.095	0.062
res 855	103	2.17	19.11	15.70	14.77	0.14	$K_{0.01}=0.161$

Ces ajustements (sauf res 795, res 950, res 825) sont confirmés par le test de Kolmogorov-Smirnov avec  $\alpha = 0.2$  (0.1 pour res 775 et 0.01 pour res 855) (cf. 4.3.2).

Pour les trois cas "pathologiques" (res 795, res 950 et res 825), l'estimation a donné une valeur  $X_0 > X_{\min}$ . Dans ce cas, on a fixé  $X_0 = X_{\min}$  et, avec le même  $\theta$ , on a calculé  $\gamma = X_{\min} + 2\theta$ . Cette modification ajuste la partie "intéressante" de la loi avec un niveau de signification relatif à 0.01 (voir tableau suivant).

Echan.	n	$\theta$	$\gamma$	$X_{\min}$	$X_0$	$d_{\max}^{3\%}$	$d_{\max}^{2\%}$	
res 795	301	2.22	21.62	13.98	13.98	0.14	0.044	$K_{0.01}=0.094$
res 950	301	1.93	18.68	14.52	14.52	0.08	0.044	$K_{0.01}=0.094$
res 825	301	2.76	20.11	13.31	13.31	0.04	0.017	$K_{0.2}=0.062$

La fig. 4.7 représente la loi de répartition empirique et la loi de répartition de valeurs extrêmes ajustée pour l'échantillon res 800.

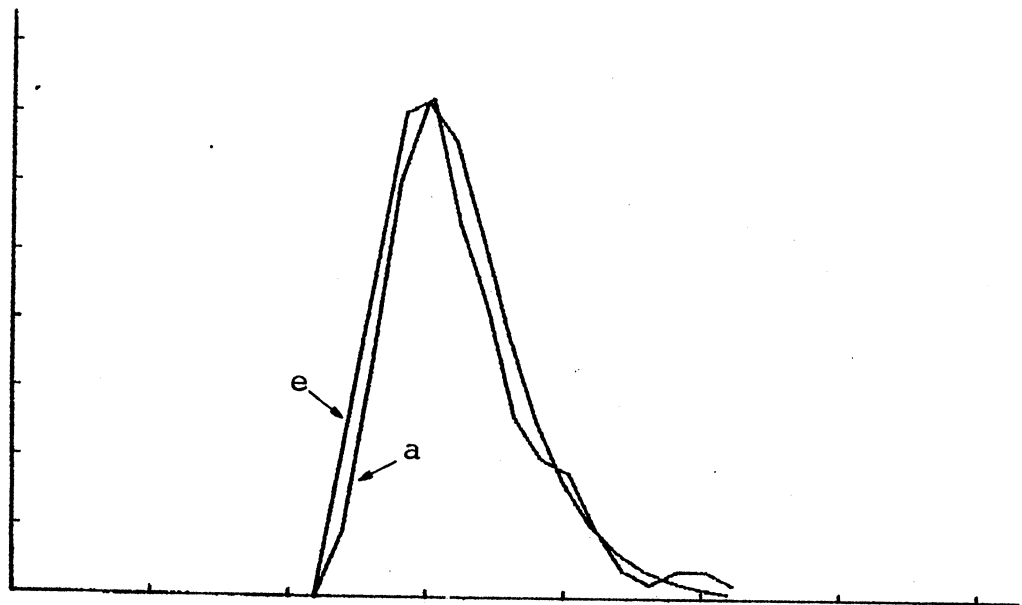


Fig. 4.7

e - empirique

a - ajustée

De toutes les trois lois testées, les meilleurs résultats sont obtenus par la loi de répartition des valeurs extrêmes. Pour les études suivantes, on va considérer uniquement cette loi.

#### 4.4 - Estimation de nombre d'essais à effectuer

Les ajustements confirmés en 4.3.3 nous permettent d'estimer le nombre d'essais à effectuer pour obtenir un placement au voisinage ((du point de vue perte) du "meilleur" placement.

