Modeling and optimizing flexible capacity allocation in semiconductor manufacturing

Carl Johnzén

N° d'ordre : 521 GI

THÈSE

présentée par

Carl Johnzén

Pour obtenir le grade de Docteur
de l'Ecole Nationale Supérieure des Mines de Saint-Etienne

Spécialité : Génie industriel

Modeling and Optimizing Flexible Capacity Allocation in Semiconductor Manufacturing

soutenue à Gardanne le 6 avril 2009

Membres du jury

President	Prof. L. Mönch	University of Hagen
Rapporteurs	Prof. M. Jacomino	Institut Polytechnique de Grenoble
	Prof. J. Rooda	Eindhoven University of Technology
Examinateurs	Prof. M.-C. Portmann	Ecole des Mines de Nancy (INPL)
	Prof. M. Sevaux	Université de Bretagne-Sud
Directeur de thèse	Prof. S. Dauzère-Pérès	Ecole des Mines de Saint-Etienne
Tuteur industriel	Ir. P. Vialletelle	STMicroelectronics

Spécialités doctorales :	Responsables :
SCIENCES ET GENIE DES MATERIAUX	J. DRIVER Directeur de recherche – Centre SMS
MECANIQUE ET INGENIERIE	A. VAUTRIN Professeur – Centre SMS
GENIE DES PROCEDES	G. THOMAS Professeur – Centre SPIN
SCIENCES DE LA TERRE	B. GUY Maître de recherche – Centre SPIN
SCIENCES ET GENIE DE L'ENVIRONNEMENT	J. BOURGOIS Professeur – Centre SITE
MATHEMATIQUES APPLIQUEES	E. TOUBOUL Ingénieur – Centre G2I
INFORMATIQUE	O. BOISSIER Professeur – Centre G2I
IMAGE, VISION, SIGNAL	JC. PINOLI Professeur – Centre CIS
GENIE INDUSTRIEL	P. BURLAT Professeur – Centre G2I
MICROELECTRONIQUE	Ph. COLLOT Professeur – Centre CMP

Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

AVRIL	Stéphane	MA	Mécanique & Ingénierie	CIS
BATTON-HUBERT	Mireille	MA	Sciences & Génie de l'Environnement	SITE
BENABEN	Patrick	PR 2	Sciences & Génie des Matériaux	CMP
BERNACHE-ASSOLANT	Didier	PR 0	Génie des Procédés	CIS
BIGOT	Jean-Pierre	MR	Génie des Procédés	SPIN
BILAL	Essaïd	DR	Sciences de la Terre	SPIN
BOISSIER	Olivier	PR 2	Informatique	G2I
BOUCHER	Xavier	MA	Génie Industriel	G2I
BOUDAREL	Marie-Reine	MA	Génie Industriel	DF
BOURGOIS	Jacques	PR 0	Sciences & Génie de l'Environnement	SITE
BRODHAG	Christian	MR	Sciences & Génie de l'Environnement	SITE
BURLAT	Patrick	PR 2	Génie industriel	G2I
COLLOT	Philippe	PR 1	Microélectronique	CMP
COURNIL	Michel	PR 0	Génie des Procédés	SPIN
DAUZERE-PERES	Stéphane	PR 1	Génie industriel	CMP
DARRIEULAT	Michel	ICM	Sciences & Génie des Matériaux	SMS
DECHOMETS	Roland	PR 1	Sciences & Génie de l'Environnement	SITE
DESRAYAUD	Christophe	MA	Mécanique & Ingénierie	SMS
DELAFOSSE	David	PR 1	Sciences & Génie des Matériaux	SMS
DOLGUI	Alexandre	PR 1	Génie Industriel	G2I
DRAPIER	Sylvain	PR 2	Mécanique & Ingénierie	SMS
DRIVER	Julian	DR	Sciences & Génie des Matériaux	SMS
FOREST	Bernard	PR 1	Sciences & Génie des Matériaux	CIS
FORMISYN	Pascal	PR 1	Sciences & Génie de l'Environnement	SITE
FORTUNIER	Roland	PR 1	Sciences & Génie des Matériaux	SMS
FRACZKIEWICZ	Anna	MR	Sciences & Génie des Matériaux	SMS
GARCIA	Daniel	CR	Génie des Procédés	SPIN
GIRARDOT	Jean-Jacques	MR	Informatique	G2I
GOEURIOT	Dominique	MR	Sciences & Génie des Matériaux	SMS
GOEURIOT	Patrice	MR	Sciences & Génie des Matériaux	SMS
GRAILLOT	Didier	DR	Sciences & Génie de l'Environnement	SITE
GROSSEAU	Philippe	MR	Génie des Procédés	SPIN
GRUY	Frédéric	MR	Génie des Procédés	SPIN
GUILHOT	Bernard	DR	Génie des Procédés	CIS
GUY	Bernard	MR	Sciences de la Terre	SPIN
GUYONNET	René	DR	Génie des Procédés	SPIN
HERRI	Jean-Michel	PR 2	Génie des Procédés	SPIN
KLÖCKER	Helmut	MR	Sciences & Génie des Matériaux	SMS
LAFOREST	Valérie	CR	Sciences & Génie de l'Environnement	SITE
LERICHE	Rodolphe	CR	Mécanique et Ingénierie	SMS
LI	Jean-Michel	EC (CCI MP)	Microélectronique	CMP
LONDICHE	Henry	MR	Sciences & Génie de l'Environnement	SITE
MOLIMARD	Jérôme	MA	Mécanique et Ingénierie	SMS
MONTHEILLET	Frank	DR 1 CNRS	Sciences & Génie des Matériaux	SMS
PERIER-CAMBY	Laurent	PR1	Génie des Procédés	SPIN
PIJOLAT	Christophe	PR 1	Génie des Procédés	SPIN
PIJOLAT	Michèle	PR 1	Génie des Procédés	SPIN
PINOLI	Jean-Charles	PR 1	Image, Vision, Signal	CIS
STOLARZ	Jacques	CR	Sciences & Génie des Matériaux	SMS
SZAFNICKI	Konrad	CR	Sciences & Génie de l'Environnement	SITE
THOMAS	Gérard	PR 0	Génie des Procédés	SPIN
VALDIVIESO	François	MA	Sciences & Génie des Matériaux	SMS
VAUTRIN	Alain	PR 0	Mécanique & Ingénierie	SMS
VIRICELLE	Jean-Paul	MR	Génie des procédés	SPIN
WOLSKI	Krzysztof	CR	Sciences & Génie des Matériaux	SMS
XIE	Xiaolan	PR 1	Génie industriel	CIS

Glossaire :

PR 0	Professeur classe exceptionnelle
PR 1	Professeur 1ère catégorie
PR 2	Professeur 2ème catégorie
MA(MDC)	Maître assistant
DR (DR1)	Directeur de recherche
Ing.	Ingénieur
MR(DR2)	Maître de recherche
CR	Chargé de recherche
EC	Enseignant-chercheur
ICM	Ingénieur en chef des mines

Centres :

SMS	Sciences des Matériaux et des Structures
SPIN	Sciences des Processus Industriels et Naturels
SITE	Sciences Information et Technologies pour l'Environnement
G2I	Génie Industriel et Informatique
CMP	Centre de Microélectronique de Provence
CIS	Centre Ingénierie et Santé

Acknowledgments

The work presented in this PhD thesis has been realized within the framework of *Convention de Formation par la Recherche* (CIFRE n 104) in accordance with the *Association Nationale de la Recherche Technique*, which supports companies to involve PhD students in their work. The thesis also has been written as a part of the European Union project HYMNE.

The first two years of the work on this thesis has been done within the framework of Crolles2 Alliance, tripartite joint operation between Freescale Semiconductor, NXP and STMicroelectronics (ST). Since the first of January 2008 the work has been carried out in cooperation with ST as the only partner.

During my thesis writing I was an employee at ST at their site in Crolles, France. This has not only helped to finance my PhD studies but also given me practical support. I have been able to use their resources in my work such as an office space, computer and more important data from their production for the tests that I have performed. The administrative aid from ST has made it possible for me to concentrate on my research. I am very grateful for the assistance of STMicroelectronics. I especially want to express my gratitude toward Philippe Vialletelle my supervisor from the Industrial Engineering group at STMicroelectonics in Crolles, who shared his experiences and ideas. Although overloaded with work, his clear-sighted ideas always helped me to move my research forward.

Without mentioning all my colleagues at ST, I would like to at least mention the guidance and friendship that has been offered to me by François Buttin, Mariangela Cardille, Manuel Cali, Leon Vermariën and my fellow PhD students Jean-Etienne Kiba and Casper Veeger.

The scientific part of the research has been done within the framework of *Centre Microéléctronique de Provence - Georges Charpak* (CMP) at *Ecole Nationale Supérieure des Mines de Saint-Etienne* (EMSE) in Gardanne, France. When STMicroelectronics got into a difficult financial situation EMSE continued to finance my visits at their research center and to conferences for which I am very grateful.

I'm very thankful to my scientific supervisor, professor Stéphane Dauzère-Pérès.

His guidance and expertise within the operational research area has helped me to raise the quality of my research several levels. I truly appreciate his ability to scrutinize data. He could always identify something that did not work optimally, recommend which methods should be used for solving my models, find bugs in my computer programs, and suggest better ways to write articles or make presentations.

At the *Sciences de la fabrication et logistique*-department (SFL) at CMP I also want to thank Claude Yugma with whom I have been able to discuss my ideas, and Alexandre Derreumaux who has helped me a great deal with the programming performed during my research. Without mentioning all the names, I want to express my gratitude to the rest of the staff at SFL-department. I have always appreciated the warm and welcoming atmosphere that I found there. Finally, I want to thank my fiancée Veronika Gumpinger, who has been the greatest possible moral support.

Table des matières

Allocation flexible des capacités pour la fabrication de semi-conducteurs : Modélisation et optimisation

0.1 Introduction

Cette partie constitue un résumé étendu en français de ma thèse. Elle comportera un bref résumé de chacune de ses parties. Je mettrai l'accent sur la présentation de mes travaux, je dirai le pourquoi, le comment et le cadre d'utilisation du développement des modèles. Les modèles seront détaillés mais aucune démonstration ne sera présentée ici (cf. l'intégralité de la thèse).

Mes travaux de recherche ont été menés dans une usine de fabrication de semi-conducteurs et, c'est donc naturellement ceux-ci qui ont fait l'objet de mon étude. Toutefois, certains des problèmes étudiés peuvent se rencontrer dans d'autres industries et domaines. Par conséquent, j'ai tenté de présenter mes travaux dans un cadre le plus général possible.

Plan de lecture

Je commencerai par présenter le cadre d'étude de ma thèse, en expliquant les concepts de base pour faciliter la compréhension des travaux. Le plan de lecture sera le suivant :

– Cadre d'étude : quelle partie de la chaîne de fabrication a été considérée pour ce travail ?

- Définition et contexte d'utilisation de la qualification.
- Présentation de différents modèles de flexibilité.
- Extensions des modèles de flexibilité.
- Tests effectués dans le contexte de l'ordonnancement
- Description de l'implémentation du logiciel
- Et, pour finir, la conclusion et les perspectives

Plus on a de flexibilité, plus l'utilisation de la qualification se justifie. Par conséquent, les modèles ont été développés afin de savoir dans quel cadre attribué de la flexibilité (de capacité) à un atelier.

Quatre mesures de flexibilité ainsi que des variantes de celles-ci ont été obtenues. Afin de déterminer les meilleures variantes de ces mesures de flexibilités des études et des comparaisons ont été menées. Un équilibre optimal de la charge de travail doit être préalablement calculé pour l'utilisation de deux de ces mesures de flexibilité. Pour cela deux méthodes d'équilibrages ont été développées.

Des extensions de ces modèles de flexibilité ont été développées. Dans un premier temps, il s'agit de déterminer comment avec k qualifications (parmi N possibles) optimiser la flexibilité et ceci, avec un temps de calcul raisonnable. Ainsi, pour parvenir à des solutions rapides, robustes et efficaces, des heuristiques ont donc été mises en place et testées. Les performances de ces heuristiques ont été comparées les unes avec les autres. De plus, la prise en compte des spécificités des mesures liées à la flexibilité a été étudiée afin de réduire les temps de calcul. Sous certaines conditions, il est possible de voir les qualifications qui augmentent la flexibilité la plus.

Dans un second temps, une application dynamique de la flexibilité des modèles, en considérant l'évolution de l'en-cours (WIP en anglais) au cours de plusieurs jours, a été mise en place. D'autres facteurs, tels que : le degré de facilité de qualification de certaines recettes par rapport à d'autres, l'appartenance à un groupe de recettes et l'intégration des statistiques sur la disponibilité des recettes sur les équipements ont été pris en compte.

Afin de tester l'impact de la qualification flexible sur les performances de la production, des essais ont été menés avec des logiciels de simulation. Deux logiciels d'ordonnancement seront présentés : un pour l'atelier de photolithographie et l'autre pour l'atelier de gravure. Sur la base des résultats des tests effectués sur ces logiciels, il m'a été possible de démontrer que la qualification des mesures de flexibilité peut à la fois réduire les temps de production et apporter une meilleure robustesse des ateliers face aux pannes.

A la fin de la thèse deux types de logiciels seront présentés. Un premier qui propose les qualifications fondées sur les modèles de flexibilité puis un logiciel de simulation pour l'atelier de gravure.

Enfin, les conclusions sur ma thèse seront présentées ainsi que les conséquences à en tirer à partir des résultats de simulation obtenus. Les perspectives ainsi que les cadres d'utilisation de mes travaux seront alors discutées.

Contexte

Les circuits intégrés (*IC* en anglais) font partie intégrante de notre quotidien. Ils nous apparaissent sous diverses formes. Parmi les plus connues nous avons : les ordinateurs, les téléphones portables, les voitures mais aussi, des moins biens connues telles que les vêtements, les chaussures, les congélateurs, etc. En fonction de l'utilisation que l'on veut en faire, la diversité de ces circuits intégrés entraine une production de plus en plus spécialisée.

Fabrication de semi-conducteurs

Le processus de fabrication de semi-conducteurs peut se résumer en deux étapes (voir Figure 1) : le traitement front-end et le traitement back-end. En traitement *front-end*, on a :

- La fabrication des zones actives
- La fabrication des caissons
- Et enfin la fabrication des transistors

En effet, dans les fonderies ou *wafer fab* en anglais, à partir d'un lingot de silicium (ou d'un autre matériau semi-conducteur) des plaquettes minces (*wafer* en anglais) sont tranchées. Ce sera le substrat utilisé pour la fabrication des différentes zones citées ci-dessus. Une plaquette avec un diamètre de 300 mm peut contenir entre 100-1000 circuits intégrés En traitement *back-end* on a :

- La mise en place des interconnexions
- Le packaging

Les *IC* sont découpés. Leur fonctionnalité est testée puis, ils sont emballés. La thèse se focalise essentiellement sur le traitement niveau *front-end*.

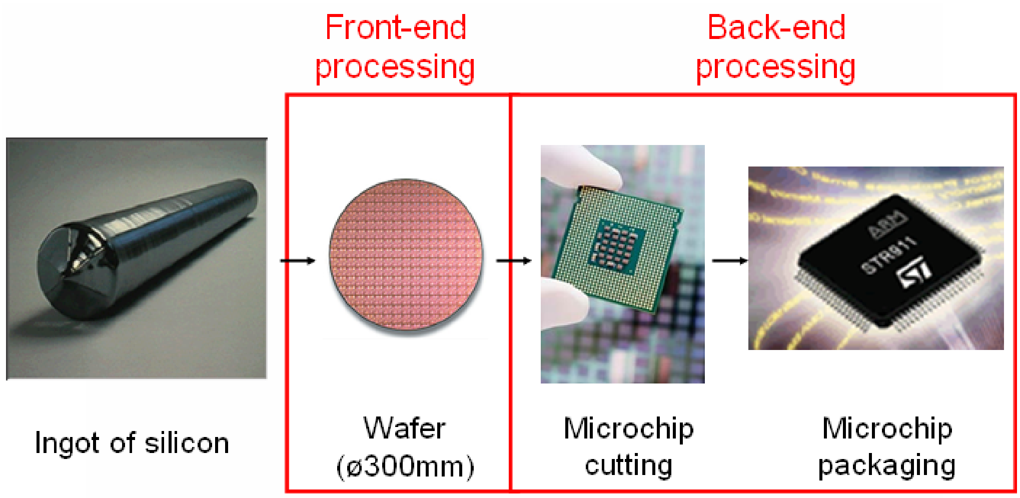


Fig. 1 – Fabrication de semi-conducteurs

La demande croissante en *IC* dirige le rythme et la quantité de production de l'industrie du semi conducteurs : produire plus en un temps plus court. Ceci explique l'augmentation des tailles des plaquettes des nouvelles générations d'usines. En effet, la taille la plus fréquente des plaques actuelles est de 200mm contre 300mm pour les usines les plus modernes. Les plaquettes sont transportées par lot de 25 dans des boîtes hermétiques et sécurisées appelées FOUP (*Front Opening Unified Pod*) (voir Figure 2). Le poids d'un lot de 25 plaquettes de 300mm atteint 8 kg. Si en plus du poids on ajoute le risque d'accidents et le risque de contamination, malgré les mesures drastiques de propreté, c'est naturellement vers un système de transport automatisé appelé AMHS pour *Automated Material Handling Systems* que la nouvelle génération d'usine se tourne.

FIG. 2 – Un lot pour 25 plaquettes

Le traitement *front-end* est effectué en plusieurs étapes : déposition, génération de masque, élimination et modification des couches sur la plaquette. Pour une meilleure compréhension de la création d'une couche, la figure ci-dessous nous en propose un modèle simplifié (voir Figure 3).

- L'étape de **déposition** correspond à l'ajout d'une couche sur la plaquette
- Celle de **Génération de masque** utilise la technologie de photolithographie pour ajouter des motifs sur les IC.
- La **gravure** élimine les zones générées par le masque (gravure sèche) ou supprime des éléments indésirables du matériau photosensible (gravure humide).
- **Modification des couches** de plaquettes : elle est effectuée par implantation ionique de manière à ce que les IC obtiennent les propriétés électroniques attendues.

Gestion des équipements et des recettes

Les machines de production dans les usines de semi-conducteurs sont très chères. En effet, une machine de l'atelier photolithographie peut couter jusqu'à plus de $20 millions. Par conséquent, il est nécessaire d'utiliser les équipements de la manière la plus efficace possible. Dans la suite de ce résumé, un outil pour améliorer l'utilisation des équipements sera proposé.

La recette va permettre de connaître comment on doit effectuer le process sur un équipement. Une recette contient des informations sur les caractéristiques de la

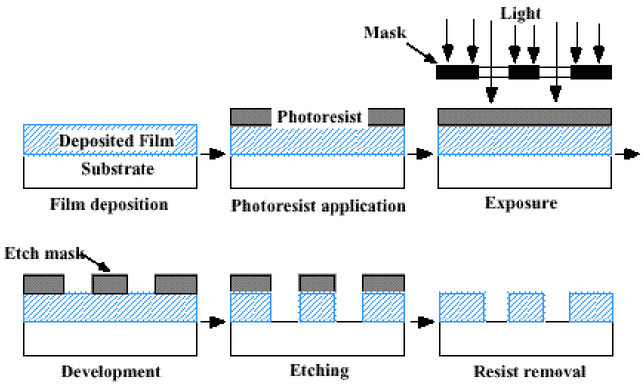


FIG. 3 – Traitement *front-end*.

machine telle que : la température à laquelle doit être processé le lot, la composition métallique et la pression de gaz nécessaire, par exemple. Avant qu'un produit ne passe sur une machine, il faut d'abord que sa recette soit clairement définie. C'est seulement après avoir identifié clairement la recette et l'avoir validé, que l'ingénieur process décide de la qualification de la recette sur la machine. Plus de détails sur la qualification sont donnés dans la prochaine section.

Si une recette pouvait être qualifiée sur toutes les machines d'un atelier l'opérateur aurait une flexibilité optimale pour traiter la recette. Toutefois, avant de permettre une qualification les ingénieurs process doivent effectuer plusieurs tests pour qualifier la recette sur les machines. Ceci prend du temps. Du fait de la volonté de rentabilité des équipements de l'usine, les managers aimeraient utiliser des équipements plutôt pour la production que pour effectuer des tests de qualifications.

De plus, les recettes doivent être qualifiées de telle sorte que leur affectation soit optimale. Dans cette thèse des modèles de flexibilité ont été développés dans ce but. Ils permettront de déterminer pour quel type de qualification la flexibilité est améliorée.

La section suivante traitera plus en profondeur de l'importance de gérer les qualifications et présentera les différents domaines d'étude des qualifications.

0.2 Gestion de Qualifications

Afin de réaliser un process sur un équipement, la recette de ce process doit être qualifiée sur l'équipement. Une recette d'un process définit toutes les caractéristiques qui devront être installées sur l'équipement afin de réaliser le process. Par exemple, la température du process, la composition métallique, la pression de gaz.

La qualification d'un équipement exige des essais, des ajustements et parfois, des installations. Ces mesures exigent l'implication de l'ingénieur process. C'est pourquoi une qualification peut prendre beaucoup de temps.

Ce n'est que lorsque la recette a été correctement qualifiée sur un équipement, qu'elle peut être traitée. Toutes ces étapes de qualification décrites plus haut entraînent indirectement un coût pour l'entreprise. Aussi, une gestion optimale de la qualification pourrait réduire ces coûts.

L'idée est qu'avec la gestion de qualifications on ait une meilleure répartition de la charge de travail d'un atelier. Pour ce faire, il a été pris en compte comme quantité d'en-cours le nombre de plaquettes. Cette idée se généralise très bien à d'autres types de produits pour la gestion des qualifications.

Pour terminer cette section, notons que très peu d'études ont été menées sur la gestion de qualifications dans la littérature. De plus, ces études se sont le plus souvent focalisées sur un domaine bien précis. A ma connaissance, aucun papier ne traite de manière générale et en profondeur la gestion de la qualification d'où la pertinence de la contribution de ces travaux de thèse.

Types de gestion de qualifications

Pour les composants de la production ayant besoin d'être qualifié, il est nécessaire de comprendre le lien entre machines et la gestion de qualification. Afin de parfaitement décrire ces liens, quatre différents domaines de la gestion de qualification peuvent être identifiés.

- **Efficacité des équipements.** Une bonne gestion des qualifications pourrait permettre une allocation optimale de la charge de travail et donc une meilleure utilisation des équipements.

- **Equipement en panne.** L'indisponibilité des équipements est un phénomène redondant (panne, maintenance préventive, ...). Sa prise en compte dans l'optimisation de la qualification des équipements devrait permettre de réduire l'impact de ces pannes.
- **Réduction du flux réentrant.** Une bonne qualification de l'équipement peut être utilisée pour réduire le degré de réentrance (DOR) de la ligne de production (c'est-à-dire le nombre de boucles imbriquées), ce qui pourrait réduire la complexité du système de production déjà non négligeable.
- **A.P.C.** (Advanced Process Control, APC). Amélioration de la qualité des plaquettes peut être réalisée via l'A.P.C en qualifiant les recettes de processus cruciaux sur les « meilleurs » équipements.

0.3 Modélisation de Flexibilité

Dans cette section, plusieurs modèles sont présentés afin d'évaluer la flexibilité d'un ensemble d'équipements devant traiter un ensemble de lots avec différentes recettes.

Le Tableau 1 ci-dessous, présente un exemple avec les caractéristiques suivantes :

- Trois (3) recettes
- Trois (3) équipements.

La lettre X signifie que la recette peut être effectuée sur cet équipement. La lecture du tableau est la suivante :

- En-cours recette 1 = 50 plaquettes, Equipements qualifiés : B et C.
- En-cours recette 2 = 400 plaquettes, Equipements qualifiés : A
- En-cours recette 3 = 450 plaquettes, Equipements qualifiés : B et C.

Qualifications : du fait de la contrainte de la recette 2, le meilleur équilibrage de charge de travail que l'on peut obtenir est donné sur la dernière ligne du tableau. Elle correspond à

- 400 plaquettes sur l'équipement A,
- 250 plaquettes respectivement sur B et C.

Il apparaît au vu de ces résultats que l'équipement A a une charge de travail

	Equipement			
Recette	A	B	C	Quantité de l'en-cours
1	-	X	X	50
2	X	-	-	400
3	-	X	X	450
Distribution	400	250	250	

TAB. 1 – Qualifications des recettes sur des équipements.

plus importante que B et C. Si une partie de la charge de travail sur l'équipement A pouvait être répartie sur les deux autres, la charge de travail pourrait être mieux équilibrée sur les équipements. Cela permettrait d'atteindre une plus grande flexibilité dans l'atelier. Pour évaluer la flexibilité d'un tel système des mesures de flexibilité doivent être formulées puis vérifiées.

0.3.1 Mesures de flexibilité

Très peu d'auteurs dans la littérature ont développé des modèles mathématiques pour mesurer la flexibilité. Parmi ceux-ci, il n'existe aucun modèle afin de mesurer la flexibilité de la répartition des capacités dans un atelier. Quatre mesures de flexibilité ont été développées dans cette thèse :

- *WIP flexibility* = flexibilité de l'en-cours (WIP)
- *Time flexibility* = flexibilité du temps
- *Toolset flexibility* = flexibilité des équipements
- *System flexibility* = flexibilité du système

Ces mesures permettent de calculer la flexibilité des ateliers en considérant le nombre de plaquettes et comment les distribuer sur les équipements qualifiés. L'idée est qu'un atelier est plus flexible si la charge de travail peut être équilibrée entre les équipements tout en tenant compte des pannes d'équipements.

Notations

Les paramètres suivants ont été utilisés pour la définition des mesures de flexibilité.

T	Nombre d'équipements dans l'atelier.
R	Nombre de recettes dans l'atelier.
NQT_r	Nombre d'équipements qualifiés pour la recette r.
γ	L'exposant de l'équilibrage ($\gamma > 1$).
WIP_r	Quantité de l'en-cours pour la recette r.

De plus les variables suivants ont été utilisés :

$WIP_{r,t}$	Quantité de l'en-cours de la recette r attribuée à l'équipement t.
$WIP(t)$	Quantité d'en-cours attribuée à l'équipement t.
$C(t)$	Temps de la production affectée à l'équipement t.

La quantité d'en-cours attribuée à l'équipement t est définiée comme la somme de la quantité de l'en-cours de la recette r attribuée à l'équipement t : $WIP(t) = \sum_{r=1}^{R}$

Les trois derniers paramètres ($WIP_{r,t}$, $WIP(t)$ et $C(t)$) sont déterminés par des variables qui sont les solutions de problème d'optimisation.

Toutes les mesures de flexibilité ont été normalisées pour avoir une valeur comprise entre 0 et 1. Pour une meilleure interprétation elles ont été ramenées à un pourcentage (entre 0% et 100%).

Flexibilité des équipements

La flexibilité peut être obtenue en ayant une capacité suffisante pour toutes les recettes. La mesure de Flexibilité des équipements (2.2) souligne l'importance d'avoir qualifié plus d'équipements pour les recettes avec le plus grand nombre de l'en-cours.

$$F^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} \frac{WIP_r}{NQT_r}} \tag{2.2}$$

Si tous les équipements étaient qualifiés pour toutes les recettes, la Flexibilité des équipements serait maximale.