La formule (10) nous donne la fonction de répartition de la valeur minimale d'un échantillon distribué selon une loi de valeurs extrêmes. Si on veut que cette valeur soit dans l'intervalle  $(x_0, x_0 + a)$  (où  $a \in (1,2)$  c.à.d. on veut obtenir un placement avec une perte 1-2 % plus grande que celle du meilleur placement), avec une probabilité  $p$  proche de 1, par exemple  $p = 0.9$ , on a :

$$\Pr (Y < x) > p = \Pr (Y > x) < 1 - p$$

Plus précisément :

$$B^n < 1 - p$$

où

$$B = 1 - \exp[-\exp(-(x-\gamma) / \theta)]$$

D'où

$$n > \ln(1-p) / \ln B \quad (11)$$

Pour  $p = 0.9$ , on a :

$$n > -2.303 / \ln B \quad (12)$$

Pour obtenir une perte supérieure de  $a$  de la perte minimum, il faut calculer la partie droite de (12) dans le point  $x = x_0 + a = \gamma - 2\theta + a$ , ce qui donne :

$$n_{a\%} > -2.303 / \ln[1 - \exp(-\exp(2 - a/\theta))]$$

Les valeurs de  $n_{a\%}$  pour  $a = 1, 1.5$  et  $2$  ainsi que les valeurs observées en pratique sont présentées dans le tableau suivant :

Echan.	$x_0$	$p = 0.9$			XP		
		$n_{1\%}$	$n_{1.5\%}$	$n_{2\%}$	1%	1.5%	2%
res 1	12.08	256	99	46	$x_{56} = 13.05$	$x_{12} = 13.42$	$x_{12} = 13.42$
res 57	11.93	80	27	12	$x_{65} = 12.64$	$x_7 = 13.36$	$x_7 = 13.36$
res 67	12.85	82	28	13	$x_2 = 13.82$	$x_2 = 13.82$	$x_2 = 13.42$
res 775	17.79	217	81	37	$x_{108} = 17.96$	$x_{29} = 18.87$	$x_{29} = 18.87$
res 780	14.95	103	35	16	$x_2 = 15.84$	$x_2 = 15.84$	$x_2 = 15.84$
res 795	13.98	255	98	45	$x_{108} = 13.98$	$x_{108} = 13.98$	$x_{108} = 13.98$
res 800	11.00	217	81	37	$x_{188} = 12.19$	$x_{188} = 12.19$	$x_{15} = 12.64$
res 850	15.18	210	78	36	$x_{116} = 16.05$	$x_{25} = 16.52$	$x_{25} = 16.52$
res 947	10.94	137	48	21	$x_3 = 11.60$	$x_3 = 11.60$	$x_3 = 11.60$
res 950	14.52	187	68	31	$x_{26} = 15.43$	$x_{26} = 15.43$	$x_{26} = 15.43$
res 970	15.29	145	51	24	$x_{36} = 16.28$	$x_2 = 16.46$	$x_2 = 16.46$
res 825	13.31	394	168	82	$x_{106} = 13.31$	$x_{106} = 13.31$	$x_{49} = 15.25$
res 855	14.77	243	93	43	$x_{103} = 15.70$	$x_5 = 15.96$	$x_5 = 15.96$

Dans ce tableau, la colonne  $XP_{a\%}$  ( $a=1, 1.5, 2$ ) donne la première valeur observée dans l'intervalle  $(x_0, x_0 + a)$



4.5 Utilisation pratique

Les ajustements décrits en 4.3.3 nous permettent d'envisager l'utilisation pratique suivante :

On effectue un placement automatique et un certain nombre de placements aléatoires. A partir de l'échantillon obtenu on estime les paramètres de la respectives et, en particulier, le  $X_0$ . On calcule le nombre d'essais à effectuer pour obtenir un placement dans l'intervalle  $(X_0, X_0 + a)$  (par exemple  $a = 1 \%$ )

Si ce nombre est acceptable, compte tenu du temps de calcul que l'on est disposé à payer, on continue les essais sinon, on s'arrête. Dans les deux cas, on garde le meilleur placement obtenu.

Pour valider cette approche, on l'a appliquée sur les échantillons déjà considérés. On a effectué 20 placements et ensuite on a estimé le nombre d'essais nécessaires pour obtenir un placement dans l'intervalle  $(X_0, X_0 + 1)$ . Les résultats de cette étude sont présentés dans le tableau suivant :

Echan.	$\theta$	$\gamma$	$X_0$	$X_{\min}$	$n_{1\%}$ ( $p=0.9$ )	$X_{1\%}$	$X_{1\%}^p$
res 1	1.36	15.48	12.76	13.42	-	-	-
res 57	1.19	14.81	12.28	13.36	58	13.28	$X_{64} = 12.64$
res 67	1.32	15.36	12.71	13.82	73	13.71	$X_{86} = 13.05$
res 775	1.15	22.69	20.38	21.59	51	21.38	$X_{28} = 18.86$
res 780	1.77	18.31	14.77	15.84	153	15.77	$X_{162} = 15.10$
res 795	1.47	22.30	19.65	19.36	-	-	-
res 800	1.33	14.98	12.31	12.64	-	-	-
res 850	1.24	19.68	17.20	17.76	-	-	-
res 947	1.39	13.58	10.79	11.60	-	-	-
res 950	1.29	18.45	15.86	17.20	69	16.86	$X_{25} = 15.43$
res 970	1.73	18.75	15.30	16.46	144	16.30	$X_{35} = 16.28$
res 825	2.02	19.84	15.80	17.11	208	16.80	$X_{21} = 15.42$
res 855	2.04	19.25	15.16	15.96	-	-	-

Les lignes non remplies de ce tableau correspondent aux échantillons dont la valeur minimale est, après 20 essais,

dans l'intervalle  $(x_0, x_0 + 1)$

Il faut noter que le résultat  $x_0 > x_{\min}$  indique un mauvais placement automatique et que la poursuite des essais peut améliorer de beaucoup le meilleur placement obtenu jusque là.

### 3.6 Conclusions

Dans ce chapitre, on a présenté l'utilisation de méthodes stochastiques combinées avec un programme de placement automatique.

Cette nouvelle approche nous donne une possibilité d'améliorer le placement automatique dans des problèmes où le temps de calcul n'est pas critique. En outre, elle nous permet d'estimer la perte minimum que l'on peut obtenir et le nombre d'essais à effectuer pour obtenir un placement de perte donnée proche de la perte minimum.

Les expérimentations pratiques confirment la validité de cette approche.



## BIBLIOGRAPHIE

On propose ici une vaste bibliographie consacrée au problème de découpe et ses dérivées. Comme nous l'avons précisé en Introduction, quelques articles ne sont pas cités directement dans cette thèse, mais nous avons considéré opportun de profiter de l'occasion pour donner une bibliographie complète, comprenant les publications les plus récentes.

Afin d'en faciliter l'utilisation, pour chaque publication, on donne une clé relative au type de problème traité, à la méthode utilisée et à (aux) l'industrie(s) d'application(s). La clé est de la forme :

- \* ) type de problème
- \*\* ) méthode de résolution
- \*\*\* ) industrie d'application

Le premier terme existe toujours (hormis quelques publications de caractère général comme Kirkpatrick [1982], etc).

Le deuxième ou troisième terme peut être absent.

Le troisième terme se réfère à l'industrie d'application citée dans l'article respectif.