Flexibilité de l'en-cours

L'idée : meilleure est la répartition de la quantité de l'en-cours sur les équipements, plus grande sera la flexibilité de l'opérateur. La mesure de la flexibilité de l'en-cours (2.4) évalue comment les quantités d'en-cours peuvent être équilibrées sur un ensemble d'équipements.

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} \sum_{r=1}^{R} WIP_{r,t}/T\right)^{\gamma}}{\sum_{t=1}^{T} \left(\sum_{r=1}^{R} WIP_{r,t}\right)^{\gamma}} \tag{2.4}$$

En augmentant le paramètre gamma, la valeur de la mesure de la flexibilité de l'en-cours est modifiée. Toutefois, les quantités de l'en-cours mieux équilibrées mesurent une plus grande flexibilité de l'en-cours que dans le cas où les quantités d'en-cours sont moins bien équilibrées.

Afin d'utiliser la mesure de flexibilité de l'en-cours, l'équilibre optimal des quantités de l'en-cours sur les équipements doit être déterminé. Pour ce faire, il est nécessaire de résoudre un problème d'optimisation.

Flexibilité du temps

Les recettes peuvent exiger différents temps de traitement sur les différents équipements. Au lieu d'examiner les quantités de l'en-cours, un point de vu serait d'examiner les délais de production nécessaires sur les équipements. Pour cela une mesure de flexibilité du temps a été développée (2.8).

$$F^{time} = \frac{C_{ideal}}{\sum_{t=1}^{T} \left(\frac{\sum_{r=1}^{R} WIP_{r,t}}{TP_{r,t}}\right)^{\gamma}} \tag{2.8}$$

Contrairement à la mesure de flexibilité de l'en-cours, la mesure de flexibilité du temps (2.8) dépend à la fois de la manière dont le temps de production est équilibré sur les équipements, et comment le temps total de la production est réduit au minimum. Grâce à l'exposant de l'équilibrage, gamma, il est possible de choisir qui de la minimisation ou de l'équilibre devrait être le plus important.

Flexibilité du système

Pour la flexibilité du système une mesure (2.10) a été mise au point. Pour cela les trois mesures sont prises en compte et il est également possible d'ajuster la valeur d'un paramètre (a, b ou c) afin de souligner l'importance pour l'une des flexibilités.

$$F^{SYS} = a \cdot F^{TS} + b \cdot F^{WIP} + c \cdot F^{time} \tag{2.10}$$

Dans Expression (2.10), les paramètres - tous entre 0 et 1 - sont définis de telle sorte que $a + b + c = 1$. Cela permet de s'assurer que la mesure alternative de flexibilité du système n'obtient que des valeurs comprises entre 0 et 1.

Examples

Les exemples dans les tableaux 2, 3 et 4 montrent comment les mesures de flexibilité peuvent être utilisées pour optimiser le choix des qualifications. Dans le tableau 2, trois équipements (A, B, C) et trois recettes (1, 2, 3) sont montrés. La recette 1 a un en-cours de 10 plaquettes et elle est qualifiée sur l'équipement A. La recette 2 a un en-cours de 30 plaquettes et elle est qualifiée sur l'équipement B. Enfin, la recette 3 a un en-cours de 40 plaquettes et elle est qualifiée sur l'équipement C. Si l'en-cours est distribué sur les équipements, l'équipement A doit fabriquer 10 plaquettes, l'équipement B 30 plaquettes et l'équipement C 40 plaquettes.

	Equipements			
Recettes	A	B	C	En-cours
1	X	-	-	10
2	-	X	-	30
3	-	-	X	40
Distribution	10	30	40	

TAB. 2 – Exemple de recettes qualifiées dans un atelier.

Supposons que dans cet atelier deux qualifications de plus sont possibles : la recette 2 sur l'équipement A comme dans le tableau 3 ou la recette 3 sur l'équipement B comme dans le tableau 4.

	Equipements			
Recettes	A	B	C	En-cours
1	X	-	-	10
2	**X**	X	-	30
3	-	-	X	40
Distribution	20	20	40	

TAB. 3 – Exemple avec la recette 2 qualifiée sur l'équipement A.

	Equipements			
Recettes	A	B	C	En-cours
1	X	-	-	10
2	-	X	-	30
3	-	**X**	X	40
Distribution	10	35	35	

TAB. 4 – Exemple avec la recette 3 qualifiée sur l'équipement B.

Pour calculer F^{time}, il faut connaître les temps de process qui sont donnés dans le tableau 5).

Les valeurs des mesures de flexibilité pour les qualifications possible sont montrées dans le tableau 6. Pour l'exemple, les paramètres suivants ont été utilisés : $\gamma = 4$, $a = 0.4$, $b = 0.3$ and $c = 0.3$.

Si on ne voulait améliorer que F^{TS}, il faudrait qualifier la recette 3 sur l'équipement B ($OQ_{3,B} = 1$). La recette avec le plus grand en-cours devient plus flexible et robuste si un équipement tombe en panne. Par contre, si on voulait améliorer F^{WIP} ou F^{time} ou F^{SYS}, il faudrait qualifier la recette 2 sur l'équipement A ($OQ_{2,A} = 1$) ;

	Equipements		
Recettes	A	B	C
1	100	100	125
2	50	75	100
3	75	100	125

TAB. 5 – Temps de process en nombre de plaquettes par heure

$OQ_{2,A}$	0	1	0
$OQ_{3,B}$	0	0	1
F^{TS}	0.33	0.38	0.54
F^{WIP}	0.45	0.53	0.50
F^{time}	0.52	0.84	0.52
F^{SYS}	0.42	0.56	0.52

TAB. 6 – Mesures de flexibilité pour les exemples

l'en-cours serait mieux équilibré sur les équipements.

0.3.2 Optimisation de l'équilibrage du travail

Afin d'utiliser deux des mesures de flexibilité, l'équilibre optimal de travail doit être trouvé. Pour cela deux méthodes d'équilibrage de travail ont été développées : une qui est optimale pour la mesure de flexibilité de l'en-cours et une autre qui est optimale pour la mesure de flexibilité du temps.

Nous présenterons dans un premier temps la méthode pour la flexibilité de l'en-cours. La méthode d'équilibrage pour la mesure de flexibilité de l'en-cours est effectuée pour une recette à la fois. Les quantités de l'en-cours de toutes les recettes ont déjà été distribuées sur les équipements de manière valide mais non-optimale donc, la quantité de l'en-cours d'une recette peut ensuite être redistribuée. La quantité de l'en-cours de la recette sera d'abord répartie sur l'équipement qui a actuellement la plus faible quantité de l'en-cours et ceci se poursuivra jusqu'à ce que la quantité de l'en-cours de l'équipement soit égale à la quantité de l'en-cours de un ou plusieurs autres équipements. Si, toutefois, il y a suffisamment de plaquettes, les quantités de l'en-cours sont distribuées également sur tous les équipements qui ont la quantité de l'en-cours la plus faible. Lorsque la quantité de l'en-cours d'une recette a été redistribuée sur les équipements, la méthode se poursuit avec la distribution de la quantité de l'en-cours de la prochaine recette. Cela continue jusqu'à ce qu'il n'y ait plus de redistribution de quantités de l'en-cours. Cela peut conduire à une amélioration de l'équilibre et il a été prouvé que cette méthode équilibre de manière optimale les quantités de l'en-cours pour la mesure de flexibilité de l'en-cours.

En outre, des moyens optimaux de distribution des travaux pour la mesure de flexibilité du temps ont été étudiés. Auparavant, une approche en deux étapes à été mise au point. Elle consiste dans un premier temps à minimiser le temps maximal de production des équipements. Puis dans un second temps à minimiser la somme du temps de production sur l'ensemble des équipements. Il a toutefois, été montré que cette approche ne pouvait pas être utilisée pour la mesure de flexibilité du temps. Pourquoi ? Parce qu'il existe des cas « pathologiques » où on ne peut pas conclure par cette approche comme le montre le tableau 7.

	Equipement				
Attelier	A	B	C	Somme	Temps max
1	5	5	0	10	5
2	6	2	2	10	6

TAB. 7 – Deux ateliers avec différents distribution de temps de production.

En effet, dans le tableau 7, le premier critère nous donne comme meilleur temps minimisé la valeur 5. Tandis que le deuxième critère ne permet pas de différencier les deux solutions, vu que les sommes minimisées qui conduisent à la même valeur sont les mêmes. Le critère 1 sera donc le critère qui permettra de départager les deux solutions, donc de privilégier l'atelier 1. Pourtant l'atelier 2 a plus de flexibilités.

De même, on peut avoir le cas où le premier critère ne permet pas de différencier les deux solutions (voir le tableu 8). Le second critère proposera donc une solution pour départager les deux solutions mais elle sera également fausse.

	Equipement				
Attelier	A	B	C	Somme	Temps max
1	6	6	0	12	6
2	6	4	4	14	6

TAB. 8 – Deux ateliers avec différents distribution de temps de production.

Pour terminer on peut avoir le cas où aucun des critères ne permet de prendre une décision. Dans ce cas, l'approche considérera deux solutions pourtant distinctes comme identiques. Ce cas est présenté dans le tableau 9.

	Equipement			Somme	Temps max
Attelier	A	B	C	Somme	Temps max
1	6	3	3	12	6
2	6	4	2	12	6

Tab. 9 – Deux ateliers avec différents distribution de temps de production.

Conclusion

Une autre méthode d'évaluation doit être trouvée. La méthode d'*active set* : Elle consiste à redistribuer la quantité de l'en-cours d'une recette (1 à la fois). Certains des équipements sont exposés à des contraintes ne leur permettant pas d'avoir une partie de l'en-cours de la recette actuelle. Ces contraintes sont appelées active set. Pour les autres équipements la quantité de l'en-cours de la recette est distribuée selon la méthode réduite de Newton avec la condition de Wolfe. Après chaque calcul, il est vérifié si la distribution est optimale ou si l'un des équipements d'active set doit être inclu ou si un équipement doit être exclu.

0.3.3 Tests Numériques

Des tests numériques ont été effectués pour évaluer la mesure de flexibilité. Les résultats de ces tests ont permis de distinguer deux cas possibles.

- Cas 1 : une recette est qualifiée et la flexibilité d'en-cours ou du temps augmente.
  Conséquence : l'augmentation de la flexibilité entraîne la diminution du temps de production
- Cas 2 : une recette est qualifiée et la flexibilité d'en-cours ou du temps reste la même.
  Conséquence : aucun changement sur le temps de production.

Conclusion : Le temps de production est lié à la flexibilité.

Cela s'explique par une meilleure distribution du travail sur les équipements. En outre, il a été remarqué que γ est un paramètre (de réglage) permettant de déterminer la « bonne » valeur du temps de production. Pour les petites valeurs de γ, le temps de production total de la solution est minimisé. Tandis que, les grandes valeurs de γ conduisent à un bon équilibrage du temps de production sur les équipements.

0.4 Optimisation des qualifications

Plusieurs aspects doivent être pris en compte pour la qualification de recettes. Cette section présentera les extensions des modèles de flexibilité étudiés.

Qualifications multiples

But : Augmenter la flexibilité.
Méthodes : Trouver la combinaison optimale de plusieurs qualifications.
Problème : complexité élevée, soit environ N^k combinaisons à vérifier, combinaison de k éléments parmi N soit $N \times (N-1) \times ... \times (N-k+1) = A_N^k$.

Considérant qu'il faut environ 2ms pour calculer la flexibilité du temps (sur un ordinateur « normal » sous VBA pour Excel : logiciel standard à l'entreprise participante), pour trouver les meilleures combinaisons de cinq qualifications parmi 100 possibles, il faudrait environ 219 jours pour calculer la solution optimale.

Heureusement, l'ordre des équipements n'a pas d'importance : la combinaison AB égale la combinaison BA, ce qui permet une réduction de la complexité du problème à un problème du binôme de Newton (C_N^k). Ainsi, on peut réduire le temps de calcul précédent à deux jours.

Néanmoins, malgré cette réduction, le nombre de combinaisons possibles reste toujours trop élevé. Pour palier à ce problème quatre heuristiques rapides ont été introduites. Une heuristique est une méthode qui ne trouve pas nécessairement la meilleure des solutions, mais peut trouver une bonne solution en un temps relativement court. Voici une brève description de ces heuristiques :

0.4.0.1 Heuristique « glouton »

On appelle algorithme glouton un algorithme qui suit le principe de faire, étape par étape, un choix optimum local, dans l'espoir d'obtenir un résultat optimum global. Dans le cas qui nous intéresse, on choisit une qualification à la fois, puis détermine celle qui augmente la mesure de flexibilité à chaque étape.
Temps de calcul : inférieur à une seconde.

0.4.0.2 Heuristique recherche locale 1

L'heuristique recherche locale 1 combine l'heuristique « glouton » avec un algorithme de recherche locale. A chaque étape, l'heuristique vérifie que la solution précédente (n-1) est la meilleure avec la nouvelle solution trouvée et elle fait l'inversion si nécessaire.
Temps de calcul : entre 6 et 20 secondes

0.4.0.3 Heuristique recherche locale 2

Une alternative plus rapide à l'heuristique recherche locale 1 a été développée : l'heuristique recherche locale 2. Elle laisse l'heuristique « glouton » trouver le nombre de qualifications souhaité. Par la suite la recherche locale 2, teste toutes les possibilités d'avoir la meilleure flexibilité sur la solution trouvé. Elle opère comme l'heuristique recherche locale 1 mais sur toutes les solutions trouvées.
Temps de calcul : entre 2 et 20 secondes

0.4.0.4 Heuristique recherche taboue

L'idée de la recherche taboue consiste, à partir d'une position donnée, à en explorer le voisinage et à choisir la position dans ce voisinage qui minimise la fonction objectif. Le risque cependant est qu'à l'étape suivante, on retombe dans le minimum local auquel on vient d'échapper. C'est pourquoi il faut que l'heuristique ait de la mémoire. Le mécanisme consiste à interdire (d'où le nom de tabou) de revenir sur les dernières positions explorées. Le temps de calcul dépend de la mise en œuvre du problème, mais est généralement plus long que pour les heuristiques de recherche locale.

0.4.0.5 Conclusions

Il a été prouvé (voir la thèse) que l'heuristique « glouton » trouve les qualifications optimales pour la mesure de flexibilité des équipements mais pas pour les autres mesures de flexibilité. De manière générale aucune des heuristiques ne trouve de solution optimale pour le calcul de la flexibilité de l'en-cours ou du temps.

Heuristique	Temps de calcul	Proximité de la solution optimale	Fréquence de succès
Glouton	rapide	parfois	parfois
Recherche locale 1 et 2	moins rapide que le glouton	meilleur que le glouton	souvent
Tabou	long	très proche	presque toujours

0.5 Extensions

0.5.1 Quantités dynamiques de l'en-cours

Pour améliorer l'efficacité de la qualification, le futur est un paramètre à prendre en compte. Dans cette thèse une approche dynamique a été développée. Une étude préalable a été effectuée sur les paramètres pouvant influencer la quantité d'en-cours sur plusieurs périodes. Une période peut être définie par exemple comme 24 h.

Il résulte des études faites par la gestion de production que la manière d'évaluer la quantité d'en-cours sur plusieurs périodes T_d est différente. Face au besoin de différencier la qualité de l'estimation de la quantité de l'en-cours sur différentes périodes un poids d a été introduit. Pour la première période T_1, on connaît avec précision la quantité de l'en-cours. Pour les périodes qui suivent, cette quantité devient plus incertaine.

Deux formules d'anticipation des quantités dynamiques d'en-cours pour les mesures de flexibilité ont été développées. Pour la première formulation (4.1), le poids de chaque période est un coefficient multiplicatif de la quantité d'en-cours de chaque période. Ce produit est ensuite additionné et considéré pour le calcul de la mesure de flexibilité du système F^{SYS}.

$$
\begin{array}{ll}
\max F_1^{SYS} & \\
WIP_r = \sum_{p=1}^{P} WIP_{r,p} d_p & \forall r \\
\sum_{t=1:Q_{r,t}=1}^{T} WIP_{r,t,p} = WIP_{r,p} & \forall r = 1,..,R \quad p = 1,..,P \\
\sum_{r=1}^{R} \sum_{t=1}^{T} Q_{r,t} = k_0 + k & \\
Q_{r,t} \in \{0,1\} & r = 1,..,R \quad t = 1,..T \\
WIP_{r,p} \geq 0 & \forall r = 1,..,R \quad p = 1,..,P
\end{array}
\tag{4.1}
$$

Dans la seconde formulation, 4.2, les poids périodiques sont multipliés avec chacune des mesures de flexibilité.

$$
\begin{array}{ll}
F_2^{SYS} = \max \sum_{p=1}^{P} F_p d_p & \\
\sum_{t=1:Q_{r,t}=1}^{T} WIP_{r,t,p} = WIP_{r,p} & \forall r = 1,..,R \quad p = 1,..,P \\
\sum_{r=1}^{R} \sum_{t=1}^{T} Q_{r,t} = k_0 + k & \\
Q_{r,t} \in \{0,1\} & r = 1,..,R \quad t = 1,..T \\
WIP_{r,p} \geq 0 & \forall r = 1,..,R \quad p = 1,..,P
\end{array}
\tag{4.2}
$$

0.5.2 Extensions additionelles

0.5.2.1 Niveaux de facilité à qualifier

Qualifier une recette sur un équipement demande de la main d'œuvre et prendre du temps. C'est pourquoi certaines recettes sont considérées comme plus faciles à qualifier que d'autres. Pour mieux formaliser ce constat, plusieurs niveaux de qualification ont été proposés, par exemple :

- Niveau 1 : facile à qualifier
- Niveau 2 : moyennement difficile à qualifier
- Niveau 3 : difficile à qualifier

Un certain nombre de qualifications sont autorisées pour chaque niveau de facilité. Par exemple trois qualifications faciles, une moyenne et une difficile.

0.5.2.2 Sous-groupes de recettes

Avec un groupe de recettes de même type on forme des sous-groupes. Les recettes à l'intérieur d'un sous-groupe ont certaines propriétés en commun et peuvent donc partager les mêmes tests de qualifications. Par conséquent, une fois qu'une recette a été qualifiée sur un équipement, le reste des recettes du même sous-groupe sera plus facile à qualifier sur le même équipement.

0.5.2.3 Disponibilité de recette sur l'équipement

Dans l'usine, on dispose de statistiques sur le temps de disponibilité d'une recette pour les équipements. Un petit taux de disponibilité indique qu'il pourrait être difficile de traiter la recette sur cet équipement et, dans ce cas, il serait préférable de traiter la recette sur un autre équipement.

Par conséquent, les renseignements sur la disponibilité ont été intégrés dans la mesure de la flexibilité de l'équipement par l'introduction d'un paramètre alpha. Si par exemple $\alpha = 0,8$ (80% de disponibilité) cela signifie que la disponibilité est intégrée dans la mesure de flexibilité des équipements comme dans (4.4).

$$F_{\alpha}^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} \frac{WIP_r}{\sum_{t=1}^{T} \alpha_{r,t} \cdot Q_{r,t}}} \tag{4.4}$$

Conclusion : avec cette intégration de la disponibilité dans la mesure de flexibilité des équipements, il est maintenant possible de dire quelle recette et quel équipement permettront d'augmenter le plus la flexibilité, tandis que précédemment on ne pouvait dire que l'impact des recettes, pas des équipements. Il est donc préférable de qualifier une recette sur un équipement qui est souvent disponible.

0.6 Impact des qualifications sur l'ordonnancement

Dans les usines de semi-conducteurs l'ordonnancement est un facteur clé pour la performance de l'usine. Afin d'évaluer l'impact des mesures de flexibilité sur l'ordonnancement (et donc sur la performance de l'usine) des tests ont été effectués associant deux simulateurs d'ordonnancement :

- Le premier pour l'atelier de photolithographie
- Le second pour l'atelier de gravure sèche.

0.6.1 Le simulateur d'ordonnancement pour l'atelier de photolithographie

Le simulateur d'ordonnancement pour l'atelier de photolithographie fonctionne en deux étapes.

- Etape 1 : Ordonnancement des lots en fonction des priorités qui leur ont été affectées ainsi que de leur temps d'attente (règle globale).
- Etape 2 : Dispatching sur les équipements dans l'ordre obtenu précédemment (règles locales).

Les règles locales sont les suivantes :

1. **Disponibilité** : si l'équipement est disponible au cours de la période en cours
2. **Charge de l'équipement** : plus une machine est chargée moins les lots seront dispatchés sur la machine
3. **Emplacement du masque** : haute priorité si le masque est déjà sur l'équipement. Faible priorité s'il se trouve sur un autre équipement et priorité moyenne si le masque est ailleurs mais pas sur un autre équipement.
4. **Configuration du batch** : si le lot a la même capacité que le lot précédent un train de batch peut être construit, ce qui peut diminuer des configurations supplémentaires. Un train de batch représente des lots nécessitant la même recette et qui sont processés les uns après les autres.

Le simulateur d'ordonnancement considère des règles globales et locales et forme un planning.

0.6.2 Le simulateur d'ordonnancement pour l'atelier de gravure sèche

Le simulateur pour l'atelier de gravure sèche fonctionne à peu près de la même façon que le simulateur pour l'atelier de photolithographie. Il y a cependant quelques différences importantes dues au fait qu'un équipement de gravure sèche est un équipement de cluster pour le traitement parallèle. En outre un équipement de gravure sèche a plusieurs ports où les lots sont placés pour décharger et charger leurs plaquettes. De ce fait, le principe essentiel est qu'un lot sera expédié vers un équipement quand un port de charge de l'équipement est devenu disponible.

Le lot qui est envoyé dépend d'une combinaison des règles globales et locales. Au niveau global, les règles sont les mêmes que pour le simulateur de l'atelier de photolithographie. Comme l'équipement est déjà décidé, les règles locales établissent la combinaison de chambres qui seront utilisées pour le traitement. Seules les chambres qui sont qualifiées pour la recette du lot sont prises en compte. Les chambres qui seront effectivement mises en œuvre sont décidées par les règles locales :

- Maximiser l'usage des chambres désirées : la capacité des chambres qui termineront le traitement en premier (chambres désirées) est la même que pour le lot en cours.

- Maximiser l'usage des chambres qui ont la même capacité : des chambres qui ne termineront pas en premier, mais où la capacité est la même comme pour le lot en cours.
- Minimiser l'usage des chambres ayant des capacités différentes : le lot utilise une chambre qui n'est pas désirée, où la capacité est différente et où un autre lot avec la même capacité peut être traité.
- Utilisation maximale du nombre de chambres : le nombre maximum de chambres utilisées par le lot.
- Limite de capabilité - le même type de travail ne doit pas occuper trop des équipements dans l'atelier

0.6.2.1 Mesure de performance

Les mesures de performances qui ont été utilisées pour les tests sont les suivantes :

- DAO - Journées-en-opération : combien de jours en moyenne les lots restent dans l'atelier
- DAO - l'écart-type de la DAO
- 8ème décile de la DAO
- Charge maximal d'un équipement : la plus grande quantité de l'en-cours d'un équipement dans l'atelier

Comme la direction de l'usine veut éviter que les lots restent trop longtemps dans l'usine, il est important de réduire la moyenne de DAO des lots. L'écart-type de la DAO et le huitième décile de la DAO indiquent la variabilité des temps de cycle des lots. La charge maximale d'un équipement indique si les lots peuvent être bien répartis sur les équipements. Une valeur haute de la charge maximale d'un équipement indique que cela peut prendre beaucoup de temps jusqu'à ce que le dernier lot soit traité.

0.6.2.2 Impact de qualifications flexibles sur l'ordonnancement

Plusieurs tests ont été réalisés afin de voir comment les qualifications influencent sur la performance de la planification. Il a été démontré que les qualifications qui améliorent la flexibilité de temps et la flexibilité de l'en-cours améliorent le temps et la charge maximale des équipements dans un atelier. Quand les mesures de flexibilité de temps ou de flexibilité de l'en-cours approchent 100%, le temps de cycle et la charge maximale des équipements approchent leur optimum.

Les essais ont indiqué que les qualifications fondées sur la mesure de la flexibilité des équipements n'améliorent pas nécessairement la performance. Cela ne se produit généralement que lorsque les qualifications font croître de la même manière la flexibilité de l'en-cours et la flexibilité du temps. D'autre part, il a été montré que des qualifications basées sur la mesure de la flexibilité des équipements peuvent améliorer la robustesse de l'atelier. Les tests ont notamment montré que, lorsqu'un équipement devient indisponible dans l'atelier, la production peut se poursuivre sans de trop grandes perturbations quand des qualifications basées sur la mesure de flexibilité d'équipement ont été préalablement menées. Ce n'est en général pas le cas pour des qualifications basées sur les mesures de flexibilité de l'en-cours et de flexibilité du temps.

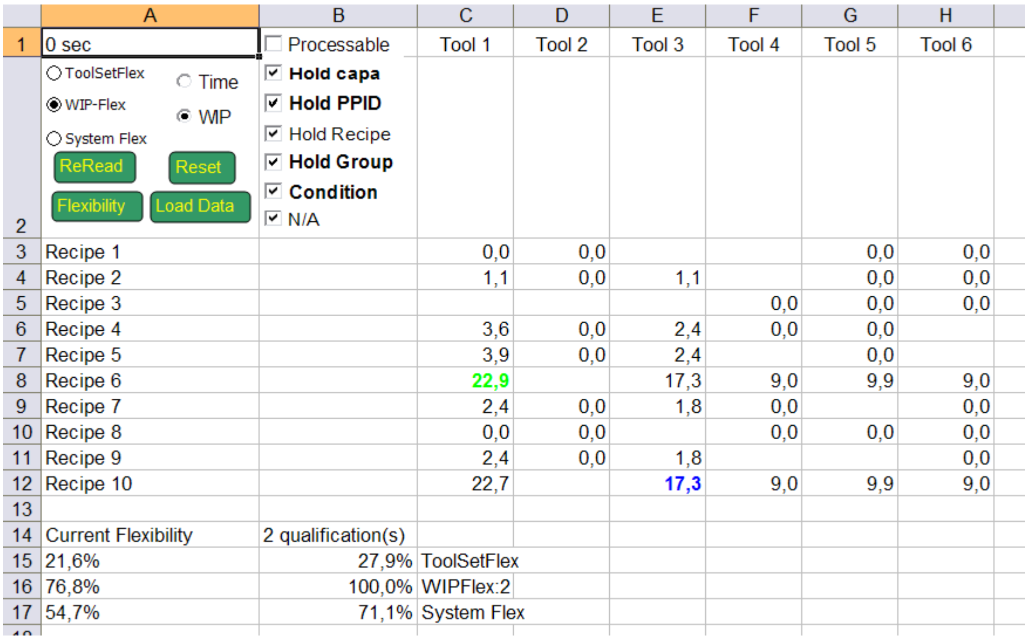
0.7 Implémentations

Au cours de la thèse, deux logiciels ont été programmés pour réaliser les essais et les simulations qui ont été effectués tout au long de la recherche. Les deux logiciels ont été programmés en VBA avec des résultats affichés en MS Excel qui sont les logiciels standards comme prototype de l'entreprise participante.

0.7.1 Le logiciel pour la gestion des qualifications

Le premier logiciel est basé sur la gestion des qualifications dans un atelier (voir Figure 4). Les recettes avec des quantités de l'en-cours sont affichées et il est indiqué sur quels équipements les recettes sont qualifiées. À titre de contribution, le logiciel utilise également le temps de processus des recettes sur les équipements.

Pour les qualifications actuelles les mesures de flexibilité sont calculées. Afin de mesurer les valeurs de la flexibilité de l'en-cours et la flexibilité du temps, le logiciel calcule le meilleur équilibre de la charge de travail pour ces mesures.



	A	B	C	D	E	F	G	H
1	0 sec	Processable	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6
2	ToolSetFlex, Time, WIP-Flex, WP, System Flex, ReRead, Reset, Flexibility, Load Data	Hold capa, Hold PPID, Hold Recipe, Hold Group, Condition, N/A						
3	Recipe 1		0,0	0,0			0,0	0,0
4	Recipe 2		1,1	0,0	1,1		0,0	0,0
5	Recipe 3					0,0	0,0	0,0
6	Recipe 4		3,6	0,0	2,4	0,0	0,0	
7	Recipe 5		3,9	0,0	2,4		0,0	
8	Recipe 6		22,9		17,3	9,0	9,9	9,0
9	Recipe 7		2,4	0,0	1,8	0,0		0,0
10	Recipe 8		0,0	0,0		0,0	0,0	0,0
11	Recipe 9		2,4	0,0	1,8			0,0
12	Recipe 10		22,7		17,3	9,0	9,9	9,0
13								
14	Current Flexibility	2 qualification(s)						
15	21,6%	27,9%	ToolSetFlex					
16	76,8%	100,0%	WIPFlex:2					
17	54,7%	71,1%	System Flex					

FIG. 4 – Le logiciel pour la gestion des qualifications

En outre, le logiciel montre à quel point la flexibilité peut augmenter si une qualification d'une recette est effectuée sur les équipements pour l'ensemble de ces qualifications. La qualification qui peut augmenter le plus la mesure de flexibilité est marquée en vert.

De plus, il est possible de choisir le nombre de qualifications qui doivent être proposées. Le logiciel calcule avec une des heuristiques, des qualifications qui devraient être choisies en vue d'accroître au maximum la mesure de la flexibilité. Ces qualifications sont marquées en bleu. Si l'une était auparavant verte, elle est conservée en vert. Les valeurs des mesures de flexibilité pour faire connaître ces qualifications sont montrées à l'utilisateur.

Il est également possible de définir si les qualifications sont faciles ou difficiles à mener, et le nombre de chaque type pouvait être accompli. Par défaut, toutes les recettes sont faciles à qualifier. Aussi les groupes qui appartiennent à des recettes et le nombre de groupe qui devraient être qualifiés peuvent être définis par l'utilisateur. De même la disponibilité des recettes sur les équipements peut être remplie par l'utilisateur. Par défaut les recettes appartiennent à un seul groupe et la disponibilité des recettes est fixées à 100%.

0.7.2 Le simulateur d'ordonnancement pour l'atelier de gravure sèche

Le second logiciel est un simulateur d'ordonnancement pour l'atelier de gravure sèche (voir Figure 5). Le modèle a été développé depuis un simulateur d'ordonnancement pour l'atelier de photolithographie. Les règles du nouveau simulateur ont cependant été adaptées pour les spécificités des équipements de la gravure sèche. Le traitement parallèle sur les différentes chambres a été intégré. En outre, le fait que les équipements aient des ports pour charger des lots a été intégré. Une particularité qui a été négligée est ce que l'on appelle les écluses de charge (load locks). Il a été considéré que les plaquettes sont transportées directement des lots aux chambres. En outre l'ordonnancement de l'intérieur des équipements a été réalisé aussi simplement que possible : dès qu'une chambre est disponible, la prochaine plaquette est mise dessus. La contrainte qui vient de la charge de travail du transporteur dans l'équipement a également été ignorée.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1							3,00							
2	Load	Qualifications			Schedule									
3							Schedule			EL23S01	EL23S01	EL23S01	EL23S01	EL23S02
4	LotID	WIP	Prio	Capa	Recipe	DAO	Order	RcpNb	DAO out	PM1	PM2	PM3	PM4	PM2
17	Q821ZER	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	1,08	19	5	0,00	0	0	0	0	0
18	Q818CCC	25	9	ETCH_SI_GATE	BGATE-E-C0	1,69	12	4	1,80	0	0	0	0	0
19	Q819RFW	25	9	ETCH_SI_GATE	BGATE-E-C0	1,44	16	4	0,00	0	0	0	0	0
20	Q819ZWV	25	9	ETCH_SI_GATE	BGATE-E-C0	1,61	15	4	0,00	0	0	0	0	0
21	Q818AVE	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,13	7	6	0,35	1	1	0	0	1
22	Q821ADX	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,03	26	5	0,00	0	0	0	0	0
23	Q821BYY	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,06	23	5	0,00	0	0	0	0	0
24	Q821GVX	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,63	21	5	0,00	0	0	0	0	0
25	Q821KXV	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,02	27	5	0,00	0	0	0	0	0
26	Q820MXB	25	9	ETCH_SI_GATE	BGATE-E-C0	1,96	10	4	2,06	0	0	0	0	0
27	Q818TWW	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,11	17	6	0,00	1	1	0	0	1
28	Q819NJC	25	9	ETCH_SI_GATE	BGATE-E-C1	1,86	13	7	2,06	0	0	0	0	0
29	Q818DAX	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,13	11	6	0,32	1	1	0	0	1
30	Q819VKX	24	9	ETCH_SI_GATE	BGATE-E-C0	2,13	8	4	2,20	0	0	0	0	0
31	Q821NMP	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,61	22	5	0,00	0	0	0	0	0
32	Q821RCR	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,01	30	5	0,00	0	0	0	0	0
33	Q817KSM	25	9	ETCH_NIT_SIPROT	BSIPR-E-C12	0,86	1	8	0.92	1	1	0	0	1

FIG. 5 – Le simulateur d'ordonnancement pour l'atelier de gravure sèche

Le simulateur fonctionne de manière à ce que, dès qu'un port devient disponible, le lot considéré comme le meilleur selon les règles globales et locales soit dispatché sur l'équipement et dans ses chambres. En réalité, dans certains cas, il serait préférable de ne pas dispatcher des lots qui sont actuellement dans l'atelier sur l'équipement libre. Cela pourrait par exemple se produire lorsqu'il n'y a que quelques lots restants dans l'atelier et sachant qu'un lot de haute priorité va bientôt arriver à l'atelier. Ces cas ont été ignorés.

En dépit des petites simplifications du logiciel les résultats des simulations semblent bien fonctionner. Des ajustements des paramètres de contrôle ont été réalisés en vue d'accroître les performances du simulateur. Puisqu'il y a plusieurs objectifs, tous ne peuvent pas être optimisés. Toutefois, les paramètres ont été ajustés de sorte que des performances assez proches des optimas sont obtenues pour tous les objectifs.

0.8 Discussion et Perspectives

La gestion des qualifications est rarement prise en compte dans la littérature scientifique, et est généralement examinée en termes généraux sans définir les orientations possibles pour de nouveaux travaux de recherche. C'est pourquoi, dans cette thèse, les enjeux de la gestion des qualifications ont été caractérisés, et différentes pistes de recherche ont été présentées.

Afin de trouver quelles recettes doivent être qualifiées, quatre mesures de flexibilité ont été développées :
- F^{TS} = flexibilité des équipements,
- F^{WIP} = flexibilité de l'en-cours,
- F^{time} = flexibilité du temps,
- F^{SYS} = flexibilité du système.

A l'aide de ces mesures, il est possible de montrer pour quelle recette et sur quel équipment faire une qualification. Il a été montré que les mesures sont efficaces et que les qualifications qui augmentent la flexibilité de l'en-cours ou la flexibilité du temps peuvent améliorer les performances d'un atelier. Les qualifications qui augmentent la flexibilité des équipements peuvent améliorer la robustesse de l'atelier si un équipement devient indisponible.

Quatre heuristiques ont été proposées et mises en œuvre pour déterminer des propositions de qualifications en un temps raisonnable : une heuristique gloutonne, deux heuristiques de recherche locale et une métaheuristique de type recherche taboue.

L'heuristique gloutonne trouve très rapidement une solution. Les deux heuristiques de recherche locale sont bien évidemment plus lentes que l'heuristique gloutonne mais elles trouvent plus souvent la solution optimale. La recherche taboue trouve très souvent la solution optimale mais elle est bien plus lente que les autres méthodes. Parce que les heuristiques ont des performances différentes, le choix de l'une ou de l'autre dépend du temps que l'utilisateur peut attendre avant de prendre une décision, et par conséquent souvent du niveau de décision.

Différentes extensions ont été mises en œuvre afin d'intégrer certaines contraintes industrielles, comme l'anticipation de la dynamique de l'évolution des en-cours, la facilité à réaliser des qualifications ou le groupage des recettes. Les utilisateurs doivent décider quelles sont les extensions les plus pertinentes suivant le contexte.

Impact des qualifications sur l'ordonnancement

Trois outils d'ordonnancement ont été utilisés afin de voir quels sont les impacts d'une gestion optimisée des qualifications sur la planification dans les ateliers de photolithographie, de gravure sèche et de diffusion. Différents tests ont montré que les qualifications qui augmentent les mesures de flexibilité améliorent de manière significative les performances de l'ordonnancement dans une grande majorité de cas. Néanmoins, certains cas montrent que des paramètres additionnels doivent être considérés.

Perspectives

Il est possible de faire d'autres tests permettant d'évaluer quels sont les impacts des qualifications optimisant les mesures de flexibilité. Toutefois, nous avons déjà une idée assez précise de ces impacts, et il semble préférable de travailler sur les définitions de nouvelles extensions pour différents contextes.

Par exemple, une définition plus précise des différents niveaux de facilité de qualifications pourrait être faite. Le modèle existe déjà mais il faudrait étudier plus précisément comment définir les niveaux. Cette remarque est aussi vraie pour la définition des groupes de recettes.

General Introduction

Semiconductor manufacturing is a continuously developing industry. Many of the emerging issues in semiconductor manufacturing are new in industrial engineering and require new points of working. Semiconductor integrated circuits (IC) are processed on circular wafers, that undergo between 250 and 500 process steps [71]. Each process step is performed in a workshop with a set of tools. The tools have different characteristics which effect the quality and the throughput speed of the process. In order to undergo a process step, the process type (called recipe) needs to be qualified on the tool, before the wafer can be processed on the tool. Qualifications take time to setup and maintain and therefore semiconductor factory management cannot allow all process types to be qualified.

The number of process steps, the different characteristics of the tools and most of all process qualifications on tools makes semiconductor manufacturing an unique industry and interesting to study.

Additionally, semiconductor manufacturing could be divided in at least two types of facilities (also called fabs) :

- *Low mix/high volume fabs* where only a few products are performed in high volumes,
- *High mix/low volume fabs* where many different products are performed in low volumes.

In the former type, manufacturers try, in a more traditional manufacturing style, to separate production lines on different tools, such that production of the different products do not effect each other too much. If a tool breaks down or needs maintenance it will only impact one production line. In the second manufacturing type of fabs, manufacturers need to implement a completely different strategy. There are not enough tools to separate the production lines. Therefore process types need to be qualified such that manufacturing is flexible enough to be continued for all lines even if tools break down or are maintained.

In this thesis a novel strategy how qualifications can be used in order to increase flexibility are studied. At first, a qualification management approach for process qualifications on tools in a workshop is developed. The flexibility measures are integrated in the approach in order to suggest qualifications in workshops. The qualification approach also considers different constraints and it is studied how it can be optimized. Finally it will studied how the approach can impact the manufacturing performance by using tests with scheduling simulators for different workshop types.

In Chapter 1 the industrial and scientific context of the thesis will be presented. This will help to understand the need of flexibility measures which are developed in Chapter 2. In Chapter 3, the properties of the flexibility measures are studied and it is studied how the flexibility measures can be optimally implemented in the qualification approach. Extensions to the approach are introduced in Chapter 4. In order to test if the approach can improve the performances in wafer fabs, impact of the approach on scheduling is tested in Chapter 5. Conclusions and perspectives of the research are drawn in the last Chapter 6.

Chapitre 1

Industrial and Scientific Context

In order to understand the objectives that have driven industry to where it is today, the history of industrial engineering is briefly described. Industrial engineering has during the last century become an "ideology" how to make production cheaper, more effective and more flexible.

It will also be seen how industrial engineering works for semiconductor manufacturing. In semiconductor manufacturing two production strategies have evolved. In the first strategy, the objective is to manufacture high volumes for a small number of products. For this type of manufacturers, economies of scale and thus high output, is the most important driver. In the second strategy, low volumes for many different products are produced. These manufacturers need to have a production which is flexible and can rapidly be adapted to the diverse demands of their costumers. It is foremost for the second strategy that this thesis has been initiated.

To better understand how semiconductor manufacturing works, the production will be described, linking to the details which concern the models that are developed in the following chapters. It is explained what the recipe of a process is and why recipes need to be qualified on a tool before processes are performed.

As it will be seen, not so much work has been done within qualification management (QM). Earlier work does not consider the consequences of qualifications that are changed. For the work which does consider the potential of QM, the main part only considered how process control can be improved with new qualifications. In one case the aspect of reducing reentrancy in the production has been considered. It will, however, be argued that can only be valid for high volume manufacturing with a small number of different products. So far, no study has been performed which shows how qualifications can be used to improve the flexibility of production. What is meant with flexibility is defined in Chapter 2.

In this chapter, the concept of qualifications is researched. Since the area of qualification management has not been so well studied in earlier work, the areas of QM will be defined and the importance of QM will be stressed. Also different challenges for new research within QM need to be outlined. With that in mind, it is possible to go on to see how qualifications can be performed in order to increase flexibility in semiconductor manufacturing.

1.1 A Brief Introduction to Industrial Engineering

Industrial engineering was developed at the beginning of the industrial revolution in the end of the 19th century to consider implementations and improvements for industrial manufacturing systems. The first academic industrial engineering department was opened at PennState University in 1908.

For the past 100 years the science of industrial engineering has been strongly associated with the car industry. In the beginning of the last century, Henry Ford's automobile factories introduced the moving assembly line in their production [100]. The moving assembly line was not so much a mechanical revolution as it was a new concept for production process. The new concept revolutionized production management with much faster production times and, most importantly, an end product which was much cheaper than its ancestor. The Ford Model T was the only car most people could afford to buy, and Ford's competitors had to adjust their production in order to regain their market shares. The car manufacturers that did not adjust their production disappeared in only a few years.

However, Ford's production system was rigid or as Henry Ford himself stated in his autobiography [27] : "*Any customer can have a car painted any color that he wants as long as it is black.*"

Ford's competitor General Motors (GM) developed a more flexible production line where customers could choose different properties for one model. GM kept the moving conveyor belt in their factories but introduced customization production. Notably GM cars could be chosen in different colors.

In the 50s Japanese factory managers understood that their productivity lagged far behind the American, and they went overseas to learn from their American counterparts. Indeed they did not only learn from the Americans but also improved their production system. Slowly the concept of Kaizen production took form in Japan : a continuous improvement of each operation in the production process [47].

Europeans and Americans had made many revolutionary industrial inventions from the 19th century until the middle of the 20th century. Factories were built to construct the new inventions, but as the inventions became obsolete, factories had to close and factory workers became unemployed. Japanese managers understood that factories constantly need to be updated to produce new and better versions of their products. Even more importantly, everybody working in an industry constantly needs to consider how his or her work could be done better and more efficiently.

Toyota Motor Cooperation understood that inventory really is waste. First of all, it is a waste of space, since additional machines could be placed on the space where inventory was kept. Secondly, it was a waste of money since the money that was paid for the inventory could be used more wisely. Hence they implemented the Just-In-Time philosophy : Material needed in the production process should not be delivered before it was needed. The production should not be sent forward to the next workshop in the assembly line until it could be processed there. In the end the finished product should not be ready until there is a customer for the product

The Toyota production system is recognized as one of the best in industrial engineering. It is considered to be both cost effective and adaptive to new demands. However, a fire at one of Toyota's main factories in 1997 put the whole company at risk. Since Toyota had reduced their inventories they no longer had any products to deliver. Toyota managed to handle the crisis but the losses were severe.

In the spirit of the industrial improvements during the last century, a simple but efficient model for improving production will be developed in this thesis. The goal is to achieve a more flexible production system that can be used for all workshops in semiconductor manufacturing. However, as will be shown later in this thesis, increased flexibility does not only mean faster production at any price, but also, a more robust production system can be achieved to reduce the risk in production.

1.2 Semiconductor Manufacturing

Since the mid 20th century, product development within the semiconductor industry has exploded : from the first transistor [36] to the first microchips [98] and today's nanotechnology products. This development can be summarized by the Moore's law [76]. Co-founder of Intel corporation Gordon E. Moore, predicted in 1965 that the number of transistors that could be placed on an integrated circuit would double approximately every two years. Since this prediction has been more or less true, the prediction has become known as the Moore's law. The prediction is expected to be fulfilled until at least 2015 if not longer [56]. The diagram in Figure 1.1 shows the number of transistors on Intel processors and Moore's law (dotted line) from 1971 until 2004. It can, however, be questioned whether Moore's law really is an observation or prediction in its true sense or if it became a road map that the semiconductor manufacturers feel must be followed.

1.2.1 Industrial engineering in wafer fabs

Some semiconductor manufacturers have continued with a production process similar to traditional manufacturing; trying to minimize costs by only producing a few products with as high volumes as possible in so called low mix/high volume fabs. However, the demand for semiconductor products are diversified, which made some semiconductor companies develop a high mix of several products which are only produced in small volumes. The latter strategy requires new tactics and new kinds of solutions for industrial engineering in semiconductor fabs. For these kinds of high mix/low volume manufacturers [65], tools no longer belong to only one production line. Since there are so many production lines and not enough space or economic resources to have independent tools for each line, a tool must handle many different process types. For more about the differences between different semiconductor manufacturers a benchmark has been evaluated by [63].

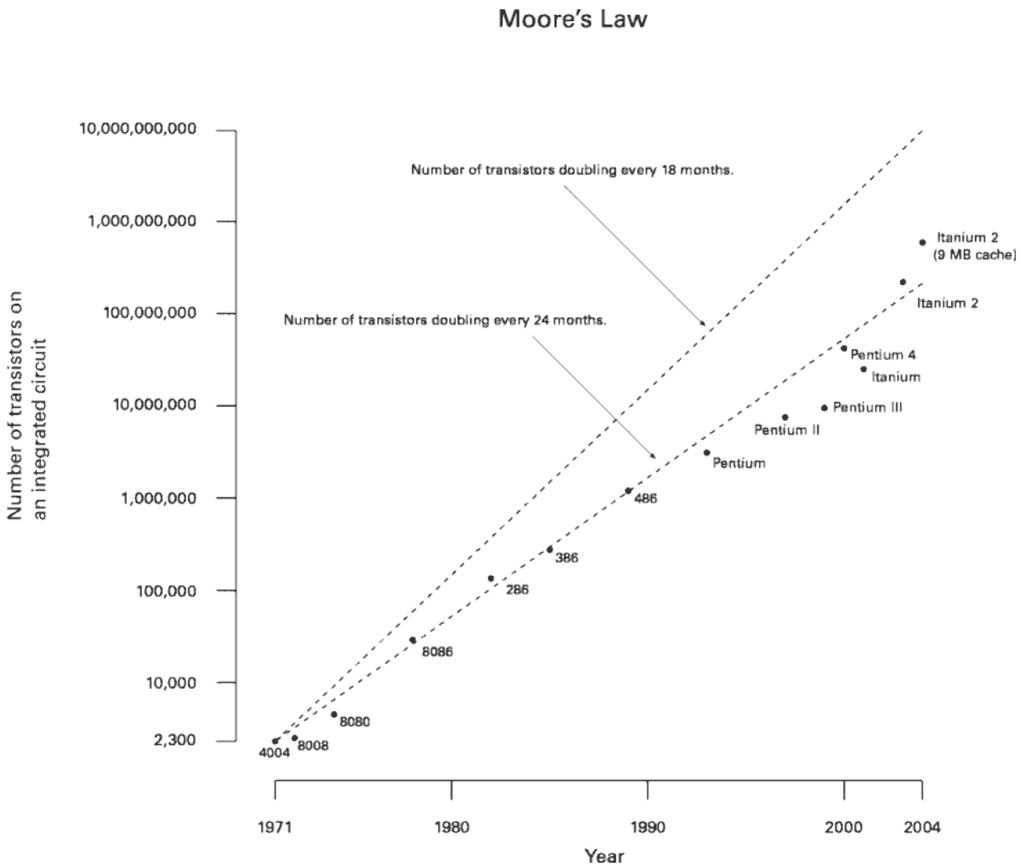


FIG. 1.1 – Moore's law

There is a big difference between low mix/high volume fabs (e.g. Intel, AMD and Samsung) and high mix/low volume fabs (e.g. STMicroelectronics and Infineon). In the former, production remains rather constant and the fab management may process products of one type on the same tools without bigger adjustments. Keeping separate production lines on different tools allows influences of unavailable tools to only affect one production line.

In high mix/low volume fabs, the product mix changes often and the tools must constantly be adjusted for new product types. High mix/low volume fabs must be managed so that capacity is flexible and can rapidly be adjusted for different processes. When a tool breaks down or needs maintenance, production needs to be continued on other tools.

In this thesis a flexibility model for production in high mix/low volume fabs is developed, such that production can be managed more flexible in workshops.

1.2.2 Basic concepts

Integrated circuits (IC) integrate our daily life more and more and are not only to be found in computers, cell phones and cars, but also clothes and in all sorts of packages. ICs may serve as processors or memories (high volume products) or multi-process products (customized by special demands). Depending on the usage of the IC, the processing differs quite much and manufacturers more and more try to specialize on one of the domains.

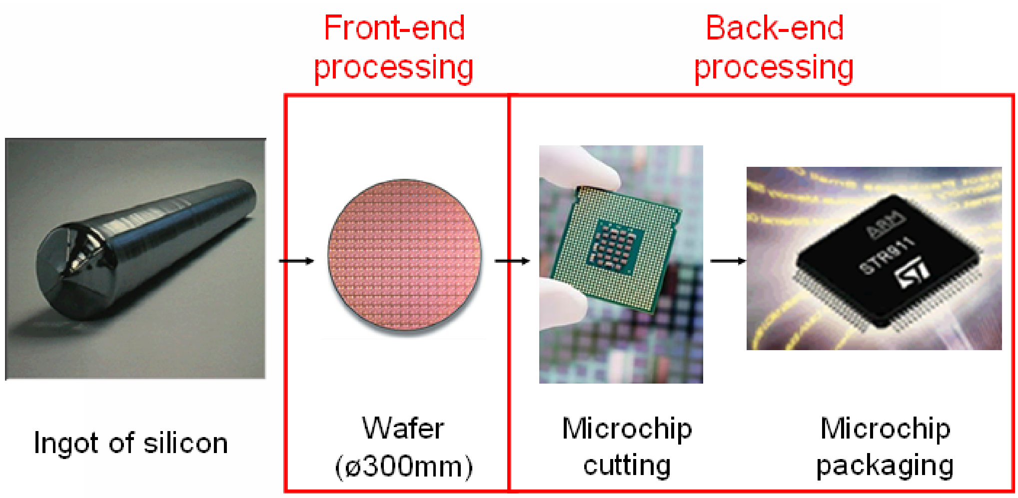


Fig. 1.2 – Microchip processing

Semiconductor production consists of two parts (see Figure 1.2) : Front-end processing and back-end processing. In front-end processing, 0.75 mm thin wafers, which have been sliced out from an ingot of silicon (or another semiconductor material), are used for processing IC's. This process is effectuated in a semiconductor fabrication plant, a so called wafer fab. A wafer with a diameter of 300 mm may contain between 100 and 1000 ICs. When the complex process of manufacturing ICs on wafers has been finished the back-end processing starts : the ICs are cut out from the wafers, their functionality is tested and then they are packaged.

A fully equipped front-end 300mm wafer fab costs over $ 1 billion (2001) to build and the architects need to consider how the fab can optimally use its space in order be as cost effective as possible [103].

1.2.3 Wafer and IC dimensions

The dimensions of wafers have increased until today's 300mm-standard in order to improve yield and reduce costs. Since most of the production errors occur at the border a larger wafer has a higher yield of functional ICs. Also the fixed costs of a large wafer can be split on more products which reduce the cost per finished product.

The first wafers had a diameter of only 25 mm. During the second quarter of 2008 and for the the first time, the 300 mm wafer standard was processed in a larger quantity than the 200 mm standard [25], [108]. The first next generation fab for 450 mm wafers is planned to be built in 2012 [40], [41]. There are however, skeptics of growing dimensions – many 300mm fabs are not yet profitable and manufacturing 450mm wafers will require many new challenges to be solved.

FIG. 1.3 – A 25-wafer lot for 300 mm processing

In 300 mm wafer fabs, up to 25 wafers are transported in a lot (see Figure 1.3) called Front-Opened Unified Pod (FOUP). A fully loaded lot with 300 mm wafers has considerable weight and semiconductor manufacturers are carrying out more internal transport with Automatic Material Handling Systems (AMHS). More about AMHS can be found in [73, 74, 75].

At the same time, as wafers grow larger, the components of the ICs become smaller. The smallest component of an IC is now measured in nanometers ; a so called grid of an IC may be 60, 45 or even 32 nm small.

1.2.4 Fabrication

Front-end processing is performed in several process steps that are repeated many times : deposition, removing, patterning and modification of layers on the wafer. The list below expresses a simplified model of how the different steps are carried out in order to create a layer on the wafer.

- Deposition processes add material layers on the wafers which can be done with different technologies. Typically Chemical Vapor Deposition (CVD) is used to produce a pure and solid film on the wafers. Also for example tungsten (metalization) can be used to create electrical connections between the insulating materials such as silicon dioxide (oxidation process).
- Patterning processes use photolithography technology to add patterns on the ICs. This is done by posing a photoresist and a mask on the wafer, and then letting the wafer be exposed to ultraviolet light.
- Removing processes are carried out at different work areas. Etch processing removes the areas defined by the patterns (dry etch) or remove unwanted bulk from the photoresist (wet etch). In Chemical Mechanical Polishing (CMP), the top of a layer is polished away in order to achieve a flat layer with desired dimensions. Furthermore the stripping process removes contaminants which may harm the properties of the IC.
- Modification of wafer layers can be performed by ion implantation so that the ICs get the right electrical properties.

Figure 1.4 shows the sequence of deposition, photolithography and etch processing.