Pour former cette clé, nous avons utilisé le thésaurus ci-dessus :

<u>Type de problème</u>	<u>Méthode d'utilisation</u>
CS - Contraintes spéciales	BB - branch and bound
D1D - Découpe d'une dimension et demie	BP - bin-packing
DB - découpe bidimensionnelle	EA - exploration arborescente
DF - demande faible	GC - génération de colonnes
DS - découpe en stades	H - heuristique
DU - découpe unidimensionnelle	IA - intelligence artificielle

FD - pièces identiques	MC - méthode de Monte-Carlo
FI - pièces de formes irrégulières	PO1- programmation en 0-1
NG - découpe non-guillotine	PD - programmation dynamique
PS - problème stochastique	PE - programmation en nombres entiers
	PL - programmation linéaire
	PN - programmation non-linéaire
	PR - procédure récursive
	PT - problème de transport
	SD - problème de sac-à-dos
	SG - méthode de sous gradient
	VC - problème de voyageur de commerce

### Industries d'applications

A - acier  
B - bois  
E - emballage  
M - moquette  
P - papier  
T - tôle  
V - verre  
VT - vêtement

- 1 - Adamovicz M., Albano A. [1976 a], Nesting two-dimensional shapes in rectangular modules. Computer Aided Design 8 (N° 1), 27-33, January 1976
  - \*) DB, FI
  - \*\*) H
  - \*\*\*) T
  
- 2 - Adamovicz M., Albano A. [1976 b], A solution of the rectangular cutting stock-problem. IEEE Trans. on Systems, Man and Cybernetics SMC-6 (N°4), 302-310, April 1976.
  - \*) DB
  - \*\*) PD, H
  - \*\*\*) T
  
- 3 - Albano A. [1977], A method to improve two-dimensional layout. Computer Aided Design 9, 48-52
  - \*) DB, FI
  - \*\*) H
  - \*\*\*) T
  
- 4 - Albano A., Orsini R. [1980], A heuristic solution of the rectangular cutting-stock problem. The Computer Journal 23 (N°4), 338-343, November 1980
  - \*) DB
  - \*\*) PD, EA, H
  - \*\*\*) T
  
- 5 - Albano A., Sapuppo G. [1980], Optimal allocation of two-dimensional irregular shapes using heuristic search methods. IEEE Trans. on Systems, Man and Cybernetics SMC-10 (N°5), 242-248, May 1980
  - \*) DB, FI
  - \*\*) H, IA
  - \*\*\*) T

- 6 - Barnett S., Kynch G. [1976], Exact solution of a simple cutting problem. Opns. Research 15, 1051-1056  
\*) DB, FD  
\*\*\*) E
- 7 - Bengtsson B.E. [1982], Packing rectangular pieces - a heuristic approach. The Computer Journal 25 (N°3), 353-357  
\*) DB, NG  
\*\*) H
- 8 - Bischoff E.E. [1982], Generating Stacking Patterns for pallet loading. Presented at EURO-V-TIMS XXV Lausanne, July 12-14, 1982  
\*) DB, FD  
\*\*\*) E
- 9 - Bischoff E.E., Downsland W. B. [1982], An application of the micro to product design and distribution. Journal of the Operational Research Society, vol. 33, 271-280  
\*) DB, FD  
\*\*\*) E
- 10 - Böhme D., Graham A. [1979], Practical experiences with semi-automatic and automatic partnesting methods Computer Applications in the Automation of Shipyard Operation and Ship Design. Kuo C., MacCallum K.J., Williams T.J. (eds.) vol. 3, 213-220. IFIP, North-Holland (1979)  
\*) DB, FI  
\*\*) MC  
\*\*\*) T

- 11 - Chambers M.L., Dyson R.G. [1976], The cutting-stock problem in the flat glass industry - Selection of stock sizes.  
Operational Res. Quart. 27, 949-957  
\*) DU + CS  
\*\*\*) V
- 12 - Christofides N. [1974], Optimum cutting of 2D-rectangular shapes CAD'74 ; International Conference on Computers in Engineering and Building Design. IPC Science & Technology Press LTD., Guildford, Surrey, England.  
\*) D1D  
\*\*) PN, H
- 13 - Christofides N., Whitlock C. [1977], An algorithm for two dimensional cutting problems. Operations Res. 25, 30-44.  
\*) DB, DF  
\*\*) PD, PT, BB
- 14 - Costa M.C. [1980], Problemes de découpes linéaires ; formalisation et solutions économiques. Thèse de docteur de 3ème cycle. Université Pierre et Marie Curie, Paris 6, Juin 1980  
\*) DU, DS, CS, PS  
\*\*) PL, PD, BB, PO1, H  
\*\*\*) A, B, P, V, M
- 15 - Costa M.C. [1982], Formalisation et résolution des problèmes de découpes linéaires. RAIRO Rech. Opérat. vol.16 N°1 65-82  
\*) DU, PS  
\*\*) PL, PD, BB, PO1, H  
\*\*\*) A, B, P, V, M



- 16 - Coverdale I., Wharton F. [1976], An improved heuristic procedure for a nonlinear cutting stock problem. Management Science 23, 78-86  
\*) DU, CS  
\*\*) H
- 17 - De Cani P. [1978], A note on the two-dimensional rectangular cutting stock problem. J. Oper. Res. Soc. 29, 703-706  
\*) DB, CS
- 18 - Dyson R.G., Gregory A.S. [1974], The cutting stock problem in the flat glass industry. Operational Res. Quart. 25, 41-53.  
\*) DB, CS  
\*\*) PL, PD, VC  
\*\*\*) V
- 19 - Eilon S., Christofides N. [1971], The loading problem Management Science 17, 259-268  
\*) DU  
\*\*) PO1, H  
\*\*\*) E
- 20 - Garfinkel R.S. [1977], Minimizing wallpaper waste, part.1 a class of travelling salesman problems. Operational Research, vol 25-5, 741-751  
\*) DU, CS  
\*\*) VC  
\*\*\*) P
- 21 - Gilmore P.C., [ ], Cutting stock, linear programming knapsacking, dynamic programming and integer programming, some interconnections.  
\*) DU,  
\*\*) PL, PD, PE, SD

- 22 - Gilmore P.C., Gomory R.E. [1961], A linear programming approach to the cutting-stock problem. Operations Research 9, 849-859  
\*) DU, CS  
\*\*) PL, GC, SD, PD  
\*\*\*) P
- 23 - Gilmore P.C., Gomory R.E. [1963], A linear programming approach to the cutting-stock problem, Part III Operations Research 11, 863-888  
\*) DU, CS  
\*\*) PL, GC, SD, EB  
\*\*\*) P
- 24 - Gilmore P.C., Gomory R.E. [1965], Multistage cutting-stock problems of two and more dimensions. Operations Research 13, 94-120  
\*) DB, DS  
\*\*) PL, GC, PD  
\*\*\*) P
- 25 - Gilmore P.C., Gomory R.E. [1966], The theory and computation of knapsack functions. Operations Research 14, 1045-1074  
\*) DU, DB  
\*\*) PL, GC, PD
- 26 - Golden B. [1976], Approaches to the cutting stock problem AIIE Transactions 8, 265-274  
\*) DU, DB  
\*\*) PL, PO1, BP, SG  
\*\*\*) P, E
- 27 - Gribov A.B. [1973], Algorithm for solving the problem of a plane cutting layout. Cybernetics 9, 1036-1043  
\*) DB  
\*\*) PL, PD

- 28 - Haessler R.W. [1971], An heuristic programming solution to a non-linear cutting stock problem. Management Science 17, B 793-B 802  
\*) DU, CS  
\*\*) H
- 29 - Haessler R.W. [1975], Controlling cutting pattern changes in one-dimensional trim problems. Operations Research 23, 483-493  
\*) DU, CS  
\*\*) H
- 30 - Haessler R.W. [1977], A procedure for solving 1.5 dimensional coil slitting problem. AIIE Transactions vol.10 N°1, 70-75  
\*) D1D, CS  
\*\*) H  
\*\*\*) A
- 31 - Haessler R.W., Vonderembse M.A. [1979a], A procedure for solving the master slab cutting stock problem in the steel industry. AIIE Transactions, vol.11 N° 2, 160-165  
\*) DU  
\*\*) PL, GC, H  
\*\*\*) A
- 32 - Haessler R. W. [1979b], Solving two-stage cutting stock problem.  
Omega vol.7, N° 2, 145-151  
\*) DB, DS, CS  
\*\*) PL, H
- 33 - Haessler R.W., Vonderembse M.A. [1982], A mathematical programming approach to schedule master slab casters in the steel industry. Mgmt Science vol. 28, N° 12, 1450-1461  
\*) DU, CS  
\*\*) PL  
\*\*\*) A