One of the characteristics of semiconductor manufacturing is the reentrance flow of the production (shown in Figure 1.5). In a modern wafer fab, a product achieves

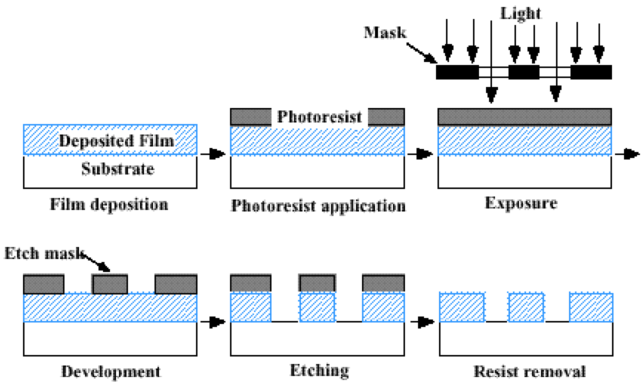


FIG. 1.4 – From deposition via photolithography to removal processing

many layers, and hence passes the same work area many times during its production cycle. Wafers may also be the target for rework when the quality of the last layer does not meet the requirements.

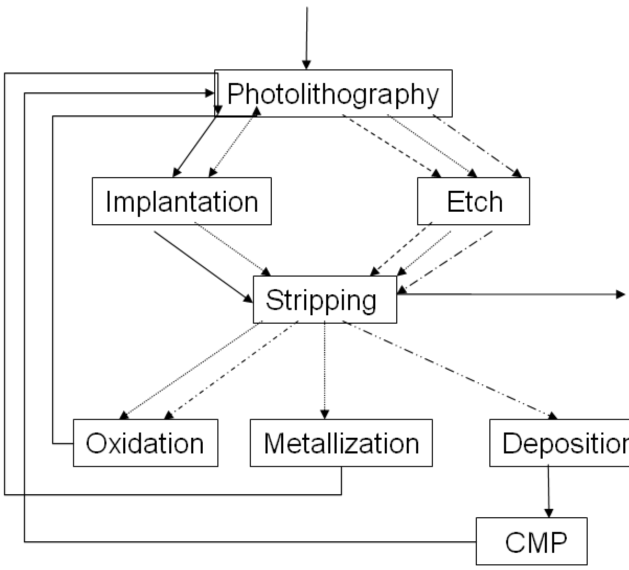


FIG. 1.5 – A simplified reentrance flow model in semiconductor manufacturing [113].

Depending on the complexity of the IC, the number of layers and the priority of its lot, processing a wafer takes from 8 to 30 weeks [87]. The large numbers of wafers, tools, and process steps make semiconductor manufacturing very complex where many new industrial engineering problems occur.

1.2.5 Wafer fab modeling

In semiconductor manufacturing, tools are becoming more and more advanced, and hence the costs of investing in a new tool increases. A photolithography tool may cost over $20 million [35]. As the fab management strives to use such expensive tools as much as possible for production and not tests, detailed models of the manufacturing must be built. Simulation can run these models before a change is realized in the fab, such that possible problems do not effect the production.

Van Campen [103] has designed a model of a whole wafer fab before it was built. In the model, lots are simulated from when they enter into the production system until all processes are finished and the lot exits the system. Both single process task and multiprocess tasks are considered for which waiting times and throughput times have been implemented. Furthermore the flow of the lots between the different workshops in the fab has been modeled. The model has been implemented in a mathematical formal specification language χ [43], which can be used for modeling, simulation and verifications of hybrid systems.

Wafer fab decisions are mainly directed on three levels in the fab [72] : workshops (base), work areas (middle) and the entire wafer fab (top). In workshops (also called work centers), tools with similar processes are grouped together. Workshops are put together to work areas, where consecutive processes of workshops form operations. Example of work areas are the diffusion area (for example oxidation), photolithograhpy area and etching area [70]. Together the work areas constitute the wafer fab with its reentrance flows which are shown in Figure 1.5.

In order to measure the performances of semiconductor manufacturing, different metrics have been developed for the different levels of the fab. The most well-known metric for measuring equipment effectiveness is probably Overall Equipment Effectiveness (*OEE*), which in its original version was developed by Nakajima [79] and which has become standardized by the Semiconductor Equipment and Materials International (SEMI) organization [95]. *OEE* has grown relatively popular as is it a simple metric to use, it combines availability efficiency, operational efficiency, rate efficiency and quality efficiency. Recently, however, some researchers noted some imperfections of the *OEE* metrics, and therefore de Ron et Rooda [22] have developed a new measure called the Equipment Effectiveness (E). Notably E only depends on equipment-dependent states. The same authors have also developed a fab performance metric, called manufacturing performance. Normally fab managers try either to maximize throughput in the fab or minimize cycle time. In the best case, they try to optimize both by studying trade off curves for both measures. As the manufacturing performance measure includes a ratio of both measures, fab managers could more easily obtain an optimum of both throughput and cycle times.

At the local level, local rules are often used which foremost should satisfy the global objectives of a fab [19]. But optimizing something locally does not by default optimize the global performance. In order to optimize global criteria, operational completion times limits are set for the local level, with due dates when lots should be completed in a work area [39]. This is made in order to reduce variability of the lots, which makes it easier to plan at the global level. A way to measure how well due dates of lots can be respected, is the total weighted tardiness (TWT), which measures how delayed are lots in a work area [66, 72, 71]. In order to estimate the times that the lots spend in a workshop, Effective Process Time (EPT) can be used [44, 48]. EPT not only considers process times but also set up times, tool down times and other sources of variability [104].

It is important to use the tools as efficiently as possible and thus reduce tool idle times, particularly in bottleneck areas. In this thesis, a model is developed, which can be used to increase the flexibility of the production to improve performance in the workshops and the entire fab. In order to improve flexibility, qualifications, which are described in next section, will be used.

1.3 Qualifications

Setups in semiconductor fabrication can be divided in two different groups. The first type of setups only need to be performed once and assures that the right temperature, right metal composition and right gas pressure can be obtained on the tool. This is what is called *qualification*. Recipes stay qualified on tools until an operator disqualifies the recipe. This can be done if it is seen that the right characteristics cannot be obtained on the tool or the quality cannot be maintained. The qualification can be removed temporarily or for good.

Furthermore there are also setups that need to be performed each time before the process of a product is started. These setups assure that temperature, metal compositions and gas pressure have the right values *i.e.* the values that have been qualified. This kind of setups normally always needs to be performed before a process except when the previous process was the same as the current one.

Setup leading to qualification is a typical characteristic for the semiconductor industry, and as it will be seen in the literature section, it has not been very well studied until now. This makes Qualification Management (QM) for semiconductor manufacturing a very interesting area to research [53]. Therefore, in the reminder of this chapter, qualifications will be defined more precisely, it will be seen what has been done within QM until now and it will be motivated why it is such an important area to study.

1.3.1 Recipe qualifications in semiconductor manufacturing

To understand what is meant with a qualification in semiconductor manufacturing, it is first necessary to understand what a *recipe* is. Just as in cooking, a recipe in semiconductor production refers to the different *ingredients*, characteristics and actions that need to be carried out for the completion of a process. By ingredients for a semiconductor manufacturing process, different components are concerned which are added for each process, such as metal composition, preparation of photo reticles, temperatures and gas pressures. Different products might need a single recipe, but in order to see if the recipe can be used in a tool, the recipe must be qualified on the tool. In order to qualify a recipe, several setup procedures might be necessary.

Setup tests will verify if the right temperature or gas pressure can be achieved and if the right metal compositions can be handled by the tool, or if the right computer setting exists on the tool. When it has been verified that the recipe is processable on the tool, the recipe is said to be qualified.

All this leads to that qualifications directly and indirectly cost money for the company. An optimal QM strategy for the company should minimize these costs and increase the efficiency of the tools in a workshop by allowing a more flexible way to allocate the capacities (i.e. qualify recipes).

Additionally the work-in-process (WIP) quantities play an important role for the flexibility model which is presented in Chapter 2. With WIP quantity is meant the number of wafers that should be processed at a workshop. In most cases, the number of wafers will be considered but the number of lots or batch quantities could also be considered.

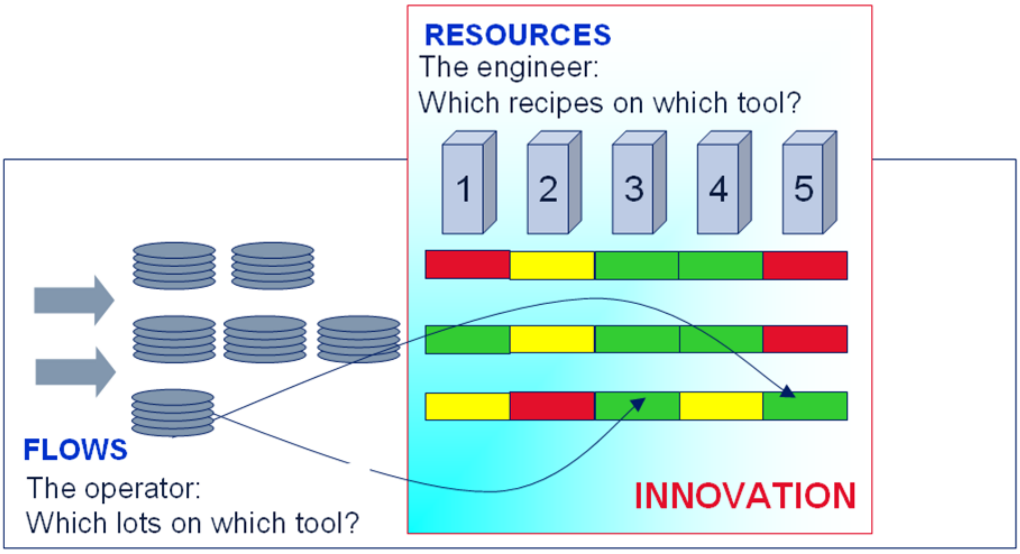


FIG. 1.6 – Good resource management leads to improved product flow.

The novel idea is that performing qualifications can help to better allocate capacity for production such as displayed in Figure 1.6. With additional qualifications (on the horizontal axis), the engineer can enable resources such that the operator has a greater flexibility (see Chapter 2) where to dispatch/schedule the lots in order to improve the flows.

Before motivating the importance of QM the literature on qualifications is studied.

1.3.2 Literature on qualification management

The literature on QM is limited and, due to the increasing complexity of semiconductor manufacturing, articles that describe specific implementations of software often gradually become obsolete. Many articles concentrate on the recipe aspect of QM such as Williams [109]. He denotes a number of basic functions of recipe management : Recipe storage, configuration control, name resolution, recipe modification etc. What he describes is actually an old system, which nowadays with a functional recipe storage and automatic choice of recipes is no longer a problem. Furthermore, Achacoso and Pisapia [2] describe the efforts by factories and their suppliers to develop a standard for recipe management services. The motivation is to enhance performance by improving tool utilization through automation. These articles are useful for understanding how recipes can be stored. But none of them study how qualifications can be used to improve performance in the fab.

Databases can serve as a ground for which QM can be used, but if data is not presented in a comprehensive way, they can be hard to interpret. Yurtsever and Comerford [114] describe a graphical program for visualizing tool statuses in a wafer fab. The program displays the actual layout of the fab at the same time as the process statuses of the tools are shown. However, there is no possibility to directly see where further qualifications will have the biggest effect. Furthermore Pierce and Yurtsever [85] present a Graphical Monitoring System, GraMMs, that has been installed in wafer fabs. GraMMs includes four other programs : Dynamic Dispatch, WIP Monitoring System (WMS), Equipment Management System (EMS), and Throughput Monitoring System (TMS). By allowing easy configuration between different monitoring systems, data can be more easily accessible and the effects from different parts of the system are easier to grasp. Also this system can help to understand the current situation of the fab. There is, however, no way to analyze how further qualifications can improve the output of the fab.

Connection between fault detection and QM has been derived. Ono *et al* [84] present recipe evaluation that quantifies defect distribution on wafers. Recipe inspections can distinguish defect wafers and can separate wafers with defects by qualifying wafers with respect to their defined capability. Furthermore, Zahara and Fan [115] describe an algorithm for recipe qualification, which they implement for a given process, where recipes can be modified after each run. Both these works use qualifications to see how wafer quality can be improved. Still, output performance is missing, which give a motivation to study how qualifications can improve fab output performances.

As the complexity in fabs increases, there has been a tendency to search for methods that decrease the reentrancy flow in the fabs. Ignizio [46] has presented a method for measuring reentrancy of a product line. It makes it possible to conduct research on whether decreasing the reentrancy actually improves the productivity of the fab. Process steps can be considered to be qualified in order to decrease the Degree of Reentrancy (DoR) in a fab. This is something most low mix/high volume fabs try to achieve in order to affect as few product lines as possible. In fact, in high mix/low volume fabs, a high DoR might indirectly be needed since there are too many product lines and not enough tools to separate each line. In such fabs there must be a certain flexibility in order to still be able to process a product when a tool gets overloaded, breaks down or needs maintenance. Robinson *et al.* [88] have listed the five most important capacity loss factors in semiconductor manufacturing stating equipment downtime as the most important factor :

1. Equipment downtime
2. Yield loss
3. Set up
4. Batching policy
5. Dispatch policy

Qualifying recipes on several tools can help avoiding the impact of the equipment downtime, allowing process to continue even if a tool breaks down or needs maintenance.

In other contexts and industries, a few articles have considered QM. Fuchs *et al.* [28] describe the characteristics of QM in supply chain management in a general context. A web-portal for the actors is suggested from where decisions about qualifications can be carried out. It could be possible to implement a similar system for semiconductor manufacturing. However, the research is not well described in the article and therefore it is hard to draw any conclusions on semiconductor manufacturing.

Jordan and Graves [55] and Graves and Tomlin [37] consider process flexibility for product-plant configurations in the automobile industry. They consider that in order to have a more flexible production, a car model can be processed at many sites. Qualifying (or linking as they express it) a new car to a new plant increases the flexibility of the production system for the car manufacturer in terms of capacity and uncertain customer demands of a product.

A similar idea will be introduced in Chapter 2 in order to increase flexibility in semiconductor workshops. To increase flexibility in a workshop, recipes can be qualified on more tools. This can be used both in terms of using capacity more efficiently and having a more robust production system when tools become unavailable.

1.3.3 Importance of qualification management

QM in wafer fabs concerns all components of manufacturing that have to be qualified ; on what grounds the decisions about qualifications should be made and which means can be used to handle the qualifications. It is necessary to understand how these components of a wafer fab are related with QM. Four different QM areas can be identified.

Tool efficiency QM. Qualifications can enable the workload to be better allocated so the tools can be used more efficiently.

Down-time QM. Qualifications on tools such that the work is not too much affected when tools are not available.

Reentrancy reducing QM. Unnecessary qualifications increase the Degree of Reentrancy (DoR) of the production line, which could make the production system unnecessary complex. More on DoR can be found in [46].

APC QM. Yield improvement and tool performance through Advanced Process Control (APC) can be achieved by qualifying recipes for crucial processes on the right tools. APC is a way to control the output of a specific process by studying the interaction between several parameters, such that the behavior of the process can be predicted, for example fault detection for improving yield [11, 77, 78].

The competitiveness in the semiconductor industry is continuously increasing. Semiconductor companies must be able to cope with rapid product changes. Active effective configuration of process qualifications on toolsets for changing product lines must be easy to implement and to maintain. These changes of configurations yearn for active and flexible QM.

One of the more important things that can be achieved with QM is the ability to see which aspects affect possible qualifications and in turn, how qualifications affect the output. Active planning where qualifying needs to be done in order to anticipate and avoid long lead times is needed. If the future WIP quantities (i.e. number of lots to process for each product type) are known, proper qualifications can be conducted in order to effectively handle the planned production volumes.

If performance measures for qualification settings are derived, different configurations could be compared in order to see which qualification (or set of qualifications) leads to the best performances. An optimization model could be set up in order to see which configuration is the most suitable. A company that derives such measures based on its objectives would know what actions are needed in order to reach their objectives. Objectives are different between companies. Examples of objectives are reduced cycle time, increased capacity, bottleneck avoidance and improved flexibility as will be seen in the remainder of this thesis.

In Chapter 2 models are presented, which can be used to evaluate the flexibility in a workshop based on the recipe qualifications on the tools. The impact of such flexible qualifications on scheduling in workshops will be studied. It is shown that flexible qualifications can both decrease the cycle times of the products and reduce the impact of tool unavailability. In fact, tools often become unavailable in wafer fabs, either for maintenance work or when they break down.

The model also considers the changes and the uncertainties that occur in the fab : WIP quantities of different sizes and different recipe mixes, tool breakdowns etc. Moreover, the solution should be robust while considering different possible scenarios of WIP quantities and tool downtime.

There are numerous challenges and branches for new research in the area of QM for wafer fabs. Below a list of different branches are given :
Visualization of qualifications on tools. The recipe, automation sequence, capability statuses on the tools should be properly visualized so that operators, engineers and management have a clear view of what can be processed and what can be qualified. Tools of the same type can be grouped together in a computer program so that similar processes can be easily compared.
Yield improvement. Also Advanced Process Control (APC) has to be integrated in QM. Aspects on how Fault Detection and Classification (FDC) can be an integral part of how the qualifications in the fab should be performed ; which qualifications of processes need to be performed to avoid faults and errors in the production.
Static vs. dynamic and stochastic approaches. It is important that such a model not only can be used in the static case where only the current WIP quantities and the current tool configurations are known, but also for different predicted or plausible scenarios, and for dynamically changing WIP quantities over several time periods. It cannot be taken for granted that the qualifications that are optimal for the current situation are the best ones for the forthcoming time periods.
Reentrancy. Interesting studies can be carried out on how disqualification of productions steps on some tools will affect the trade-off between lost flexibility and decreased reentrancy.
Capacity constraint and planning procedures. With most scheduling and dispatching programs, only feasible WIP quantities are scheduled. The remaining work is left as non-processable. Instead, new process possibilities could be found for such unprocessable work with new qualifications.
Costs and easiness to change qualifications. While considering qualifications, one should not only search for the best configuration settings. Some qualifications can be hard to carry out, and are maybe less desirable to perform than others. There may also be a need for additional expertise or know-how, in order to define various types of qualifications : some qualifications require longer time than others which may hinder the production, and some qualifications can be too costly to carry out.

Such factors should be considered in mathematical models for QM.
Performance measures. Studies on which performance measures must be used to measure how good a set of qualifications is. Example of possible measures could be improved cycle time and capacity or increased flexibility for capacity allocation as presented in Chapter 2.

1.3.4 Flexible qualifications

In this thesis it will be studied how qualifications can be used in order to increase flexibility of capacity allocation in semiconductor workshops. What is meant with flexibility is researched and defined in Chapter 2. Capacity can only be allocated on tools which have already been qualified. If there are only a few recipes qualified per tool, there is not so much flexibility for operators to allocate workload on the tools. In this thesis, two definitions of workload are considered : (1) the number of jobs (wafers, lots or batches), which will be called WIP quantity throughout the thesis, and (2) the total production times on the tools. On the places where the term workload is used, it will be clearly stated which definition is used. The idea is that additional recipe qualifications in a workshop lead to increased flexibility where to process the recipes. In Table 1.1, an example of qualifications in a workshop with three recipes and three tools is shown.

	Tools			
Recipes	A	B	C	WIP quantities
1	-	X	X	50
2	X	-	?	400
3	?	X	X	450

Tab. 1.1 – Example of a toolset with qualified recipes.

There are 50 wafers needing Recipe 1, 400 wafers needing Recipe 2 and 450 wafers needing Recipe 3. An “X” signifies that the recipe is qualified and thus can be processed on that tool : Recipe 1 is qualified on Tool B and Tool C, Recipe 2 on Tool A, and Recipe 3 on Tool B and Tool C. The question mark “ ?” signifies that the recipe could be qualified on that tool : Recipe 2 could be qualified on Tool C and Recipe 3 could be qualified on Tool A.

If there would exist a flexibility measure the flexibility of the workshop could now be evaluated, and it could be compared with the flexibility measure for the cases where either Recipe 2 is qualified on Tool C or Recipe 3 is qualified on Tool A. In this way it would be possible to see which qualifications would optimize flexibility in a workshop. In Chapter 2 such flexibility measures are developed. With these flexibility measures, it will be possible to see which qualifications or recipes on tools optimize the flexibility in a workshop.

1.3.4.1 Flexibility and robustness of qualifications

Flexibility and robustnesses are closely linked to each other. If a flexible system can be seen as a system which allows several different decisions, a robust system is a system which will not lead to a poor solution independently of actual events that may occur. Obviously, by having the possibility to make several different decisions, it is possible to better adapt a solution to events. Hence, a flexible system is often robust. On the other hand, it is not sure that a robust system always allows several decisions, i.e. is flexible.

An important and often used contribution to robust optimization has been developed by Kouvelis and Yu [60]. They have defined three different approaches of robust solutions for optimization problems with several possible scenarios. *Absolute Robustness* is the most conservative approach, which only tries to minimize the worst case (for a minimization problem). Less conservative is *Robust Deviation*, in which the goal is to minimize the deviations of the solution for all scenarios. Finally they also developed *Relative Robustness*, in which the maximum deviation is minimized relative to all scenarios.

Instead of considering scenarios, Ben Tal and Nemirovski [12, 13] propose a robust approach for linear optimization problems using oval feasible spaces for uncertain parameters. This approach has the disadvantage of making linear problems non-linear, which leads to more complex problems and longer computational times. Therefore Bertsimas and Sim [14, 15] propose another approach with linear intervals for uncertain parameters. A different approach has been proposed by Rossi [90] and

Rossi *et al.* [92, 91]. A solution is said to be robust in an interval if it can guarantee a global optimal performance with a defined range.

1.3.4.2 A robust network approach

If qualifications are seen as links between recipes and tools, then the problem can be seen as a network problem. This problem is similar to the so called Capacitated Facility Location Problem (CFLP), which is a common problem in telecommunications and has been studied by Aardal *et al.* [1].

In the CFLP, a set of clients (recipes) should be connected to a number of sites (tools) such that the capacity of the sites is sufficient for the client's demand (WIP quantity). The objective of CFLP is to minimize the costs of opening facilities and openings links between facilities and sites. In Figure 1.3.4.2, the red circles represent clients and the blue squares the sites where the clients' demands can be fulfilled.

As far as we known, flexibility has not been studied for CFLP. Instead of flexibility, robustness of the network can be studied. Robust solutions of a special case of CFLP, the Capacitated Concentrated Location Problem (CCLP), has been studied by Johnzén in his master thesis [49]. CCLP considers the case when every site has to be connected to exactly one facility. Figure 1.3.4.2 shows how the clients can only be connected to one site in CCLP. A robust approach of qualifications would be interesting to research, but lies beyond the scope of this thesis.

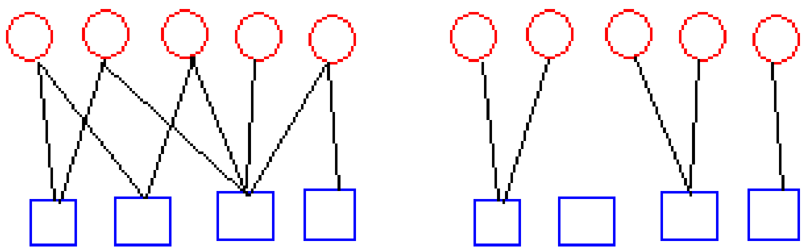


FIG. 1.7 – CFLP and CCLP network problems.

1.4 Conclusions

In this chapter some basic concepts in industrial engineering and semiconductor manufacturing were presented. This information will help to understand the problematics throughout the reminder of the thesis. It has furthermore been explained that production in semiconductor manufacturing is complex, and that there are many aspects that need to be considered in order to optimize production. Tools in different work areas have different characteristics. It has, however, been noticed that for most areas recipes need to be qualified in order to perform processes. It has both been explained what is meant with a recipe and how they are qualified on tools. Next chapter will go into depth in the area of qualification management. Qualification management is the mean which is used in the thesis for increasing flexibility of capacity allocation in wafer fab workshops.

It has been shown that QM is an important area for wafer fab management. In spite of that, only little research has been made on the topic. Therefore the importance of qualifications has been motivated and the challenges of QM area have been enumerated. Resolving these challenges may lead to many benefits for semiconductor manufacturing companies : avoiding bottlenecks, reentrancy reducing, increasing capacity and decreasing cycle times.

It has been argued that qualifications cost money and takes time to setup and to maintain. Therefore it is not possible to qualify all recipes on all tools. A way to see how a minimum of qualifications can be performed in order to improve the production effectiveness needs to be studied. The question still remains how production can be performed more effectively.

In this thesis, it will be studied how qualifications can increase flexibility of capacity allocation in semiconductor manufacturing workshops. An example of that has already been studied and, in Chapter 2, flexibility measures are presented. With these measures it is possible to evaluate the flexibility for different qualification settings in a workshop. The functionality of the measures is explained and examples are given on how the measures can be used. With the flexibility measures it is possible to see which qualification should be chosen for recipes on tools in a workshop. The properties of the measures are studied in Chapter 3. Heuristics are studied in order to determine qualifications which optimize (or nearly optimize) the flexibility in a workshop, within reasonable time.

Further extensions to the flexibility model are proposed in Chapter 4. Finally the impact of flexible qualifications is tested on scheduling simulators for different workshops.

It will be shown in Chapter 5 that this can improve the performance of the workshops and in the end the entire fab. It may also make the production less sensitive to tool downtimes and hence create more robust production systems.

Beyond the scoop of this thesis to improve flexibility with QM, other objectives can be put in focus. In this chapter a couple of objectives were enumerate : tool efficiency QM, down-time QM, reentrancy reducing QM and APC QM. Although these areas will not be directly research in this thesis, they should no be forgotten and further research are needed. Adjacent areas can surely also be found.

Chapitre 2

Modeling Flexibility for Qualification Management

Investments for tools in semiconductor manufacturing facilities (*wafer fabs*) are higher than in most other industries – tools may cost over $20 million [35]. It is therefore crucial to use the tools as wisely as possible and to smooth bottlenecks in the fab. Furthermore wafer fabs are characterized by a high variability in the demand of products and high downtime rate for the tools, which require that wafer fab management has strategies to deal with these uncertainties in the production.

In Chapter 1 it was explained that, before processing a lot on a tool, the recipe of the corresponding process needs to be qualified on the tool. The recipe of a process contains definitions on how the tool should conduct the process : temperature, gas pressure, metal composition etc. Qualifying a recipe on a tool takes time, needs manpower and know-how and hence indirectly costs money for the company. Hence, qualification management has been studied. It was argued that only a minimum number of qualifications should be qualified. On the other hand, by actively planning and conducting qualifications of recipes on tools, tool efficiency can be improved.

In this chapter, it is first motivated why it is necessary to model flexibility. The literature on flexibility in manufacturing systems is studied. Thereafter several measures are proposed to evaluate the flexibility of a set of tools that must process a given set of lots with different recipes. For two of the measures, it is needed to find the optimal distribution of the workload (WIP quantity or production time). Therefore different distribution methods are studied.

2.1 Motivations for Modeling Flexibility

The aim of this chapter is to propose means to see which qualifications improve the flexibility of a production system. It would be convenient if flexibility could be measured such that two qualifications could easily be compared with each other to see which qualification improves the flexibility the most. However, before flexibility measures can be developed, it is needed to understand what is meant with flexibility for production systems. Therefore ideas and motivations will be suggested and challenged, which will be clarified with examples. Suggestions from the literature will be reviewed in order to see if other researchers already have defined suitable measures.

To start with, it is needed to understand what is meant with flexibility in a manufacturing engineering context. The list in Table 2.1 present conditions on how to achieve flexibility through qualifications.