- 34 - Haessler R.W., Talbot B.F. [1983], A 0-1 model for solving the corrugator trim problem. *Mgmt Science* vol. 29, N° 2, 200-209  
\*) D1D, CS  
\*\*) P01  
\*\*\*) P
- 35 - Hahn S.G. [1968], On the optimal cutting of defective sheets. *Operations Research* 16, 1100-1114  
\*) DB, DS  
\*\*) PL, GC, SD, H  
\*\*\*) V
- 36 - Haims M.J., Freeman H. [1970], A multistage solution to the template layout problem. *IEEE Trans. Systems Science, Cybernetics, SCC-6*, 145-151  
\*) DB  
\*\*) PD, H
- 37 - Herz J.C. [1972], Recursive computational procedure for two-dimensional stock cutting. *IBM Journal Res. Develop.* 16, 462-469  
\*) DB  
\*\*) PR
- 38 - Hinxman A.I. [1977], A two-dimensional trim-loss problem with sequencing constraints. *Advance papers of IJCAI-77, M.I T. (1977)*, 859-864  
\*) DB, CS  
\*\*) IA  
\*\*\*) V
- 39 - Hinxman A.I. [1980], The trim-loss and assortment problems. *European Journal of Oper. Res.* 5 (1980) 8-18 (review)

- 52 - Metropolis N, Rosenbluth A., Rosenbluth M., Teller A.,  
Teller E [1953], Equation of State calculations  
by fast computing machines. J Chem. Phys. 21  
1087-1092
- 53 - Pfefferkorn C.E. [1975], An heuristic problem solving  
design system for equipment or furniture layouts  
CACM 18, 286-297  
\*) DB  
\*\*) H, IA
- 54 - Rvachev V.L., Stoyan Yu G. [1965], Algorithm for solution  
of the problem of optimum layout of circular  
patterns with restrictions on the distance  
between patterns. Kibernetika, vol. 1, 77-83  
\*) DB  
\*\*) MC, PN
- 55 - Sculli D. [1981], A stochastic cutting stock procedure:  
cutting rolls of insulating tape. Mgmt Science  
vol. 27, N°8, 946-952  
\*) DU, CS  
\*\*\*) V, P, M
- 56 - Soenen R. [1977], Conduite automatisée d'une unité de  
coupe en confection. Automatisme, Mai 1977  
\*) D1D, FI  
\*\*) H  
\*\*\*) VT
- 57 - Stainton R.S. [1977], The cutting stock problem for the  
stock-holder of steel reinforcement bars.  
Oper. Res. Quart. 28, 139-149  
\*) DU, CS  
\*\*\*) A

58 - Tilanus C.B., Gerhardt C. [1976], An application of cutting stock in the steel industry. In Haley K.B. (ed.) Operational Research 75 North-Holland, 1976, 669-675

\*) DU

\*\*\*) SD

\*\*\*) A



DERNIERE PAGE D'UNE THESE

3È CYCLE, DOCTEUR INGÉNIEUR OU UNIVERSITÉ

Vu les dispositions de l'arrêté du 16 avril 1974,

Vu les rapports de M. ...M. SAKAROVITCH.....

M. ....

M. ...PREMTI... FREDERIK..... est autorisé  
à présenter une thèse en vue de l'obtention du grade de DOCTEUR *de troisième*  
*cycle*.....

17 JUIN 1988

Grenoble, le

Le Président de l'Université Scientifique  
et Médicale

M. TANCHE

*Avis favorable*

*Sakarovitch*



*Tanche*