1. To have the flexibility to perform WIP quantities for a recipe on at least one tool that should be qualified for this recipe.
2. An additional qualification should not decrease the flexibility.
3. An additional qualification should not decrease the total production time.
4. It is important to have more qualified tools for recipes with large WIP quantities.
5. It is not important to qualify additional tools for recipes where many tools have already been qualified.
6. It is important to optimize the workload balance for WIP quantities and in particular production times.
7. It is important to minimize the total production time.

Tab. 2.1 – A list on the dependency between qualifications and flexibility.

It might not be obvious that optimized workload balance and minimized production times lead to more flexibility. The argument why Conditions 6 and 7 increase the flexibility is that process engineers can plan when processes will start in a more flexible way when they are scheduled. In addition, these conditions can also lead to improved cycle times. It should furthermore be considered that a set of qualifications which effectively balances WIP quantities and production times does not necessarily minimizes the total production time and vice versa. Therefore the trade-off between Conditions 6 and 7 need to be considered when qualifications are performed. Optimal workload balancing is further studied in Section 2.4. It should also be noted that, although there might be no qualified tools for one recipe in a workshop as specified in the first condition, it does not mean that there is no flexibility for the production of the whole workshop. A discussion on this can be found in Section 2.6.

The idea in this thesis is to evaluate the flexibility of capacity allocation in workshops. Additionally operators have some degree of freedom to distribute WIP quantities on the tools in a workshop. The example below illustrates an example where the operator is able to optimally balance the WIP quantities and production times on the tools in a workshop.

Example 2.1. *In Table 2.2, a production system is displayed with three tools (A, B and C) which can produce three different recipes (1, 2 and 3). The WIP quantities (i.e. number of wafers) for each recipe are also given (400 for all recipes). Recipe 1 is qualified on Tool A, Recipe 2 on Tool B and Recipe 3 on Tool C. On the last row in Table 2.2 it can be seen that WIP quantities can be well balanced on all three tools.*

	Tools			
Recipes	A	B	C	WIP quantities
1	X	-	-	400
2	-	X	-	400
3	-	-	X	400
Distribution	400	400	400	

Tab. 2.2 – Example of a toolset with qualified recipes.

Moreover, when an external element affects the production, it still needs to be continued. For example, when a tool is down, production can be moved to other tools. If one of the tools in Table 2.2 would break down, the production would be disrupted. For example, if Tool A would break down, the production of Recipe 1 cannot be continued. Therefore flexibility of continuing the production on another tool needs to be anticipated.

The following example shows qualifications of recipes on tools such that the production can be continued even if one tool is unavailable for production.

Example 2.2. *In Table 2.3, the same production system as the previous example is displayed with three tools (A, B and C) which can produce three different recipes (1, 2 and 3). The WIP quantities (i.e. number of wafers) are also the same for all recipes (400). The recipes are, however, qualified differently. Recipe 1 is qualified on Tools A and B, Recipe 2 on Tools B and C, and Recipe 3 on Tools C and A. On the last row in Table 2.3, it can be seen that WIP quantities can still be well balanced on all three tools.*

	Tools			
Recipes	A	B	C	WIP quantities
1	X	X	-	400
2	-	X	X	400
3	X	-	X	400
Distribution	400	400	400	

Tab. 2.3 – Example of a toolset with qualified recipes.

If a tools becomes unavailable in the toolset displayed in Table 2.3, the production of all recipes can still be continued. If, for example, Tool B (where Recipes 1 and 2 are qualified) becomes unavailable, Recipe 1 can still be processed on Tool A and Recipe 2 can still be processed on Tool C.

It can be hard to evaluate the flexibility of a toolset just by looking at the qualifications, especially when there might be more than ten tools and hundreds of recipes. The literature on flexibility will be reviewed in order to see what has previously being done on flexibility. Measures will later be developed such that the flexibility of the qualifications in a toolset can be evaluated.

2.2 A Literature Review on Flexibility

Many articles have been written on flexibility during the last 25 years. De Toni and Tonchia [24] have defined flexibility as : "*...an ability to change something.*" A more specific definition for manufacturing is made by Aubry *et al.* [8] : "*...the ability to undergo modifications involving an acceptable loss of performances.*" Another definition of manufacturing flexibility would be the ease to go from one production state to another.

Different classes of manufacturing flexibility have been proposed in the literature. A classification of the eight different types of flexibilities in manufacturing was made by Browne [17]. Using these classes, Sethi and Sethi [96] sorted previous research works on manufacturing flexibility. Since then, flexibility has been a rather well defined area.

In spite of that, only a few authors have tried to develop mathematical measure of flexibilities. De Toni and Tonchia [23] mention that "*...the measure of flexibility is still an under-developed subject*", as they themselves define three groups of measures : direct, indirect and synthetic, and within these several subgroups.

An analysis of existing measures have been written by Giachetti *et al.* [31]. In their report they mention that many measures are only vaguely defined and that they are not quantified which make them hard to use. It should, however, be noted that some authors have actually developed flexibility measures. Das [20] has developed a measure based on tool efficiency to measure the ease with which processes can be changed on different tools. In Rossi [90] the price of flexibility is considered, while going from one production plan to another. A cycle time reduction strategy model for a wafer fab has been developed by Potti and Whitaker [86] who, among other things, use qualifications for adding flexibility in the CVD TIN work area which they recognize will improve cycle times. However they do not mention in what way the flexibility was increased by the qualifications.

For machine flexibility, two different measure types have been developed by Wahab *et al.* [107] : operational capability-based and time cost-based. The authors of the same study have also proposed a generic model which combines these measure types. A drawback with the machine flexibility is that it is only considering one tool and not the whole toolset. Additionally, Lai and Hui [61] have developed a metric for measuring the flexibility for a process to run considering an expected limit of uncertain parameters.

Jordan and Graves [55] have studied process flexibility for an automobile company with several production plants. They use the notation *no flexibility* when company's plants cannot produce more than one product each (in this thesis a workshop is considered to have no flexibility when no product can be processed anywhere) and *total flexibility* is defined as the configuration where the products can be produced in all plants. Furthermore they measure the flexibility as the probability of shortfall of a given configuration compared to the shortfall for total flexibility. Shortfall occurs when the capacity of the plants is not enough to satisfy the expected demand of the clients. By letting a product be produced at several sites, the capacity may be increased although there might be a trade-off for capacity for other products at the plants, compared to a less flexible manufacturing system.

This measure does, however, not consider the two of the factors that were discussed in the motivation part of this section : workload balance and anticipation of tool unavailability (car manufacturing plants do not break down as often as semiconductor tools). Therefore, in next section, measures serving these objectives are developed.

Nevertheless the flexibility structure model developed by Jordan and Graves could be implemented also for semiconductor workshops. In the same way as Jordan and Graves suggest *chaining* the production of a car in a new factory in order to increase flexibility, in this thesis recipes will be *qualified* on tools in order to increase flexibility.

Later Graves and Tomlin [37] developed a similar process flexibility model for the whole supply chain of an automobile company. Akşin *et al.* [4] have also used this model to study flexibility structures for multi-department structures corresponding to different kinds of client service requests, where the agents of the departments have different skills, where the maximum throughput is used as performance measure.

To our knowledge, no author has developed any models for measuring flexibility of different qualification settings in a workshop in order to balance workload and avoid disruption of production when tools are unavailable. Therefore, in this thesis, four flexibility measures will be presented. They are chosen so that they evaluate the flexibility of recipe qualifications on tools. This may increase both robustness and the possibility to well balance the WIP quantities and production times on tools.

2.3 Flexibility Measures

In order to evaluate flexibility of a toolset concerning the recipe qualifications on the tools, flexibility measures have been developed. The idea is that additional qualifications of recipes on tools in a workshop can increase the flexibility to continue to process lots under random circumstances. The way to do this is to use qualification management as presented in Chapter 1. Flexibility needs to be maintained although tools break down or need maintenance. This should however, not be done at all costs, balancing of WIP quantities and production times still need to be maintained and it is preferred to process the products as fast as possible.

In order to understand how qualifications can increase flexibility in an optimal way, flexibility measures that consider these criteria need to be developed. Once such measures have been developed, it is possible to compare the current flexibility in a workshop with the flexibility of the workshop if additional qualifications were performed.

Four flexibility measures have been developed [51, 52, 50]. Their values lie between 0 and 1, where 1 denotes maximum flexibility. In order to increase visibility for the operators in the fab, the values are expressed as percentages between 0 and 100%. The following parameters are necessary for the definition of the flexibility measures.

NQT_r	Number of qualified tools for recipe r.
WIP_r	WIP quantity for recipe r.
$TP_{r,t}$	Throughput rate in wafers per hour for recipe r on tool t.
γ	Flexibility balance exponent (> 1).
R	Number of recipes.
T	Number of tools.

$$OQ_{r,t} = \begin{cases} 1 \text{ if recipe } r \text{ is proposed to be qualified on tool } t, \\ 0 \text{ otherwise.} \end{cases}$$

$$Q_{r,t} = \begin{cases} 1 \text{ if recipe } r \text{ is already qualified on tool t,} \\ 0 \text{ otherwise.} \end{cases}$$

The number of qualified tools is calculated as $NQT_r = \sum_{t=1}^{T} Q_{r,t}$. The reason why γ is defined to be > 1 is that leads to that the two flexibility measures which use γ, will be strictly convex which lead to that the optimal balance can be found for these functions (see Section 2.4). Additionally, two of the measures contain the following variables, which are obtained by optimization procedures described in Section 2.4 :

$WIP_{r,t}$	WIP quantity of recipe r assigned to tool t.
$WIP(t)$	Total WIP quantity assigned to tool t, $WIP(t) = \sum_{r=1}^{R} WIP_{r,t}$
$C(t)$	Total production time assigned to tool t, $C(t) = \sum_{r=1}^{R} WIP_{r,t}/TP_{r,t}$

2.3.1 Toolset flexibility

A way to improve flexibility for cases when tools become unavailable is developed. It can be argued that the only way to continue production, when a tool becomes unavailable, is if the recipes are qualified on backup tools. This is especially important for recipes with a high WIP quantity.

Using this argument, the toolset flexibility measure (2.1) is developed. It stresses the importance of having many qualified tools for recipes with high WIP quantities. This is done by multiplying the variable NQT_r with the WIP quantity of the same recipe r. In this way qualifications on tools for recipes with high WIP quantities will be valued more.

$$F_{old}^{TS} = \frac{\sum_{r=1}^{R} (NQT_r \times WIP_r)}{T \times \sum_{r=1}^{R} WIP_r} \tag{2.1}$$

If all tools would be qualified for all recipes, then NQT_r would equal T for all recipes. Thus the sum in the numerator would equal the product in the denominator and $F_{old}^{TS} = 1$.

It can be argued that it is not as important to qualify additional tools for recipes where many tools have already been qualified [52]. Therefore, a new formulation for the toolset flexibility has been defined (2.2). When a new qualification is considered, the earlier measure will always propose a qualification for the recipe with the highest WIP quantity. For the new formulation, it is also considered important how many tools are already qualified for the recipe.

$$F^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} \frac{WIP_r}{NQT_r}} \tag{2.2}$$

If all tools are qualified on all recipes, then $NQT_r = T$, and thus the sum over all recipes will equal $\sum_r WIP_r/T$ and F^{TS} is equal to 1.

2.3.1.1 Similarities with machine flexibility

Some of the characteristics of the toolset flexibility are similar with machine flexibility. According to the definition of [96] : "*Machine flexibility (of a machine) refers to the various types of operations that the machine can perform without requiring a prohibitive effort in switching from one operation to another.*" On the contrary, the toolset flexibility considers the flexibility for the operations (recipes) in the toolset, instead of the tool perspective which is the case for machine flexibility. However, increasing the machine flexibility normally increases the toolset flexibility and vice versa. More importantly, the machine flexibility only considers one tool, whereas the toolset flexibility considers the whole toolset.

2.3.1.2 Comparing the toolset flexibility measures

The question is whether the old toolset flexibility measure (2.1) or the new toolset flexibility measure (2.2) better models the capacity allocation. It can be answered using an example with two recipes and five tools. Recipe 1, with 10 wafers, is qualified on one tool. Furthermore, Recipe 2 with 11 wafers is qualified on four tools. The question is, if an additional qualification is considered, for which recipe should the qualification be performed. By just considering the fact that both recipes have more or less the same WIP quantity, but that Recipe 2 has much more capacity where its WIP quantity can be processed, normally another tool should be qualified for Recipe 1. The values of the old flexibility measure for qualifying Recipe 1 is 0.61 and 0.62 for qualifying Recipe 2. According to this a new tool should be qualified for Recipe 2. This is not logical and hence it shows that the old toolset flexibility measure is not always valid. The values of the new toolset flexibility measure are 0.54 for qualifying a tool for Recipe 1 and 0.34 for qualifying a tool for Recipe 2.

This corresponds to the expected result and, hence, the new toolset flexibility will be used in the remainder of the thesis.

2.3.2 WIP flexibility

As argued earlier, the ability to balance the WIP quantities should not be neglected. Therefore a measure that evaluates how the WIP quantities can be balanced on a set of tools has been developed.

The WIP flexibility measure F^{WIP} (2.3) increases when $\sum_{t=1}^{T} WIP(t)^{\gamma}$ decreases ; the total WIP quantity is constant but, as the WIP becomes more balanced, the sum in the denominator decreases. By increasing the parameter γ, the value of F^{WIP} changes, but a better balanced WIP quantity would still give a larger F^{WIP} than a worse balanced WIP quantity.

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} \left(WIP(t)\right)/T\right)^{\gamma}}{\sum_{t=1}^{T} WIP(t)^{\gamma}} \tag{2.3}$$

Since $WIP(t) = \sum_{r=1}^{R} WIP_{r,t}$, (2.3) can be reformulated as in (2.4).

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} \sum_{r=1}^{R} WIP_{r,t}/T\right)^{\gamma}}{\sum_{t=1}^{T} \left(\sum_{r=1}^{R} WIP_{r,t}\right)^{\gamma}} \tag{2.4}$$

If the WIP quantities can be perfectly balanced on all tools, then distribution for all tools t is such that $WIP(t) = \sum_{t=1}^{T} WIP(t)/T$. Thus the sum in the denominator is equal to the numerator and $F^{WIP} = 1$. However, in order to use the WIP flexibility measure, *the optimal balance of the WIP quantities on the tools* needs to be determined. To do that, **it is required to solve an optimization problem** (see Section 2.4.1).

2.3.3 Time flexibility

Different recipes often take differently long times to be processed on different tools. Therefore it should not only be considered to optimally balance the workload

(WIP quantities or production times) on the tools, but also if products can be processed fast. The first idea is to modify the WIP flexibility measure. Instead of considering the balance of the WIP quanties on the tools, process times are considered.

From F^{WIP} (2.3), the total WIP quantity $\sum_t WIP(t)$ and the WIP quantities per tool $WIP(t)$ have been exchanged with the total production time $\sum_t C(t)$ and the production time per tool $C(t)$. The time flexibility measure is defined in (2.5). However, contrary to the WIP flexibility measure, the time flexibility measure (2.5) depends both on how the production times are balanced on the tools and how the total production time is minimized. This comes from the fact that, whereas the total WIP quantity $\sum_t WIP(t)$ is constant, the total production time $\sum_{t=1}^{T} C(t)$ is variable. As with the WIP flexibility, **it is required to solve an optimization problem** to find the optimal WIP balance for the time flexibility measure (see Section 2.4.2). Through the flexibility balance exponent γ, it is possible to choose whether minimization of the total process time or maximizing the balancing is important (see Section 2.4.2).

$$F_{old}^{time} = \frac{T \times \left(\sum_{t=1}^{T} (C(t))/T\right)^{\gamma}}{\sum_{t=1}^{T} C(t)^{\gamma}} \tag{2.5}$$

One of the ideas of the time flexibility measure (2.5) was that production times should be minimized to gain flexibility. This is, however, being contradicted in (2.5) : Since the total production time $\sum_{t=1}^{T} C(t)$ is variable, the total production time may be increased when the time flexibility measure (2.5) is being maximized. To avoid this effect, a new version of the time flexibility measure with a constant value has been proposed (2.7) [52]. As with the WIP flexibility measure, the variable is kept in the denominator and, for the numerator, a normed term is defined. The new constant C_{ideal} is the maximum value of the sum of $C(t)^{\gamma}$ when all tools that can be qualified for the recipes are qualified; The optimal value of the denominator $\sum_{t=1}^{T} C(t)^{\gamma}$ in (2.5). The definition of C_{ideal} is stated in (2.6).

$$\begin{aligned} &C_{ideal} = \max \sum_{t=1}^{T} C(t)^{\gamma} \\ &\text{with } Q_{r,t} = 1 \quad \forall r \in \{1,..,R\}, \forall t \in \{1,..,T\} \end{aligned} \tag{2.6}$$

	Tool A	Tool B	F^{time}_{old}	F^{time}
Strategy 1	10 h	10 h	0.67	0.51
Strategy 2	10 h	1 h	0.59	1.00

TAB. 2.4 – Production times for a set of tools, using two different strategies

The new time flexibility measure F^{time} (2.7) is therefore equal to 1 if all recipes are qualified. Again **it is required to solve an optimization problem** – *to find the optimal balance for* $C(t)$.

$$F^{time} = \frac{C_{ideal}}{\sum_{t=1}^{T} C(t)^{\gamma}} \tag{2.7}$$

Since $C(t) = \sum_{r=1}^{R}(WIP_{r,t}/TP_{r,t})$, (2.7) can be reformulated as in (2.8).

$$F^{time} = \frac{C_{ideal}}{\sum_{t=1}^{T} \left(\frac{\sum_{r=1}^{R} WIP_{r,t}}{TP_{r,t}} \right)^{\gamma}} \tag{2.8}$$

Note that F^{time} has high values both when the production times are optimally balanced on the tools and when they are minimized.

2.3.3.1 Comparing the time flexibility measures

In the example of Table 2.4 two different strategies for distributing the WIP quantities are considered; one which aims at balancing the process times on the tools and one which aims both at minimizing and balancing the production times. The two strategies lead to the two solutions displayed in Table 2.4, where F^{time}_{old} and F^{time} are calculated with $\gamma = 2$. Considering the production times in this table, the second strategy is preferable since the total production time is smaller.

Since strategy 2 is considered preferable, the lower value of the old time flexibility measure for strategy 2 contradicts what is expected. On the contrary, the value of the new time flexibility measure (with $C_{ideal} = 10^2 + 1^2 = 101$) is larger for strategy 2. A result which better measures what is expected. Hence, the new time flexibility measure is more suitable.

2.3.4 System flexibility

Instead of using all the previous measures separately for measuring flexibility in a workshop, one measure that can consider the effect of all this measures can be developed. Such a measure could include two or three of the measures is one measure.

Such a measure has been called the system flexibility measure. For the system flexibility measure F_{old}^{SYS} (2.9), the toolset flexibility measure (2.1) is combined with the WIP flexibility measure (2.3). This is done by multiplying the toolset flexibility measure with the WIP flexibility measure.

$$F_{old}^{SYS} = F^{TS} \times F^{WIP} \tag{2.9}$$

When both the toolset flexibility measure and the WIP flexibility measure are equal to 1, then the system flexibility measure is also equal to 1. This will only occur when all recipes are qualified on all tools.

Since the time flexibility measure works similarly as the WIP flexibility measure, it is possible to replace F^{WIP} with F^{time} the system flexibility measure (2.9).

In the original formulation of the system flexibility measure (2.9), it is not possible to increase the importance for one of the included flexibility measures (F^{TS} and F^{WIP} or F^{TS} and F^{time}). One way to do it would be to put a parameter larger or zero over each term. But in order to better control the variables another approach where the terms are added to each other instead of multiplied. In this version of the system flexibility measure the components are added with a parameter associated to each component (2.10) [52]. The parameters are dependent on each other such when one parameter is increased the others will decrease. In this way it is possible to let one of the flexibility measures be more important by increasing the value of the associated parameters a, b or c, or even to exclude one measure by setting the corresponding parameter to 0.

$$F^{SYS} = a \cdot F^{TS} + b \cdot F^{WIP} + c \cdot F^{time} \tag{2.10}$$

In (2.10), the parameters a, b and c are all in $[0, 1]$ and are defined such that

$a + b + c = 1$. This ensures that the F^{SYS} is also in in $[0, 1]$.

2.3.4.1 Comparing the system flexibility measures

The main difference between the old system flexibility measure and the new measure is that, with the new version, it is possible to control the importance of the different flexibility measures.

2.3.5 Examples

In order to see how the flexibility measures can be used, the examples in Tables 2.5, 2.6 and 2.7 will be used. In Table 2.5, a toolset with three tools (A, B, C) and three recipes (1, 2, 3) is displayed. Recipe 1 has a WIP quantity of 10 wafers and is qualified on Tool A. Recipe 2 has a WIP quantity of 30 wafers and is qualified on Tool B. Recipe 3 has a WIP quantity of 40 wafers and is qualified on Tool C. If the WIP quantities are distributed on the tools, Tool A will have a WIP quantity of 10 wafers, Tool B of 30 wafers and Tool C of 40 wafers.

	Tools			
Recipes	A	B	C	WIP quantities
1	X	-	-	10
2	-	X	-	30
3	-	-	X	40
Distribution	10	30	40	

Tab. 2.5 – Example of a toolset with qualified recipes.

Let us assume that in the toolset two additional qualifications are possible. It is either possible to qualify Recipe 2 on Tool A as shown in Table 2.6 or Recipe 3 on tool B as shown in Table 2.7.

To calculate F^{time} throughput rates of the processes are needed (see Table 2.8). Throughput rates specify how many wafers for a recipe can be processed per hour on a tool.

By calculating the flexibility measures for the qualifications in the toolset, it is possible to see which of the qualifications increases flexibility the most. The flexibility

	Tools			
Recipes	A	B	C	WIP quantities
1	X	-	-	10
2	**X**	X	-	30
3	-	-	X	40
Distribution	20	20	40	

TAB. 2.6 – Example where Recipe 2 is qualified on Tool A.

	Tools			
Recipes	A	B	C	WIP quantities
1	X	-	-	10
2	-	X	-	30
3	-	**X**	X	40
Distribution	10	35	35	

TAB. 2.7 – Example where Recipe 3 is qualified on Tool B.

measures for the qualifications in Table 2.5 is referred as *Init* In Table 2.9, the flexibility measures have been calculated for the three cases. For the example, the parameters have been set to : $\gamma = 4$, $a = 0.4$, $b = 0.3$ and $c = 0.3$.

From the results in Table 2.9, it can be seen that, if the process engineer only wants to increase F^{TS}, Recipe 3 should be qualified on Tool B ($OQ_{3,B} = 1$) ; the recipe with the largest WIP quantity will be more robust considering tool breakdowns. If the process engineer instead wants to increase F^{WIP} or F^{time} Recipe 2 should be qualified on Tool A ($OQ_{2,A} = 1$) should be performed ; the WIP quantities and production times can be better balanced on the tools. If the process engineer

	Tool		
Recipe	A	B	C
1	100	100	125
2	50	75	100
3	75	100	125

TAB. 2.8 – Throughput rates expressed as WIP quantity that can be processed per hour

Measure	*Init*	$OQ_{2,A} = 1$	$OQ_{3,B} = 1$
F^{TS}	0.33	0.38	0.54
F^{WIP}	0.45	0.53	0.50
F^{time}	0.52	0.84	0.52
F^{SYS}	0.42	0.56	0.52

TAB. 2.9 – Flexibility measures for the examples

γ	1.1	2	3	4
F^{WIP}	0.99	0.82	0.62	0.45
F^{time}	0.91	0.85	0.68	0.52

TAB. 2.10 – Flexibility measures for different values on γ

wants to consider a combination of all measures, by using F^{SYS}, qualifying Recipe 2 on Tool A ($OQ_{2,A} = 1$) is recommended.

2.3.5.1 Example γ

While changing the value on γ the values of F^{WIP} and F^{time} and thus also F^{SYS} when b or/and c are > 0. In Table 2.10, the different values of F^{WIP} and F^{time} are shown for the example in 2.5. As will be seen in the Numerical experiments in Section 2.5.2 the value on γ will also influence the outcome of F^{time} for the trade-off between balancing and minimizing the production times.

2.4 Optimizing Workload Balancing

In order to use two of the flexibility measures, an optimal workload balance needs to be found for WIP quantities and production times respectively. Two workload balancing methods have been developed. The first one which determines an optimal balance of the WIP quantities for F^{WIP} and a second one which is optimal distribution of the production times for F^{WIP}

It can be discussed if it is not always better to optimally distribute the production times than the WIP quantities. In the workfloor it is indeed optimized produciton times which will lead to a more efficient manufacturing. However as it will be seen in Section 2.5 the balancing algorithm for F^{WIP} is faster or much faster than the one for F^{time}. It will also be seen that in workshops where throughput times are quite similar for most processes the solutions for F^{WIP} and F^{time} are mostly identical. On the other hand in workshops where the throughput times differ much it is worthwhile to use the time flexibility.

2.4.1 Workload balancing for WIP flexibility F^{WIP}

In order to calculate the WIP flexibility measure, it is necessary to optimally balance the workload of the WIP quantities (e.g. wafers or lots) on the tools. If this is not the case, the measure will take a different value, even if the qualifications are the same (compare the optimally balanced WIP quantities in Table 2.11 to the case where WIP quantities are not optimally balanced in Table 2.12). Since it is not acceptable that the same qualification set may have two different WIP flexibility measures, a balancing algorithm for optimally distributing the WIP quantities on the tools has been implemented.

	Tools			
Recipe	A	B	C	WIP quantities
1	-	25	25	50
2	300	-	**100**	400
3	-	250	200	450
WIP Balance	300	300	300	

Tab. 2.11 – An example where WIP quantities are optimally balanced on the tools.

As mentioned above, to calculate the WIP flexibility measure, WIP quantities must be optimally balanced. The WIP balancing algorithm is illustrated in Figure 2.1, where one considers one recipe at the time while distributing the WIP quantities. Such a distribution is called *local distribution*. It will be proved that the method optimizes the mathematical program (2.11). The model maximizes the flexibility while the WIP quantities of each recipe are distributed on the tools, and such that the WIP quantities are larger than or equal to 0.

	Tools			
Recipe	A	B	C	WIP quantities
1	-	25	25	50
2	200	-	200	400
3	**0**	225	225	450
WIP Balance	200	250	450	

TAB. 2.12 – An example where WIP quantities are not optimally balanced.

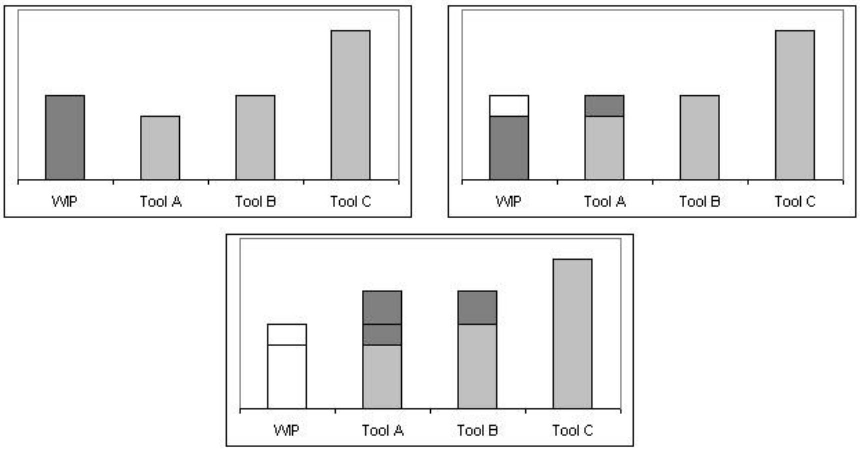


FIG. 2.1 – Illustrating the WIP balancing algorithm

$$
\begin{aligned}
&\max F^{WIP} \\
&\sum_{t=1:Q_{r,t}=1}^{T} WIP_{r,t} = WIP_r \quad r = 1,..,R \\
&WIP_{r,t} \geq 0 \qquad\qquad\qquad r = 1,..,R \quad t = 1,..,T
\end{aligned}
\tag{2.11}
$$

The algorithm is described below.

Step 0. For the initial solution S_0, the WIP quantities $WIP_{r,t}$ are distributed in a feasible but not necessarily optimal way, such that

$$\sum_{t=1:Q_{r,t}=1}^{T} WIP_{r,t} = WIP_r \quad r = 1, .., R$$

and

$$WIP_{r,t} \geq 0 \quad r = 1, .., R, t = 1, .., T$$

The recipe index is set to $r = 1$, and the solution index is set to $k = 1$.

Step 1. A tool t^* is defined as a *loading tool* t^* for recipe r if it is qualified, i.e. $Q_{r,t^*} = 1$, and has the smallest WIP quantity of all qualified tools for recipe r, i.e. :

$$WIP(t^*) \leq WIP(t) \quad \forall t \in \{t = 1, .., t^* - 1, t^* + 1, .., T\} \text{ such that } Q_{r,t} = 1$$

Several tools might be loading tools for recipe r, and the set of *loading tools* is denoted $\mathcal{T}_r^*$.

Step 2. Distribute the WIP quantity on the loading tools in $\mathcal{T}_r^*$ until :

Step 2(a). The WIP quantity, $WIP(t^*)$, on the loading tools in $\mathcal{T}_r^*$ is equal to the WIP quantity of a tool t' which is not a loading tool but is qualified for recipe r :

$$WIP(t^*) = WIP(t') \quad \forall t \in \{t = 1, .., T\} - \mathcal{T}_r^*$$

Tool t' is then added to the set of loading tools, i.e. $\mathcal{T}_r^* \equiv \mathcal{T}_r^* \cup \{t'\}$, and go to Step 2.

Step 2(b). The entire WIP quantity of the recipe, WIP_r, has been distributed on the loading tools in $\mathcal{T}_r^*$. Then $r = r + 1$, and go to Step 1.

Step 3. The flexibility of the previous solution S_{k-1} is compared with the new solution S_k.

Step 3(a). If the new value of the WIP flexibility measure is strictly larger than the old value, i.e. $F_k^{WIP} > F_{k-1}^{WIP}$, go to Step 1 with $k = k + 1$ and $r = 1$.

Step 3(b). Otherwise, if $F_k^{WIP} = F_{k-1}^{WIP}$, then the WIP quantities $WIP_{r,t}$ are optimally balanced and the algorithm stops.

It can be proved that the algorithm optimally balances the WIP quantities such that the WIP flexibility measure F^{WIP} is maximized. Let us first prove that the local distribution procedure of the WIP quantity WIP_r for a recipe r locally optimizes F^{WIP}. The definitions of local and global distributions are made.

Definition 2.1. *A local distribution is a distribution of the WIP quantities $WIP_{r,t}$ of one recipe r on all qualified tools $t = 1, .., T$ such that $Q_{r,t} = 1$.*

Definition 2.2. *A global distribution is a distribution of the WIP quantities $WIP_{r,t}$ of all recipes $r = 1, .., R$ on all qualified tools $t = 1, .., T$ such that $Q_{r,t} = 1$.*

Lemma 2.1. *If $WIP(t_1) < WIP(t_2)$ for two tools t_1 and t_2, the increase of the sum $\sum_{t=1}^{T} WIP(t)^\gamma$ will always be smaller when the WIP quantity on tool t_1 is increased than when the WIP quantity of tool t_2.*

Démonstration. If $WIP(t_1) < WIP(t_2)$, and if WIP_C denotes the WIP quantity to assign, the lemma is true if the following inequality holds :

$$(WIP(t_1) + WIP_C)^\gamma + WIP(t_2)^\gamma < WIP(t_1)^\gamma + (WIP(t_2) + WIP_C)^\gamma \quad (2.12)$$

To prove that this is true it is noted that the expression below is true for $\gamma > 1$ and $WIP \geq 0$ where the binomial coefficients are omitted (2.13).

$$\begin{aligned} (WIP(t_1) + WIP_C)^\gamma = WIP(t_1)^\gamma + \gamma WIP(t_1)^{\gamma-1} WIP_C + \\ ... + \gamma WIP(t_1) WIP_C^{\gamma-1} + WIP_C^\gamma \end{aligned} \quad (2.13)$$

Substituting this in (2.12) results in the following :

$$\begin{aligned} \gamma WIP(t_1)^{\gamma-1} WIP_C + ... + \gamma WIP(t_1) WIP_C^{\gamma-1} \\ < \gamma WIP(t_2)^{\gamma-1} WIP_C + ... + \gamma WIP(t_2) WIP_C^{\gamma-1} \end{aligned} \quad (2.14)$$

This is clearly the case since every term on the left hand side of (2.14) is strictly smaller than the corresponding term on the right hand side of (2.14).

□

Note that the sum $\sum_{t=1}^{T} WIP(t)^{\gamma}$ is equal to the inversed WIP flexibility measure, $1/F_{WIP}$, where the constant value is ignored. Hence, it means that the WIP flexibility measure will increase more if the WIP quantity of the current recipe is distributed on the tool with the smallest WIP quantity.

Let us make the following remark before proving that the optimal local distribution of the WIP quantity for a given recipe will be globally optimal when the WIP quantity is distributed on all tools.

Remark 2.1. *If the WIP quantity of a given recipe r is distributed so that the WIP flexibility measure F_{WIP} is optimized, changing the WIP distribution for another recipe r' will in general imply that the WIP distribution for recipe r is no longer optimal. This is true even if the other WIP distributions also globally increased F_{WIP}. However, if the WIP quantities are individually optimized for each recipe while considering the WIP quantities of the other recipes, the total WIP distribution will be optimized.*

Where local and global distributions are defined in Definitions 2.1 and 2.2.

Lemma 2.2. *A local distribution of the WIP quantity WIP_r for a given recipe r^* that increases the WIP flexibility measure F^{WIP} also globally increases F^{WIP}.*

Démonstration. For a local distribution of a recipe r^* on the qualified tools $t \in \{1,..,T\}$, i.e. such that $Q_{r^*,t} = 1$ which optimizes (2.11), the WIP quantities $WIP_{r^*,t}$ are variables. Furthermore the WIP quantities $WIP_{r,t}$ for all other recipes $r \in \{1,..,R\} - \{r^*\}$ are constant. For the global distribution, all WIP quantities $WIP_{r,t}$ are variables. If any of the WIP quantities $WIP_{r^*,t}$ is changed, it would be changed with as much for the global distribution without changing the values of the WIP quantities $WIP_{r,t}$ for any of the other recipes $r \in \{1,..,R\} - \{r^*\}$.

Hence a local distribution of WIP_{r^*}, which results in an increase of the WIP flexibility measure F^{WIP}, results in the same increase for F^{WIP} globally. □

Lemma 2.3. *If there is no recipe r such that redistributing its WIP quantity $WIP_{r,t}$ leads to an increase of the WIP flexibility measure F^{WIP}, then F^{WIP} is globally optimal.*

Démonstration. The only variable terms of F^{WIP} in (2.3) are the WIP quantities $WIP(t)$ on the tools. Globally $WIP(t)$ is defined as follows :

$$WIP(t) = \sum_{r=1}^{R} WIP_{r,t} \tag{2.15}$$

For the case when only the WIP quantity WIP_{r^*} of recipe r^* is distributed, the WIP quantity on tool t can be written $WIP^*(t) = WIP_{r^*,t} + \sum_{r=1;r\neq r^*}^{R} WIP_{r,t}$.

When the value of $WIP_{r^*,t}$ and the corresponding term in (2.15) are changed, the change is as large in $WIP(t)$ as in $WIP^*(t)$ and hence the local redistribution of $WIP_{r^*,t}$ increases F^{WIP} as much in both cases. □

2.4.2 Workload balancing for time flexibility F^{time}

In Section (2.4.1), a workload balancing algorithm that founds the optimal solution for F^{WIP} was presented. In this section, an algorithm is derived to optimally balance the production times on the tools to maximize F^{time}.

2.4.2.1 A minimax approach

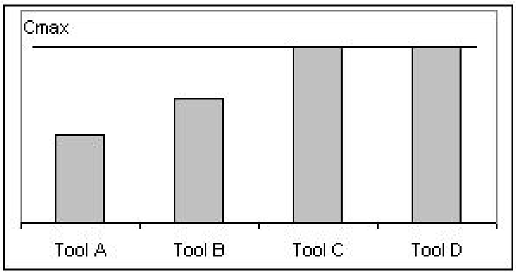
A method that first minimizes the maximal production time for the tools and then minimizes the total production time has been implemented by Rossi *et al.* [90, 92, 91]. The method, which is illustrated in Figures 2.2(a) and 2.2(b), is derived from an approach for scheduling on unrelated parallel processors [62]. The computations are conducted in two steps for the recipes for a toolset, with their corresponding throughput times :

1 Minimize the maximum production time (C_{max}) on the tools.

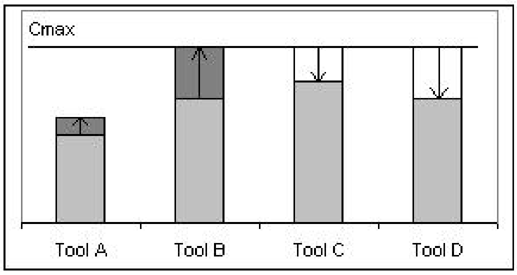
2 While keeping C_{max} fixed, the sum of z_j (the differences between the production time for each tool j and C_{max}) is maximized.

Example : A minimax approach

Although the method minimizes both the maximum production time of the tools and the total production time, two different solutions may be considered to be the same by the model. These two solutions do not in general give the same flexibility value



(a) The WIP quantities are first distributed such that the maximum production time C_{max} is minimized.



(b) In the second step, the sum of the production times is minimized while the maximum production time C_{max} is kept constant.

FIG. 2.2 – Time balancing algorithm by Rossi *et al.* [90, 92, 91]

and are not adequate for the time flexibility measure. To illustrate this point, two solutions from the previous algorithm with the production times below are considered. Since the maximum production time $C_{max} = 6h$ *and* $\sum(z_j) = 6h$ *are the same for*

	Tool A	Tool B	Tool C
Solution 1	6h	3h	3h
Solution 2	6h	5h	1h

both solutions, the algorithm considers them as equivalent. If, however, the new time flexibility measure is used, the values 0.89 (solution 1) and 0.77 (solution 2) are obtained. Considering the time flexibility measure F^{time}*, the solutions are different*

	Time flexibility
Solution 1 :	0.89
Solution 2 :	0.77

and hence the previous algorithm cannot be used. A time balancing algorithm that maximizes has to be derived. This can be done by using the active set method.

Also Aubry *et al.* [9] has developed a program which both minimizes and balanced the workload on the tools in a workshop as much as it is possible regarding the current qualifications. They do, however, only consider uniform tools, i.e. a tool has the same process time for all process types. This makes the approach limited for all cases, and therefore a method that minimizes and balances the workload for workshops with different throughput times for different process types has been developed.

2.4.2.2 Active set method

In order to find the optimal time flexibility measure F^{time}, an algorithm that optimally balances the WIP quantities on the tools has to be developed. An algorithm which maximizes F^{time} needs to both minimize the throughput time and balance the WIP quantities under the constraints. Of course, there will be a trade-off between these two criteria. With the value γ in the time flexibility expression, it is possible to control which should be most important to achieve : a minimized throughput or a well-balanced workload. The problem can be formulated as the nonlinear program (2.16) :

$$\begin{aligned}
&\max F^{time} = \frac{C_{ideal}}{\sum_{t=1}^{T}\left(\frac{\sum_{r=1}^{R} WIP_{r,t}}{TP_{r,t}}\right)^{\gamma}} \\
&\sum_{t=1;Q_{r,t}=1}^{T} WIP_{r,t} = WIP_r \qquad r = 1,..,R \\
&WIP_{r,t} \geq 0 \qquad r = 1,..,R \quad t = 1,..T
\end{aligned} \tag{2.16}$$

where $WIP_{r,t}$ is the decision variable. I.e. how should the recipes be optimally distributed in order to optimize F^{time}. $TP_{r,t}$ it the throughput time of recipe r on tool t expressed as processable wafers per hour.

In this section an implementation of the active set method [89] is derived. The method finds an optimal distribution of the WIP quantities on the tools such that the time flexibility measure is maximized for values of γ larger than 1.

The objective function : Time flexibility

The time flexibility measure defined in (2.7) is recalled below.

$$F^{time} = \frac{C_{ideal}}{\sum_{t=1}^{T} C(t)^{\gamma}} \tag{2.7}$$

Since the numerator C_{ideal} in F^{time} is constant, maximizing F^{time} is equivalent to maximizing $\frac{1}{\sum_{t=1}^{T} C(t)^{\gamma}}$ or minimizing $\sum_{t=1}^{T} C(t)^{\gamma}$.

Decomposing the problem

The problem (2.16) is decomposed by considering only one recipe at a time. Still it is needed to consider the process times of the tools from the other recipes. If F^{time} is maximized for all the qualified tools of this recipe, there can still be a lot of unqualified tools that have not been considered. Hence, F^{time} for the whole set of tools is not optimized. It will, however, be proved at the end of this section that, if F^{time} is maximized for the qualified tools of each recipe sequentially sufficiently many times, the optimal value of F^{time} will be found.

For the current recipe r^*, the production time $C(t, r^*)$ for tool t can be rewritten as :

$$C(t, r^*) = \left(\frac{WIP_{r^*,t}}{TP_{r^*,T}} + \bar{C}(t, r^*) \right) \tag{2.17}$$

where $\bar{C}(t, r^*)$ is the production time for the WIP quantities of the remaining recipes $r \notin r^*$ on tool t.

If the derivative of $C(t)^{\gamma}$ is calculated with respect to the distributed WIP quantities, the gradient-element for tool t is given in (2.18) and, if the derivative is calculated again with respect to the distributed WIP quantities, the diagonal element of the Hessian matrix will be as shown in (2.19).

$$\nabla \left(C(t)^{\gamma} \right)_t = \frac{\gamma}{TP_{r^*,T}} \left(\frac{WIP_{r^*,t}}{TP_{r^*,T}} + \bar{C}(t) \right)^{\gamma-1} \tag{2.18}$$

$$\nabla^2 \left(C(t)^{\gamma} \right)_{tt} = \frac{\gamma \cdot (\gamma - 1)}{TP_{r^*,T}} \left(\frac{WIP_{r^*,t}}{TP_{r^*,T}} + \bar{C}(t) \right)^{\gamma-2} \tag{2.19}$$

Note that both (2.18) and (2.19) are larger than 0 when $WIP_{r^*,t} \geq 0$ and $\gamma > 1$.

Constraints

Constraints (2.20) and (2.21) for the problem of finding the optimal flexibility measure F^{time} for recipe r^* are recalled from (2.16) :

$$\sum_{t=1}^{T} WIP_{r^*,t} = WIP_{r^*} \tag{2.20}$$

$$WIP_{r^*,t} \geq 0 \qquad t = 1, .., T \tag{2.21}$$

Constraint (2.20) requires that the whole WIP quantity of a recipe is distributed on all the tools, and Constraint (2.21) requires that the WIP quantity distributed on a tool has to be larger than or equal to 0. The components of the constraints can also be put into matrix form (2.22) (corresponding to Constraints (2.20)) and (2.23) (corresponding to Constraint (2.21)) where the matrices are defined in (2.24).

$$A_1 x = WIP_{r^*} \tag{2.22}$$

$$A_2 x \geq 0 \tag{2.23}$$

$$A_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \quad A_2^T = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}, \quad x = \begin{pmatrix} WIP_{r^*,1} \\ WIP_{r^*,2} \\ \vdots \\ WIP_{r^*,T} \end{pmatrix} \tag{2.24}$$

Active constraints

The active set method solves sub-problems of the main problem where some of the constraints from (2.21) are set equal to 0 instead of larger than or equal to 0 and regarded together with Constraint (2.20) in a subset called the active set (denoted W). The remaining constraints in (2.21) are called the inactive set. The reduced active set sub-problem containing only the active set constraints $\hat{A}$ can be written as in (2.25). Constraints from (2.21) can later be added to or removed from (2.25), but the first constraint (2.22) will always be a part of the reduced constraint matrix (2.25). The first constraint (2.22) will be put on the first row in $\hat{A}$.

$$\hat{A}x = \hat{b} \tag{2.25}$$

Example 2.3. *Considering an example with four qualified tools, and hence five constraints, where two are set to be inactive (*$WIP > 0$*). If the constraints for the second and fourth tools are the ones that are held active, the reduced matrix can be written as in* (2.26)*. This means that it is still possible to distribute WIP quantities on the second and fourth tools.*

$$\hat{A} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{2.26}$$

Optimality test

Suppose that there is a feasible solution from the original problem. In order to see if the solution is optimal, the optimality condition (2.27) needs to be fulfilled [80]. To do that, it is needed to calculate the null-space matrix, Z (2.28). In (2.28), B is defined as a sub-matrix of $\hat{A}$ such that it spans a square matrix and that it covers a maximal part of $\hat{A}$.

$$Z^T \nabla \left(T^\gamma\right)_t = 0 \tag{2.27}$$

$$Z = \begin{pmatrix} -B^{-1}N \\ I \end{pmatrix} \tag{2.28}$$

If the optimality condition (2.27) is fulfilled for a feasible solution x_k, the solution is an optimal solution for the reduced problem with the active set [80].

In order to know if the solution is also globally optimal, Lagrange multipliers [5] can be used. The Lagrange multipliers describe the characteristics of the solution regarding the constraints. Using the gradient of the objective function (2.18), the

Lagrange multipliers show if the value of the objective function will decrease or increase if the omitted constraints are added.

Lagrange multipliers are computed as in (2.29). In the expression, $\hat{A}_r^{-T}$ is the transpose of the reduced inverse matrix $\hat{A}_r^{-1}$. $\hat{A}_r^{-1}$ and $\hat{A}_r^{-T}$ are defined as in (2.30) and (2.31). If all Lagrange multipliers λ are positive, the value of the function $\sum_{t=1}^{T} C(t)^{\gamma}$ that is minimized can only increase by adding the constraints and hence the solution is optimal [80].

$$\lambda = \hat{A}_r^{-T} \nabla \left(T^{\gamma}\left(x_k\right)\right)_t \tag{2.29}$$

$$\hat{A}_r^{-1} = \begin{pmatrix} B^{-1} \\ 0 \end{pmatrix} \tag{2.30}$$

$$\hat{A}_r^{-T} = \left(\hat{A^{-r}}\right)^T = \begin{pmatrix} B^{-1} \\ 0 \end{pmatrix}^T = \begin{pmatrix} B^{-T} & 0 \end{pmatrix} \tag{2.31}$$

Example 2.4. *In Example 2.3, the three first columns of* $\hat{A}$ *from* (2.26) *can be chosen such that* B*, its inverse* B^{-1} *and the remaining column* N *are defined as in* (2.32)*. Lastly, in* Z*,* I *is the identity matrix. Knowing this,* Z *is calculated in* (2.33)*. The matrix achieves the value* $\hat{A}_r^{-1}$ *in* (2.34)*.*

$$B = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, B^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix}, N = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{2.32}$$

$$Z = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \begin{bmatrix} \text{active} \\ \text{pseudo-active} \\ \text{active} \\ \text{inactive} \end{bmatrix} \tag{2.33}$$

$$\hat{A}_r^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & -1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} \text{active} \\ \text{pseudo-active} \\ \text{active} \\ \text{inactive} \end{bmatrix} \tag{2.34}$$

Active, inactive and pseudo-active constraints

If the $\hat{A}$-matrix is further studied, it can be seen that, when the constraint i of A_1 has been removed, the column i in $\hat{A}$ only contains one "1" in the first row. This column is denoted as an *inactive* column of $\hat{A}$ since it corresponds to an inactive constraint. Furthermore a constraint on row i in A_1 which is kept active (the WIP quantity for the tool equals 0) corresponds to a column i in $\hat{A}$ which, except the "1" in the first row, also has another "1". Such a column in $\hat{A}$ is denoted *active*.

If all the active columns in $\hat{A}$ would be included in matrix B, one column would miss in order for B to be a square matrix. Thus one of the inactive columns in $\hat{A}$ is added to B. This column is denoted as *pseudo-active*. If the null-space matrix Z is studied, it can be seen that the rows in Z which correspond to the inactive rows in $\hat{A}$ set up the identity matrix I. Furthermore the pseudo-active row contains the value (or values) -1, and the active rows consist of zeros.

Similarly the $\hat{A}_r^{-r}$ matrix has inactive rows which contain zeros. The pseudo-active row has a "1" in the first column and the rest of the values are "-1". The rest of the rows - which are active - will set up an identity matrix which starts in the second column for the first active row ; The first column of the active constraints consists of zeros.

For the examples (2.33) and (2.34), the active, inactive and pseudo-active constraints are noted behind the matrices.

Dropping constraints

If one or more Lagrange multipliers λ are negative, the solution is not a minimum for $\sum_{t=1}^{T} C(t)^{\gamma}$. In this case, it is needed to drop one of the constraints corresponding to one of the negative multipliers. The constraint with the smallest negative

Lagrange multiplier is usually dropped [80].

Note that there are as many Lagrange multipliers as active and pseudo-active constraints. Since the pseudo-active constraint is actually an inactive constraint, the value of its multiplier is ignored.

If the Lagrange test shows that the solution is a minimum - i.e. all multipliers are larger than or equal to zero - the algorithm starts over with the WIP quantities of the next recipe. If, however, the solution was not a minimum, after dropping one of the constraints, a new solution needs to be searched. To find a new solution, a so called new search direction has to be determined.

Search direction p and step length α

If a solution is not found to be optimal, a new solution must be determined. There are many different strategies to find a new search direction p – i.e. redistribution of the WIP quantities – where an optimal distribution of the WIP quantities exists. A powerful search direction is given by using the reduced Newton method (2.35) [45].

$$p = -Z\left(Z^T\nabla^2\left(T\right)Z\right)^{-1}Z^T\nabla\left(T\right) \tag{2.35}$$

From the current WIP distribution x_k, a step in the Newton direction is taken. The step length in the search direction is given by the parameter α. If one of the inactive (or pseudo-active) constraints limits the step in the direction, it might be needed to limit the step size for that constraint. This step length α can be calculated as in (2.36), where a_i and b_i correspond to the i^{th} constraint.

Due to the structure of the problem, α can be simplified. The value of b_i is always 0 for the inactive constraints ($b_1 = WIP_R$ is always active). Furthermore, only the inactive constraint for each row is equal to 1. Hence the general expression for the step length (2.36) can be simplified as in (2.37).

$$\bar{\alpha} = \max\left\{\alpha : x + \alpha p \text{ is feasible}\right\} = \min\left\{\frac{a_i^T x - b_i}{-a_i^T p} : a_i^T p < 0\right\} \quad \forall i \notin W \tag{2.36}$$

$$\bar{\alpha} = \min\left\{\frac{WIP_i}{-p_i} : p_i < 0\right\} \quad \forall i \notin W \tag{2.37}$$

Wolfe condition

In reality the step should in general not always go as far as to the next constraint. In the original Newton method, the step length is set to $\alpha = 1$ if the constraint is not reached. The step length α can also be calculated with the Armijo condition [6] which, with help of a backtracking method, finds an optimal length for α. In our implementation, the Wolfe condition has been chosen instead [110]. It has the advantage of quickly finding an approximate step length. The idea is to find the optimal length, such that the derivate is close to 0 and such that the Wolfe condition (2.38) is fulfilled for a sufficiently small value : $0 \leq \eta < 1$.

$$\left|p^T \nabla T\left(x + \alpha p\right)\right| \leq \eta \left|p^T \nabla T(x)\right| \tag{2.38}$$

The direction p is a descent direction : $T(x) > T(x + \alpha p)$ for sufficiently small values of α. If $x + \alpha p$ is close to the optimal value x^* where $T'(x^*) = 0$, the derivate with respect to α : $T'_\alpha(x + \alpha p) = p^T \nabla T(x + \alpha p)$. The proof can be found in [110].

If α is sufficiently small : $T'_\alpha(x + \alpha p) < 0$. Somewhere close to the optimum x^* : $T'_\alpha(x + \alpha^* p) = 0$; and for $\alpha > \alpha^*$: $T'_\alpha(x + \alpha p) > 0$.

Hence the minimum value of $T(x + \alpha p)$ occurs at α^* or at $\bar{\alpha}$ if $T'_\alpha(x + \alpha p) > 0$. The latter case occurs when the minimum lies beyond one of the inactive constraints. In reality, it is not needed to search for the optimal α^*, it is sufficient to stop when the Wolfe condition (2.38) is fulfilled. This solution lies closer to the optimum than $\alpha = 0$, and since it is anyway needed to run the active set algorithm again, a solution which is sufficiently close to the optimum is accepted.

Stopping criteria

When a new distribution has been found, it is needed to test if it is optimal. Hence, the algorithm starts from the beginning by testing the new solution. The algorithm will stop in order to redistribute the WIP quantity for the next recipe. If the optimality test (2.27) is satisfied and the Lagrange multipliers (2.29) indicate that the optimal solution is a minimum (which means that F^{time} is maximized).

Active set algorithm

The algorithm is described below. This algorithm can be simplified into the pseudo-code (B.1) which is in Appendix (B).

Step 1. Start with a feasible solution.

Step 2. If the optimality test is satisfied then

Step 2(a). If all Lagrange multipliers are larger than or equal to zero. Then stop and continue at Step 1 for the next recipe.

Step 2(b). Else drop the constraint corresponding to the most negative value in λ.

Step 3. Calculate a search direction for a new solution.

Step 4. Calculate the maximal step length.

Step 5. For the search direction and the maximal step length, calculate the new solution.

Step 6. If the new step length is shorter than the full step, add the limiting constraint to the active set. If the step length equals 0, add all constraints corresponding to tools with no WIP quantities to the active set.

Step 7. With the new solution, go back to the optimality test in Step 2.

Optimality of the active set method

The active set method, using the Newton method and the Wolfe condition, minimizes the value of a nonlinear convex function (proved in [89, 45, 110, 80]). Since $grad(C(t)^\gamma) > 0$, $grad(C(t)^\gamma)_t > 0$ and $grad^2(C(t)^\gamma)_{tt} > 0$ for $WIP_{r^*,t} > 0$ and $\gamma > 1$, the optimal solution is found for recipe r^*. In order to show that the method also is globally optimal when the WIP quantities for the recipes have been redistributed sufficiently many times, the same proof as for the distribution method for the WIP flexibility is used. The following lemma will be used, where local distribution is defined in Definition 2.1 and global distribution in Definition 2.2.

Lemma 2.4. *A local distribution of the WIP quantity $WIP_{r^*,t}$ for a given recipe r^* that increases $F_{r^*}^{time}$ for recipe r^*, also globally increases F^{time} regarding all recipes r.*

Démonstration. The only variable terms of F^{time} in (2.7) are the production times $C(t)$ on the tools. Globally the production times are defined as follows :

$$C(t) = \sum_{r=1}^{R} \frac{WIP_{r,t}}{TP_{r,t}} \tag{2.39}$$

For the case when only the WIP quantity WIP_{r^*} of recipe r^* is distributed, the production time $C(t, r^*)$ of tool t can be written as (deducted from (2.17)) :

$$C(t, r^*) = \left(\frac{WIP_{r^*,t}}{TP_{r^*,t}} + \bar{C}(t, r^*) \right)$$

When the value of $WIP_{r^*,t}$ and the corresponding term in (2.39) are changed, the change is as large in $C(t)$ as in $C(t, r^*)$ and hence the local redistribution of $WIP_{r^*,t}$ increases F^{time} as much in both cases. □

Lemma 2.5. *If there is no recipe r such that redistributing its WIP quantity $WIP_{r^*,t}$ leads to an increase of the WIP flexibility measure, then the time flexibility measure F^{time} is globally optimal.*

Démonstration. If the time flexibility measure was not optimal, then there would be a recipe r^* such that redistributing its WIP quantity $WIP_{r^*,t}$ would increase F^{time}. □

2.4.3 Performance of the balancing algorithms

It has not been the intention to research which are the best performing methods for the balancing algorithms. Instead a two balancing algorithm have been derived, which find the correct the optimal values for F^{WIP} and F^{time}. Certainly improvements can be made on the performance especially for the balancing algorithm of F^{time}. For non-linear problems there are a lot of solutions methods that can be tested such as the subgradient method [97] or the interior-point method [59, 68]. A study on which method has the performance would have risked loosing the focus on the main theme of the thesis.

2.5 Numerical Experiments

Numerical experiments have been performed on data from toolsets where recipes with different WIP quantities are qualified. MS Excel has been used with a VBA program to implement the calculations. The tests have been performed on various workshops. Qualifications and throughput times are from various sites of STMicroelectronics. Although the data are real, the results should be seen as theoretical values without including setup times. The results should, however, indicate if qualifications can be used to improve performance. In Chapter 5, simulations with schedulers are performed, which describe more realistic situations.

2.5.1 Impact of qualifications

Tests have been performed, to see how the production times on the tools can be affected by conducting the qualifications that will increase the most the time flexibility measure F^{time}.

In Figures 2.3 and 2.4, it can be seen how the production times for a toolset with six tools, ten recipes and 1329 wafers in a photolithography workshop vary for each qualification. By performing only two qualifications, the maximum production time decreases by nearly 45%. The exact values can be found in Tables 2.13 and 2.14.

Number of qualifications	**0**	**1**	**2**	**3**	**4**	**5**
Maximal production time	6.73	5.74	3.65	3.04	2.76	2.75
Average production time	2.46	2.30	2.25	2.25	2.25	2.25

Tab. 2.13 – Production times in hours for Figure 2.3

The reason why the first qualification does not decrease the maximum production time as much as the second one can be seen in Figure 2.4. At the beginning, there are two tools with large production times, and two qualifications are needed to decrease the production times for both tools.

Tests have also been performed for an etch poly workshop with 10 tools, 20 recipes and 1350 wafers with qualifications which increase the WIP flexibility F^{WIP} as shown in Table 2.15 and for an implant workshop with seven tools, 274 recipes and

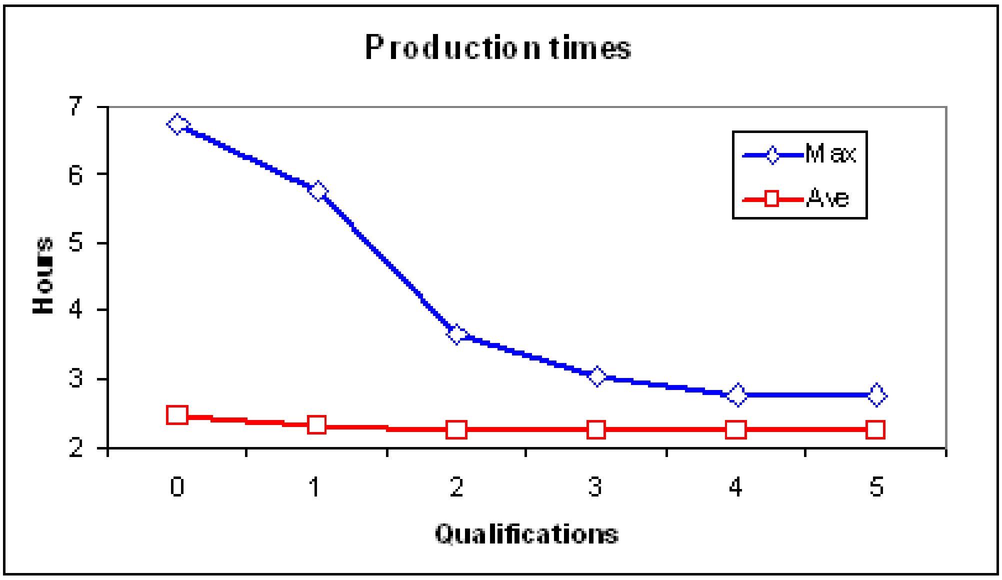


FIG. 2.3 – Production times after qualifications

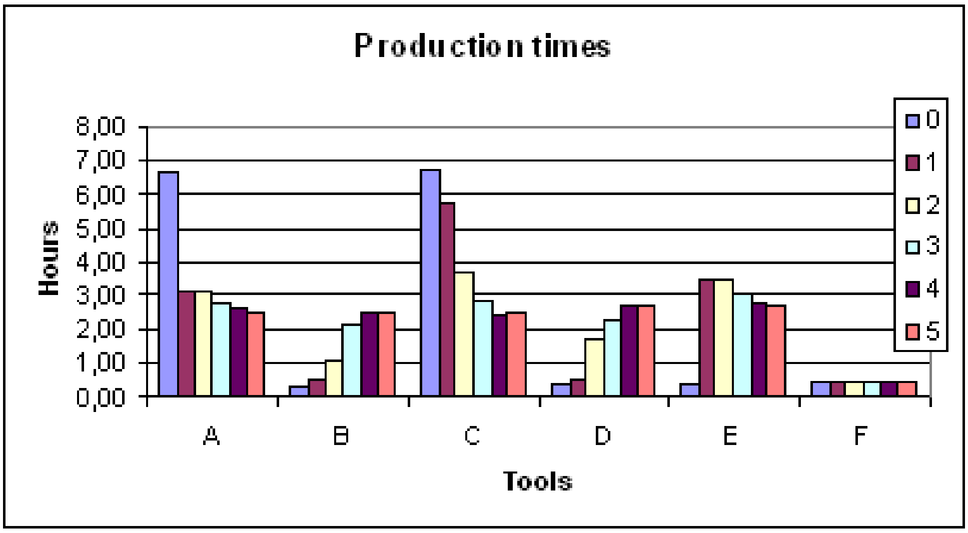


FIG. 2.4 – Production times per tool

11000 wafers with qualifications which increase the time flexibility F^{time} as shown in Table 2.16. For both series, the flexibility balance component γ has been set to

Number of qualifications	0	1	2	3	4	5
Tool A	6.63	3.15	3.15	2.78	2.61	2.52
Tool B	0.32	0.52	1.08	2.15	2.48	2.52
Tool C	6.73	5.74	3.65	2.83	2.43	2.52
Tool D	0.33	0.52	1.73	2.28	2.73	2.75
Tool E	0.34	3.44	3.44	3.04	2.76	2.75
Tool F	0.42	0.42	0.42	0.42	0.42	0.42

TAB. 2.14 – Production times in hours for Figure 2.4

be 4. The flexibility value are multiplied with 100 to show the *flexibility percentage* such as it is thought to be shown for the user version of the developed software.

Number of qualifications	0	1	2	3	4	5
$\mathbf{F^{WIP}}$ (%)	41.0	67.4	77.8	87.1	96.5	99.8
Max. WIP	346	217	217	164	152	146

TAB. 2.15 – Maximum WIP quantities in an etch poly workshop

Number of qualifications	0	1	2	3	4	5
$\mathbf{F^{time}}$ (%)	85.3	88.5	92.2	94.7	96.9	97.5
Max. time	13.8	13.8	13.4	13.2	13.2	13.1
Total time	85.2	85.4	85.0	84.7	83.9	83.9

TAB. 2.16 – Maximum production time and total production time in an implant workshop

The tests from the implant workshop are especially interesting since the throughput times show large differences between different tools for the same recipe. It also often occurs that a tool which is fast for one recipe is slow for another, whereas it is the opposite for another tool and the same recipes.

As observed in the results of Tables 2.15 and 2.16, the maximum workload can be reduced by performing the right qualifications. However, in some cases, no change is seen and, for the first qualification in implant, the total production time even increases. This is due to the fact that the flexibility balance exponent is set to a relatively high value ($\gamma = 4$) which, instead of minimizing the workload more, tries to better balance the workload on the tools. Therefore further studies on the effects of the flexibility balance exponent have been performed.

2.5.2 Impact of the flexibility balance exponent

As mentioned, with different values of the flexibility balance exponent γ, it is possible to regulate the solutions of the WIP flexibility measure and the time flexibility measure. For the WIP flexibility measure, well balanced WIP quantities still have higher value than WIP quantities that are not so well balanced. For the time flexibility measure, however, the flexibility balance exponent can be used to stress whether minimization of the total production time is more important in the measure than balancing production times on the tools or the opposite.

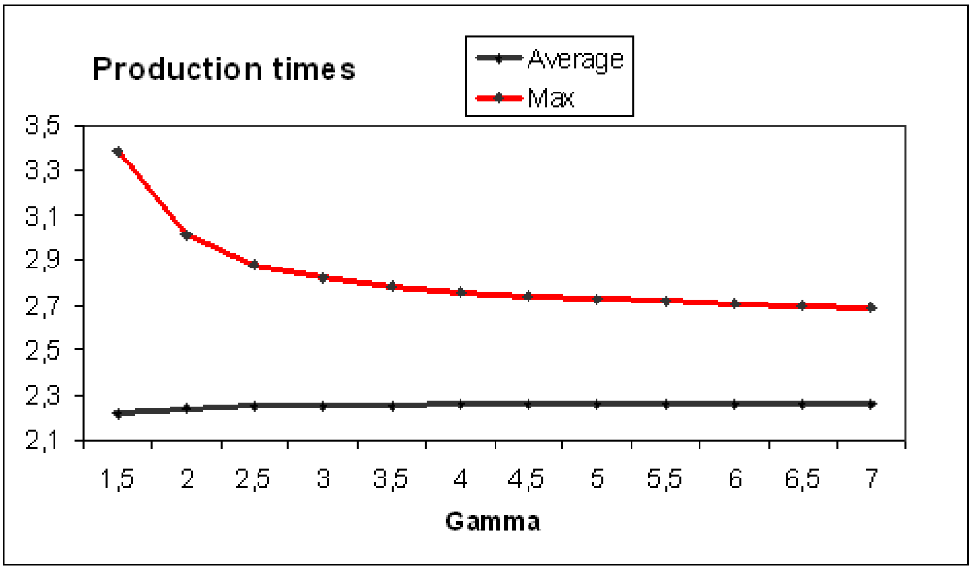


FIG. 2.5 – Production times for different values of γ

Using a small value of γ, the total production times – and hence also the average production times on the tools – are minimized but, at the same time, the maximum production time may remain large. Instead, by increasing γ, the production times on the tools become more and more balanced, and thus the maximum production time decreases at the same time as the average production time may slightly increase.

γ	**1.5**	**2**	**2.5**	**3**	**3.5**	**4**
Maximal production time (h)	3.39	3.01	2.88	2.82	2.78	2.76
Average production time (h)	2.22	2.24	2.25	2.25	2.26	2.26
Computing time (s)	0.06	0.14	0.24	0.40	0.56	0.68
γ	**4.5**	**5**	**5.5**	**6**	**6.5**	**7**
Maximal production time (h)	2.74	2.73	2.72	2.71	2.70	2.69
Average production times (h)	2.26	2.26	2.26	2.26	2.26	2.26
Computing time (s)	0.76	0.88	1.02	1.34	1.68	1.99

TAB. 2.17 – Production times in hours for Figure 2.5

In Figure 2.5 (Table 2.17 shows the exact values), it can be seen that the lines converges as γ increases. For $\gamma > 3$, only small changes can be observed. As it can be observed in Figure 2.6, the computing times of the time flexibility balancing algorithm also increases when γ increases, and it could be discussed whether it is necessary to use large values of γ.

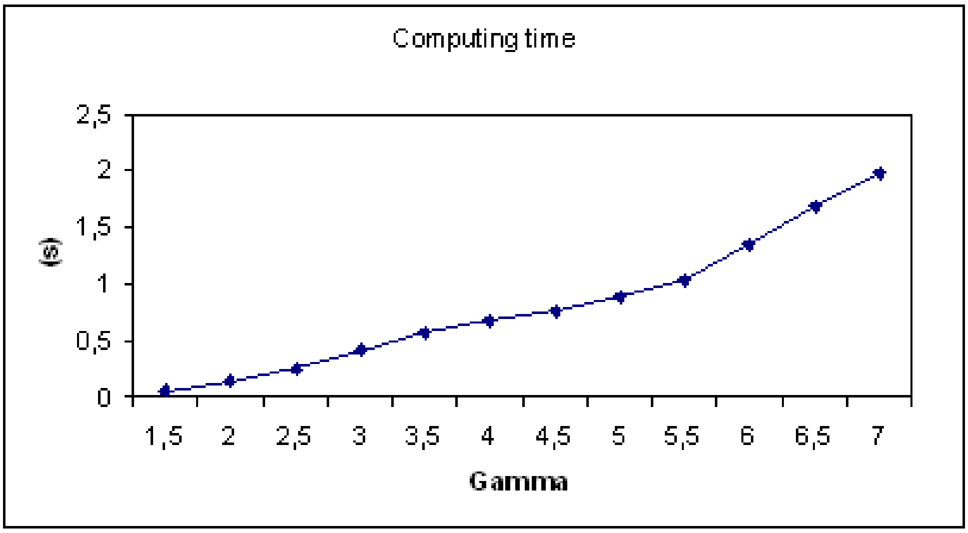


FIG. 2.6 – Computing times for different values of γ

2.5.3 Time flexibility versus WIP flexibility

To see how production times differ, an optimal solution for the time flexibility measure F^{time} ($\gamma = 4$) was compared with an optimal solution for the WIP flexibility measure F^{WIP}. The production times for the two solutions on six tools in a toolset are shown in Figure 2.7. The exact production times can be seen in Table 2.18. The total production times for the solutions are 13.44 hours (time flexibility) and 13.99 hours (WIP flexibility). The maximum production times are 3.01 (time flexibility) and 3.07 (WIP flexibility).

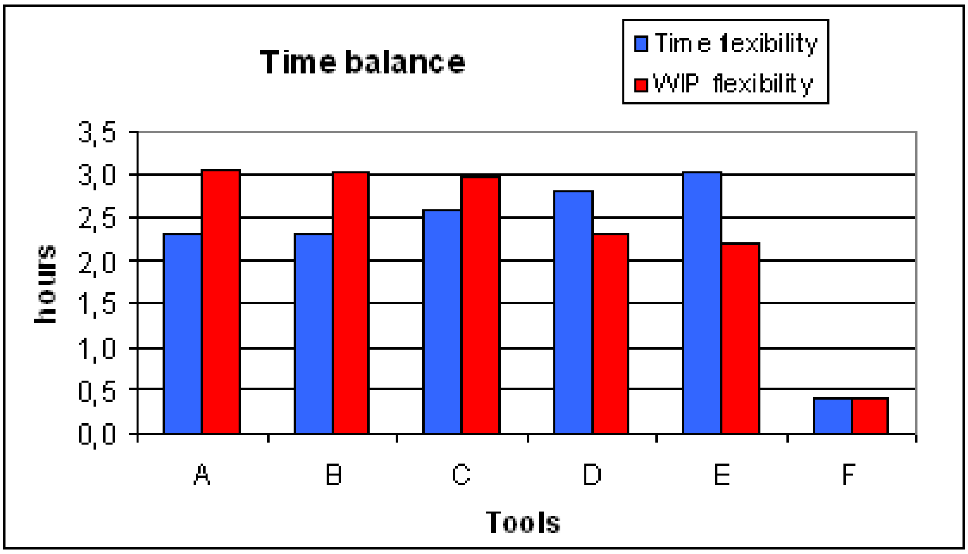


FIG. 2.7 – Production times on tools in a toolset

Production times (h)	F^{time}	F^{WIP}
Tool A	2.30	3.07
Tool B	2.33	3.02
Tool C	2.58	2.97
Tool D	2.81	2.31
Tool E	3.01	2.20
Tool F	0.42	0.42

TAB. 2.18 – Production times in hours for Figure 2.7

For the same solutions, the WIP quantities are distributed as in Figure 2.8 with exact WIP quantities in Table 2.19. The tool with the maximum WIP quantity is tool E with a WIP quantity of 356 for the time flexibility solution – this is a tool which generally has high throughput times. Since the WIP quantities are well balanced, the WIP quantity (259) is equally large on five of the six tools. Due to the fact that there are not so many qualified recipes on tool F, there is only a WIP quantity of 34 on this tool.

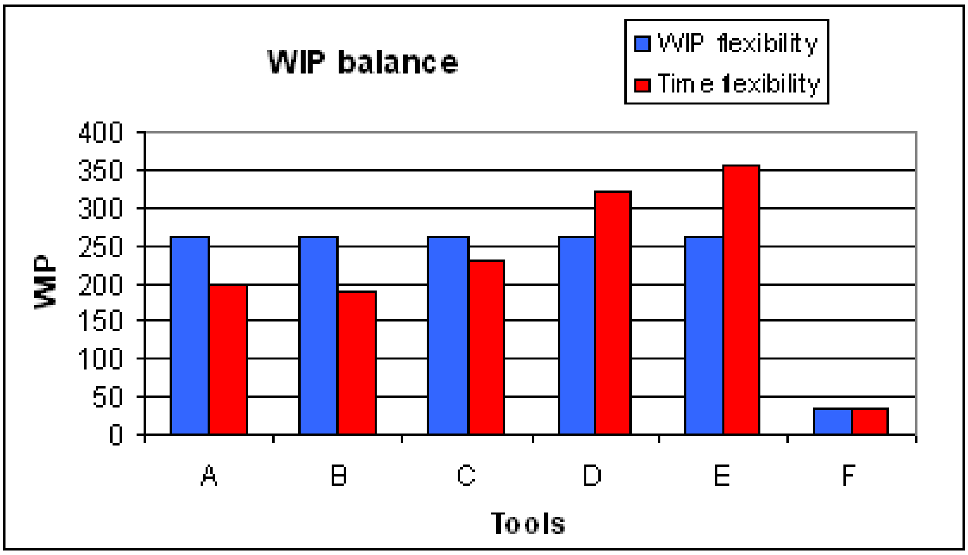


FIG. 2.8 – WIP quantities on tools in a toolset

WIP quantities	F^{WIP}	F^{time}
Tool A	259	199
Tool B	259	189
Tool C	259	229
Tool D	259	322
Tool E	259	356
Tool F	34	34

TAB. 2.19 – WIP quantities for Figure 2.8

2.6 Limitations of the Flexibility Measures

An often occurring phenomenon in reality is that there are no tools qualified for one or several recipes in the workshop. In these cases, it would most often be preferable to qualify a tool for any of the recipes with no qualified tools. The problem is how this should be considered by the flexibility measures. Therefore, the measures are slightly modified for the cases when there are recipes with no qualified tools, such that qualifying these recipes will be prioritized.

2.6.1 Limitations of the toolset flexibility measure F^{TS}

For the toolset flexibility, there is a mathematical problem when one of the recipes have no qualified tools, since there will be a division with 0 in (2.2). The way this is dealt with is that, for F^{TS} in (2.2), the term NQT_r achieves a value ϵ which is strictly larger than 0 – to avoid division with 0 – but smaller than 1 – to make sure that it is better to have one qualified tool than none.

Furthermore, it is required that the definition assures that, by qualifying a tool for a recipe which previously did not have any qualified units, F^{TS} always increases more than by qualifying a recipe which already has qualified tools even if this recipe has a very large WIP quantity. This can be seen in (2.40) where r is the recipe without qualified units, WIP_{max} the highest WIP quantity of a recipe and WIP_{rest} the WIP quantities of the remaining recipes. The term R represents the WIP quantities and qualified tools of the remaining recipes. In the left side of the expression, a tool has been qualified for the recipe with the highest WIP quantity and, on the right side, a tool has been qualified for recipe where previously no tools were qualified.

$$\begin{aligned} \frac{WIP_{rest} + WIP_{max} + WIP_r}{T\left(R + \frac{WIP_{max}}{2} + \frac{WIP_r}{\epsilon}\right)} &< \frac{WIP_{rest} + WIP_{max} + WIP_r}{T\left(R + \frac{WIP_{max}}{1} + \frac{WIP_r}{1}\right)} \\ \Leftrightarrow \quad WIP_{max} + WIP_r &< \frac{WIP_{max}}{2} + \frac{WIP_r}{\epsilon} \\ \Leftrightarrow \quad \epsilon &< \frac{2 \times WIP_r}{2 \times WIP_r + WIP_{max}} \end{aligned} \tag{2.40}$$

Hence, instead of 1, ϵ is constrained by (2.40). Additionally it is also needed that, if two recipes do not have any qualified units, the recipe with the highest WIP quantity should be qualified first. For example, if $WIP_{r1} > WIP_{r2}$, the constraint in (2.41) needs to be satisfied, which is the case.

$$\begin{aligned} \frac{WIP_{rest} + WIP_{r1} + WIP_{r2}}{T\left(R + \frac{WIP_{r1}}{\epsilon_1} + \frac{WIP_r 1}{1}\right)} &< \frac{WIP_{rest} + WIP_{r1} + WIP_{r2}}{T\left(R + \frac{WIP_{r1}}{1} + \frac{WIP_r 2}{\epsilon_2}\right)} \\ \Leftrightarrow \quad WIP_{r1} + \frac{WIP_r 2}{\epsilon_2} &< WIP_{r2} + \frac{WIP_r 1}{\epsilon_1} \\ \Leftrightarrow \quad \epsilon_2 &> \epsilon_1 \end{aligned} \tag{2.41}$$

Hence it is needed that $\epsilon_1 < \epsilon_2 < 2 \times (2WIP_r + WIP_{max})$. This can be achieved by defining ϵ as follows.

$$\epsilon = \frac{1}{WIP_r} \times \frac{WIP_r}{WIP_{max} + 2 \times WIP_r} \tag{2.42}$$

2.6.2 Limitations of the WIP flexibility measure F^{WIP}

For the WIP flexibility measure F^{WIP}, the problem is that the WIP quantities WIP_r^* for the recipe(s) r^* without qualified tools, i.e. such that $\sum_{t=1}^{T} Q_{r^*,t} = 0$, are not considered in the measure. Let Q_r denote the number of qualified tools for recipe r, i.e. $Q_r = \sum_{t=1}^{T} Q_{r,t}$.

The term $\sum_{r=1;Q_r=0}^{R} WIP_r$ is added to the denominator of F^{WIP} as shown in (2.43) below. This does not ensure that qualifying a recipe r with no qualified tool always increases more F^{WIP} than qualifying a recipe r' with at least one qualified tool, but it gives much more priority to qualifying r. Actually, it is possible to show that qualifying r is always better than qualifying r' if $WIP_r \geq WIP_{r'}$.

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} \left(WIP(t)\right)/T\right)^{\gamma}}{\sum_{t=1}^{T} \left(WIP(t) + \sum_{r=1;Q_r=0}^{R} WIP_r\right)^{\gamma}} \tag{2.43}$$

2.6.3 Limitations of the time flexibility measure F^{time}

For the time flexibility measure F^{time}, there are similar issues as with F^{WIP}. It is necessary to consider the WIP quantities WIP_{r^*} for recipes r^* with no qualified tools, i.e. such that $Q_{r^*} = \sum_{t=1}^{T} Q_{r^*,t} = 0$. A fictive worst case production time has been defined in (2.44) where $\min_{t=1,..,T}\{TP_{t,r}\}$ is the minimum throughput time for recipe r.

$$C_{r^*} = \sum_{r=1;Q_r=0}^{R} \frac{WIP_r}{\min_{t=1,..,T}\{TP_{t,r}\}} \tag{2.44}$$

The term C_{r^*} is then added to all the terms in the denominator of (2.7) as shown in (2.45) below. As for the modification for F^{WIP}, this does not ensure that qualifying a recipe r with no qualified tool always increases more F^{time} than qualifying a recipe r' with at least one qualified tool, but it gives much more priority to qualifying r.

$$F^{time} = \frac{C_{ideal}}{\sum_{\forall t}(C(t) + C_{r^*})^{\gamma}} \tag{2.45}$$

2.7 Conclusions

To the authors knowledge, no model has yet been developed which measures how qualifications may increase flexibility for capacity allocation. Therefore, in this chapter four flexibility measures were developed : toolset flexibility, WIP flexibility, time flexibility and system flexibility. Several examples showed how these measures work and which of the variants should be used.

In order to use two of the measures (WIP flexibility and time flexibility), the optimal balance of workload on the tools in a toolset needs to be found. Two different workload balancing methods have been developed. We have proved that the methods optimize the workload for the corresponding measure.

Finally numerical experiments were presented. They show that the qualifications based on the measures may reduce production times and balance workload in the fab. It has also been seen how, by adjusting the parameter γ for F^{time}, it is possible to decide which criteria has higher priority, well-balanced workload or minimized production times.

Chapitre 3

Optimizing Qualifications

The flexibility measures from Chapter 2 have been implemented in a software which is described in Appendix C. The software calculates the flexibility measures for a toolset with its current parameters (recipe on tool qualifications with associated WIP quantities). However, several aspects of qualifications still need to be considered in order to qualify recipes which will optimize the flexibility for the capacity allocation.

In this chapter it will be considered how more than two qualifications should be chosen in order to maximize the flexibility. It will be seen that finding the set of qualifications which maximize either F^{time} or F^{WIP} is a NP-hard problem. Different solving approaches will be proposed and tested.

3.1 Complexity

The problem of finding the additional k qualifications which maximize the flexibility is stated in (3.1) where WIP_r is the given WIP quantity of recipe r and $WIP_{r,t}$ the WIP quantity of recipe r assigned to tool t when solving the optimization problem. It is recalled from Chapter 2 that the parameter $Q_{r,t}$ is equal to 1 if recipe r has already been qualified on tool t, and 0 otherwise. The variable $OQ_{r,t}$ is defined for the additional qualifications that should be decided. $OQ_{r,t}$ is equal to 1 if recipe r should be qualified on tool t, and 0 otherwise. The maximum number of qualifications that can be made is k. The model $(OPT(F))$ of the problem is given below, where the objective function F is equal to the flexibility measure that has been selected, i.e. F is equal to F^{TS}, F^{WIP}, F^{time} or F^{SYS}.

$$(OPT(F)) \begin{cases} \max F & (a) \\ \sum_{t=1}^{T} WIP_{r,t} = WIP_r \quad r = 1,..,R & (b) \\ WIP_{r,t} \leq (Q_{r,t} + OQ_{r,t})\, WIP_r \quad r = 1,..,R \quad t = 1,..,T & (c) \\ \sum_{r=1}^{R} \sum_{t=1}^{T} OQ_{r,t} = k & (d) \\ OQ_{r,t} \in \{0,1\} \quad r = 1,..,R \quad t = 1,..,T & (e) \\ WIP_{r,t} \geq 0 \quad r = 1,..,R \quad t = 1,..,T & (f) \end{cases} \tag{3.1}$$

The objective (3.1.a) is to maximize the selected flexibility measure F. Constraint (3.1.b) ensures that the WIP quantity WIP_r of a recipe r is entirely distributed on the tools $t = 1,..,T$. Through Constraint (3.1.c), the WIP quantity WIP_r of a recipe r is only distributed on tool t if r is already qualified on t ($Q_{r,t} = 1$) or if r is proposed to be qualified on t ($OQ_{r,t} = 1$). Constraint (3.1.d) guarantees that the number of proposed qualifications ($OQ_{r,t} = 1$) is equal to k, the number of qualifications that can be made. Constraint (3.1.e) ensures that variables $OQ_{r,t}$ are binary, and Constraint (3.1.f)] that variables $WIP_{r,t}$ are positive.

For a toolset with T tools and R recipes, theoretically at most $N = R \times T$ qualifications are possible. To find which second qualification optimizes flexibility, $N-1$ possible qualifications need to be considered. All together there will be at most $N \times (N-1)$ combinations to search through, in order to find which two qualifications lead to the largest flexibility.

Calculating F^{time} for each qualification takes about 2 ms for the developed software on a desktop computer with an Intel Pentium 4 processor or a laptop with an AMD Turion 64x2 processor. With 100 possible qualifications in a toolset, calculating all possible combinations of two qualifications thus takes $100 \times 99 \times 2ms \approx 20s$. If further qualifications are considered, the calculation times grow exponentially when k grows. Checking all combinations of 5 qualifications would take 209 days (sic!).

Fortunately the problem is symmetric ; The combination of qualifications A and B results in the same flexibility as the combination of B and A. I.e. it does not matter which qualification is conducted first. Therefore the search can be performed more efficiently.

Finding the best combination of k qualifications out of N possible qualifications is a binomial problem with $\binom{N}{k}$ different combinations. To evaluate how the number of possible qualifications changes with k, Stirling's formula [93] can be used as in (3.2), where $\varepsilon_x \to 0$ as $x \to \infty$.

$$\binom{N}{k} = \frac{N!}{k!\,(N-k)!} \approx \frac{N^{N+1/2}}{\sqrt{2\pi}} \cdot \frac{(N-k)^{k-N-1/2}}{k^{k+1/2}} \cdot \frac{1+\varepsilon_N}{(1+\varepsilon_k)\,(1+\varepsilon_{N-k})} \tag{3.2}$$

From (3.2), it can be seen that the value of $\binom{N}{k}$ increases for $k < N/2$ as k grows. It can also be seen that the increase is exponential for $k < N/4$. This is typically the number of qualifications that are chosen in practice, and even when the number of qualifications k lies between $N/4$ and $N/2$, the number of possible combinations of qualifications is large. The exponential growth of $\binom{N}{k}$ results in that the problem of checking all combinations takes very long time.

Checking all combinations of five qualifications out of 100 possible qualifications with the binomial approach takes approximately two days. This is a considerable improvement from the first approach, but still not an acceptable computing time.

To see if it is really necessary to check all combinations in order to find the qualifications which optimize the flexibility measures, the complexity of the measures is studied.

3.1.1 Complexity of the flexibility measures

Consider a toolset where two optimal qualifications out of four possible should be found. The possible qualifications are named A, B, C and D. Suppose that the flexibility measure when A is selected, $F(A)$, is larger than the flexibility for any of the other qualifications, as shown in (3.3).

$$
\begin{aligned}
F(A) &> F(B) \\
F(A) &> F(C) \\
F(A) &> F(D)
\end{aligned} \tag{3.3}
$$

Moreover, suppose that, if A is selected, the largest improvement for the flexibility measure of an additional qualification would be to select B, i.e. (3.4) is valid.

$$
\begin{aligned}
F(A,B) &> F(A,C) \\
F(A,B) &> F(A,D)
\end{aligned} \tag{3.4}
$$

However, this does not implicitly imply that A and B would be the two qualifications which would optimize the flexibility measure. It could still be better to qualify C and D instead of A and B as indicated in (3.5).

$$F(C,D) > F(A,B) > F(A,C) \tag{3.5}$$

It could also be that selecting B and C is better than selecting A and B, and hence also better than A and C (instead of C, qualification D could be considered).

$$F(B,C) > F(A,B) > F(A,C) \tag{3.6}$$

If it could be proved that both (3.5) and (3.6) are false when (3.3) and (3.4) are true, it should be known that the combination AB is better than all other combinations of qualifications (AC, AD, BC, BD and CD), and hence, it would prove that it is not necessary to check all combinations of qualifications in order to find the optimal value of the flexibility measure, and thus an optimal solution could be found in polynomial time.

To investigate this, the complexity of optimizing for the various flexibility measures will be studied separately.

3.1.1.1 Complexity analysis for the toolset flexibility measure F^{TS}

Let us first introduce the following lemma.

Lemma 3.1. *A new qualification always increases F^{TS}.*

Démonstration. When a recipe r is qualified on a tool t, the term WIP_r/NQT_r decreases since $NQT_{r(new)} = NQT_{r(old)}+1$. This decreases the value of the denominator in the expression and thus the toolset flexibility measure (2.2) increases. □

The toolset flexibility measure F^{TS} in (2.2) is reformulated in (3.7) for an example with four recipes (A, B, C, D) where a qualification has been performed for recipe A such that the number of qualified tools NQT_A for recipe A becomes $NQT_A + 1$. The term k represents the constant term in the denominator of F^{TS}.

$$F^{TS}(A) = \frac{k}{\frac{WIP_A}{NQT_A+1} + \frac{WIP_B}{NQT_B} + \frac{WIP_C}{NQT_C} + \frac{WIP_D}{NQT_D} + c} \tag{3.7}$$

It is first checked if $F(C, D) > F(A, C)$ from (3.5) can be true. The expression is reformulated in (3.8) and simplified in (3.9). This cannot be true since it is stated in (3.3) that qualifying A is better for F^{TS} than qualifying D.

$$\frac{k}{\frac{WIP_C}{NQT_C+1} + \frac{WIP_D}{NQT_D+1} + \frac{WIP_A}{NQT_A} + \frac{WIP_B}{NQT_B} + c} < \frac{k}{\frac{WIP_A}{NQT_A+1} + \frac{WIP_C}{NQT_C+1} + \frac{WIP_B}{NQT_B} + \frac{WIP_D}{NQT_D} + c} \tag{3.8}$$

$$\frac{WIP_A}{NQT_A + 1} + \frac{WIP_D}{NQT_D} > \frac{WIP_D}{NQT_D + 1} + \frac{WIP_A}{NQT_A} \tag{3.9}$$

Similarly, it can be checked whether $F(B, C) > F(A, C)$, from (3.6), is true. This must, however, also be false since it was stated in (3.3) that qualifying A is better for F^{TS} than B.

Hence, it has been proved that finding the k qualifications that optimize F^{TS} can be done in linear time $O(k)$.

3.1.1.2 Complexity analysis for the WIP flexibility measure F^{WIP} and the time flexibility measure F^{time}

The WIP flexibility measure F^{WIP} and time flexibility measure F^{time} have more complex structures than the toolset flexibility measure F^{TS}. Whereas F^{TS} is additive – every qualification strictly increases the value of the toolset flexibility measure (proved in Section 3.1.1.1) – qualifying a recipe does not necessarily means that F^{WIP} or F^{time} increase. It is also possible to show through the example below that conducting the qualification that locally optimizes F^{WIP} or F^{time} for one qualification will not always be optimal for two or more qualifications.

Example 3.1. *Consider a workshop with seven tools (T_1, T_2, T_3, T_4, T_5, T_6, T_7) and five recipes (R_1, R_2, R_3, R_4, R_5) with WIP quantities and qualifications as shown in Table 3.1. The example is considered for optimizing F^{WIP} but can also be used for optimizing F^{time} since, if the throughput times are the same for all recipes on all tools, the optimal WIP distribution using the workload balancing algorithm in Section 2.4.1 will be optimal for F^{time}.*

	Tools							
Recipes	T_1	T_2	T_3	T_4	T_5	T_6	T_7	WIP quantities
R_1	X	-	-	-	-	-	X	112
R_2	-	-	-	X	X	X	-	180
R_3	X	-	-	-	X	-	-	258
R_4	-	-	X	-	-	-	-	262
R_5	X	-	-	-	-	X	-	235

TAB. 3.1 – Example of qualifications in a toolset.

When the WIP quantities are optimally distributed for $\gamma = 3$ as in Table 3.2, $F^{WIP} = 0.804$.

Qualifying recipe R_4 on tool T_2 increases $F^{WIP} = 0.979$. This is the single optimal qualification which optimizes the flexibility. The situation where recipe R_4 is qualified on tool T_2 is shown in Table 3.3.

The next qualification which optimizes F^{WIP} is obtained by qualifying recipe R_3 on tool T_7. This increases F^{WIP} to 0.995 (see Table 3.4).

	Tools							
Recipes	T_1	T_2	T_3	T_4	T_5	T_6	T_7	WIP quantities
R_1	0	-	-	-	-	-	112	112
R_2	-	-	-	167	0	13	-	180
R_3	91	-	-	-	167	-	-	258
R_4	-	-	262	-	-	-	-	262
R_5	76	-	-	-	-	154	-	235
Distribution	167	0	262	167	167	167	112	

TAB. 3.2 – Initial case. $F^{WIP} = 0.804$.

	Tools							
Recipes	T_1	T_2	T_3	T_4	T_5	T_6	T_7	WIP quantities
R_1	0	-	-	-	-	-	112	112
R_2	-	-	-	167	0	13	-	180
R_3	91	-	-	-	167	-	-	258
R_4	-	**131**	131	-	-	-	-	262
R_5	76	-	-	-	-	154	-	235
Distribution	167	131	131	167	167	167	112	

TAB. 3.3 – Recipe R_4 qualified on tool T_2. $F^{WIP} = 0.979$.

If the first qualification (recipe R_4 on tool T_2) is then removed to see if another qualification would optimize F^{WIP}, the same qualification is still chosen.

If (3.5) *and* (3.6) *would be true, the solution would be optimal (i.e.* (3.1) *with $k = 2$). However, the solution for two qualifications is not optimal. If, instead, recipe*

	Tools							
Recipes	T_1	T_2	T_3	T_4	T_5	T_6	T_7	WIP quantities
R_1	0	-	-	-	-	-	112	112
R_2	-	-	-	156	12	12	-	180
R_3	70	-	-	-	144	-	**44**	258
R_4	-	131	131	-	-	-	-	262
R_5	86	-	-	-	-	144	-	235
Distribution	156	131	131	156	156	156	156	

TAB. 3.4 – Recipe R_3 qualified on tool T_7. $F^{WIP} = 0.995$

R_3 is qualified on tool T_2 and Recipe R_4 on Tool T_7 (see Table 3.5), $F^{WIP} = 0.999$.

	Tools							
Recipes	T_1	T_2	T_3	T_4	T_5	T_6	T_7	WIP quantities
R_1	72	-	-	-	-	-	40	112
R_2	-	-	-	146	34	0	-	180
R_3	0	**146**	-	-	112	-	-	258
R_4	-	-	151	-	-	-	**111**	262
R_5	79	-	-	-	-	151	-	235
Distribution	151	146	151	146	146	151	151	

TAB. 3.5 – Recipe R_3 is qualified on tool t_2 and recipe R_4 on Tool T_7. $F^{WIP} = 0.999$

The example shows that (3.5) and (3.6) are not true for F^{WIP} (nor for F^{time}). Hence, a greedy heuristic (see Section 3.3.1) which always chooses the qualification which optimizes F^{WIP} for every single qualification cannot guarantee that an optimal solution for multiple qualifications is found.

In fact, it can be proved that the problem of optimizing F^{WIP} for k qualifications is NP-hard in the strong sense. To do that, the problem is reduced to the 3-partition problem which is known to be NP-complete in the strong sense [29]. A NP-hard problem is a problem which can be transformed to a NP-complete problem such that it cannot be solved in polynomial time (unless $P = NP$), which means that it is at least as difficult to solve as the NP-complete problem [29].

Theorem 3.1. *The problem of optimizing F^{WIP} for k qualifications is NP-hard in the strong sense.*

Démonstration. The proof is based on the proof in [9] for a problem with similar features : the Minimum-Cost Load-Balanced Configuration Problem (MCLBCP), which decision problem has the same complexity as the 3-partition problem.

The flexibility optimization problem will be stated as a decision problem and it will be shown that the 3-partition problem can be transformed to the flexibility decision problem and vice-versa.

The 3-partition problem considers, the sum of the "sizes" $s(a)$ such that

$$\sum_{a \in A} s(a) = mB$$

for $3m$ elements $a = a_1, a_2, ..., a_{3m}$ in a finite set A, where $s(a)$ lies between $B/2$ and $B/4$. The question is : can A be partitioned into m disjoint sets $S_1, S_2, ..., S_m$ such that $\sum_{a \in S_i} s(a) = B \quad \forall i \in [1, m]$? It can be noted that having $B/2 < s(a) < B/4$ and $\sum_{a \in S_i} s(a) = B \quad \forall i \in [1, m]$ imply that every subset S_i contains exactly three elements.

The 3-partition problem can be transformed to the flexibility problem (3.1) by defining all $3m$ elements a_i as $3m$ recipes where WIP_i of recipe i corresponds to the "size" $s(a_i)$ of a_i. The m sets can be seen as m different tools. Since the sets are disjoint, the tools are also disjoint, i.e. the WIP quantity WIP_i of recipe i must only be placed on exactly one tool j. Hence, with m tools and $3m$ recipes, $k = 3m$ qualifications are needed. This can be seen as a special case for the problem (3.1) of selecting $k = 3m$ qualifications which optimizes the flexibility.

Finally it is recalled that, in Section 2.3.2, it was stated that, in order to achieve $F^{WIP} = 1$, $WIP(j) = \sum_{t=1}^{m} WIP(t)/T$ for every tool j.

The question, transformed to a flexibility decision problem, is stated as follows : Is it possible with $3m$ qualifications to ensure that the quantities can be perfectly balanced such that $WIP(j) = B \quad \forall j \in [1, m]$ (i.e. $F^{WIP} = 1$) ? In order to see this, it is studied if the WIP quantity $WIP(j)$ of any tool j corresponds to the "size" balance of the 3-partition problem.

$$WIP(j) = \sum_{i=1}^{3m} WIP_{i,j} = \sum_{i=1} WIP_i \times Q_{i,j} = \sum_{i/a_1 \in S_j} WIP_i = \sum_{i/a_1 \in S_j} s_i = B \quad (3.10)$$

It has thus been shown that, if the 3-partition decision problem has an affirmative answer to its question, the flexibility decision problem also has an affirmative answer. Furthermore the opposite also needs to be proved : If the flexibility decision problem has an affirmative answer to its question, the 3-partition decision problem also has an affirmative answer.

It is first assumed that $k = 3m$ qualifications have been performed. Exactly one tool t is qualified per recipe i such that the whole WIP quantity of recipe i is placed on tool t. Furthermore it is assumed that the WIP quantities can be perfectly balanced on the tools such that $WIP(j) = B \in [1, m]$.

The disjoint sets $S_j \ \forall j$ include the "sizes" $s_i = WIP_i$ such that $Q_{i,j} = 1$ with three elements a_i in each set. The sum of the sizes in a disjoint set S_j can thus be written as follows :

$$\sum_{i/a_1 \in S_j} s_i = \sum_{i/a_1 \in S_j} WIP_i = \sum_{i=1} WIP_i \times Q_{i,j} = \sum_{i=1} 3mWIP_{i,j} = WIP(j) = B \quad (3.11)$$

Hence if it is possible to balance the WIP quantities perfectly on a tool with $k = 3m$ qualifications, it is also possible to partition the set A into m disjoint sets such that $\sum_{i/a_1 \in S_j} s_i = B$. $\square$

The fact that F^{WIP} is a non-additive function makes it even more difficult to solve the optimization problem (3.1). Recall that a non-additive function is a function which does not necessarily strictly increase when a qualification is added. Furthermore, it should also be noticed that the proof is also valid for the time flexibility measure F^{time} since, as already mentioned, the special case where all the throughput times are equal is equivalent to optimizing F^{WIP} with k qualifications. In general, however, the throughput times depend on tools and the problem is thus even harder.

Properties of the measures will be further studied in the next section. Solutions can be found by using the heuristics that are proposed in Section 3.3.

3.2 Properties of the Flexibility Measures

3.2.1 Properties of the toolset flexibility measure F^{TS}

The toolset flexibility measure (2.2) is recalled, in oder to study its properties.

$$F^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} \frac{WIP_r}{NQT_r}} \tag{2.2}$$

Lemma 3.2. *Any new qualification for a given recipe r is equivalent for F^{TS}, i.e. independent of the qualified tool t.*

Démonstration. Qualifying a tool t_1 or a tool t_2 for recipe r both decrease the value of the term WIP_r/NQT_r in the denominator of F^{TS} to the same value $WIP_r/(NQT_r + 1)$. □

Lemma 3.3. *Choosing recipe r such that :*

$$\arg_{r=1,..,R} \min \left\{ \frac{WIP_r}{NQT_r + 1} \right\} \tag{3.12}$$

optimizes F^{TS}.

Démonstration. The increase of F^{TS} for a qualification of recipe r can be calculated as $\Delta_{r:0\to1} F^{TS} = F_1^{TS} - F_0^{TS}$, where F_0^{TS} is the toolset flexibility before qualification and F_1^{TS} is the toolset flexibility after one qualification. In $\Delta_{r:0\to1} F^{TS} = F_1^{TS} - F_0^{TS}$, the term F_0^{TS} remains the same for qualifications of different recipes. Hence, only the qualification for the recipe r^* which leads to the largest value of F_1^{TS} needs to be found. The highest value of F_1^{TS} occurs for the qualification of recipe r which corresponds to the smallest term $WIP_r/(NQT_r + 1)$. □

Lemma 3.4. *Out of two consecutive qualifications for the same recipe r^*, F^{TS} increases more with the first qualification than the second.*

Démonstration. If Lemma 3.4 is true, the condition (3.13) needs to be fulfilled, where H is defined as follows :

$$H = \sum_{r=1;r\neq r^*}^{R} \frac{WIP_r}{NQT_r}$$

From the simplifications of (3.13) below, it can be seen that the condition is fulfilled.

$$\Delta_{r:0\to 1}F^{TS} > \Delta_{r:1\to 2}F^{TS} \tag{3.13}$$

$$\Leftrightarrow$$

$$\frac{\sum_{r=1}^{R} WIP_r}{T \times \left(\frac{WIP_{r^*}}{NQT_{r^*}+1} + H\right)} - \frac{\sum_{r=1}^{R} WIP_r}{T \times \left(\frac{WIP_{r^*}}{NQT_{r^*}} + H\right)} >$$
$$\frac{\sum_{r=1}^{R} WIP_r}{T \times \left(\frac{WIP_{r^*}}{NQT_{r^*}+2} + H\right)} - \frac{\sum_{r=1}^{R} WIP_r}{T \times \left(\frac{WIP_{r^*}}{NQT_{r^*}+1} + H\right)}$$

$$\Leftrightarrow$$

$$2 \times \left(\frac{WIP_{r^*}}{NQT_{r^*}+2} + H\right) >$$
$$\left(\frac{WIP_{r^*}}{NQT_{r^*}+2} + H\right)\left(\frac{WIP_{r^*}}{NQT_{r^*}+1} + H\right) + \left(\frac{WIP_{r^*}}{NQT_{r^*}+1} + H\right)\left(\frac{WIP_{r^*}}{NQT_{r^*}} + H\right)$$

$$\Leftrightarrow$$

$$2 \times \left(\frac{WIP_{r^*}}{NQT_{r^*}+2}\frac{WIP_{r^*}}{NQT_{r^*}} + \frac{WIP_{r^*}}{NQT_{r^*}+2} \cdot H + \frac{WIP_{r^*}}{NQT_{r^*}} \cdot H\right) >$$
$$\frac{WIP_{r^*}}{NQT_{r^*}+2}\frac{WIP_{r^*}}{NQT_{r^*}+1} + \frac{WIP_{r^*}}{NQT_{r^*}+1}\frac{WIP_{r^*}}{NQT_{r^*}} +$$
$$\frac{WIP_{r^*}}{NQT_{r^*}+2} \cdot H + \frac{WIP_{r^*}}{NQT_{r^*}+1} \cdot H + \frac{WIP_{r^*}}{NQT_{r^*}+1} \cdot H + \frac{WIP_{r^*}}{NQT_{r^*}} \cdot H$$

$$\Leftrightarrow$$

$$WIP_{r^*} \times (2NQT_{r^*} + 2 - 2NQT_{r^*} - 2) >$$
$$H \times (NQT_{r^*}(NQT_{r^*}+1) + (NQT_{r^*}+2)NQT_{r^*}) +$$
$$H \times ((NQT_{r^*}+2)NQT_{r^*} + (NQT_{r^*}+1)(NQT_{r^*}+2)) -$$
$$H \times (2(NQT_{r^*}+1)NQT_{r^*} + 2(NQT_{r^*}+1)(NQT_{r^*}+2))$$

$$\Leftrightarrow$$

$$0 > 2\left(NQT_{r^*}^2 + 2NQT_{r^*}\right) - NQT_{r^*}^2 + NQT_{r^*} - NQT_{r^*}^2 - 3NQT_{r^*} - 2$$

$$\Leftrightarrow$$

$$0 > -2$$

□

Analyzing the properties of the toolset flexibility measure

Finding the k qualifications which optimize F^{TS} can be done quite fast. Instead of calculating the flexibility for recipe qualifications on all tools, it is sufficient to calculate the toolset flexibility only for the recipes, since the toolset flexibility is the same for all recipes. It has been shown that the recipe r which has the highest value for the ratio $WIP_r/(NQT_r + 1)$ optimizes F^{TS}.

3.2.2 Properties of the WIP flexibility measure F^{WIP}

Predicting the behavior of the WIP flexibility measure F^{WIP} is less straightforward than for the toolset flexibility measure F^{TS}, since it depends on how the WIP quantities are spread on the tools in a toolset. Many properties of F^{WIP} can be stated. Let us start with the following definition.

Definition 3.1. *The total WIP quantity $\sum_{r=1}^{R} WIP_r$ is said to be* ***perfectly balanced*** *on the toolset when the total WIP quantity is equally distributed on all tools in the toolset, i.e.*

$$WIP(t) = \frac{\sum_{r=1}^{R} WIP_r}{T} \quad \forall t = 1, .., T.$$

Recalling the definition of F^{WIP} from (2.3), some properties can be derived.

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} WIP(t)/T\right)^{\gamma}}{\sum_{t=1}^{T} WIP(t)^{\gamma}} \in (0, 1] \qquad (2.3)$$

Lemma 3.5. *$F^{WIP} = 1$ if and only if the total WIP quantity $\sum_{r=1}^{R} WIP_r$ can be perfectly balanced on the toolset.*

Démonstration. By definition (see Section 2.3)

$$\sum_{r=1}^{R} WIP_r = \sum_{t=1}^{T} WIP(t)$$

and, if the total WIP quantity is perfectly balanced on all tools, then

$$WIP(t) = \sum_{t=1}^{T} WIP(t)/T \quad \forall t = 1, .., T$$

Hence, the value of F^{WIP} will be as in (3.14) :

$$F^{WIP} = \frac{T \times \left(\sum_{t=1}^{T} WIP(t)/T\right)^{\gamma}}{T \times \left(\sum_{t=1}^{T} WIP(t)/T\right)^{\gamma}} = 1 \tag{3.14}$$

Furthermore, $F^{WIP} = 1$ is only true if the value of the numerator in F^{WIP} is equal to the denominator. This can only happen when

$$WIP(t) = \sum_{t=1}^{T} WIP(t)/T \quad \forall t = 1, .., T$$

i.e. if the total WIP quantity is perfectly balanced.

□

Lemma 3.6. *A new qualification cannot strictly decrease the value of* F^{WIP}.

Démonstration. It was shown in Section 2.4.1 that the WIP balancing algorithm always finds the distribution of the WIP quantities which optimizes F^{WIP}. If the WIP quantities cannot be better distributed by an additional qualification, the balancing algorithm will still find the same distribution and F^{WIP} remains the same. However, if the qualification allows for a better distribution, F^{WIP} increases. □

Before describing the cases where F^{WIP} can be increased, further definitions are made.

Definition 3.2. *A* ***unique subset*** U_i *is a restricted subset of tools in a toolset such that the qualified recipes on the tools are only qualified on these tools and the tools are only qualified for these recipes.*

Definition 3.3. *The ratio* $\sum_{t \in U_i} WIP(t)/|U_i|$*, for a unique subset* U_i*, is called the* ***ideal WIP ratio****, where* $\sum_{t \in U_i} WIP(t)$ *is the WIP quantity that can be distributed on the tools of the unique subset and* $|U_i|$ *is the number of tools in* U_i*.*

Definition 3.4. *The* ***processable*** *WIP quantity* $\sum_{r=1;Q_{r,t}=1}^{R} WIP_r$ *for a tool* t *is the largest WIP quantity that can be processed on* t*.*

Definition 3.5. *A tool* $\underline{t}$ *is called a* ***least charged tool*** *if the following expression is true :*

$$\underline{t} = \arg \min_{t=1,..,T} WIP(t)$$

Definition 3.6. *A tool* $\underline{t}(U_i)$ *is called a* ***least charged tool*** *of a unique subset of tools* U_i *if the following expression is true :*

$$\underline{t}(U_i) = \arg \min_{t \in U_i} WIP(t)$$

Definition 3.7. *A tool* $\bar{t}$ *is called a* ***most charged tool*** *if the following expression is true :*

$$\bar{t} = \arg \max_{t=1,..,T} WIP(t)$$

Definition 3.8. *A tool* $\bar{t}(U_i)$ *is called a* ***most charged tool*** *of a unique subset of tools* U_i *if the following expression is true :*

$$\bar{t}(U_i) = \arg \max_{t \in U_i} WIP(t)$$

Remark 3.1. *An extinction is in this thesis made between what is commonly called a bottleneck tool and a most charged tool, where the bootleneck tool is the tool with the highest production time and most charged tool the highest WIP quantity.*

Example 3.2. *The example in Table 3.6 consists of four tools (A, B, C and D) and four recipes.*

	Tools				
Recipes	A	B	C	D	WIP quantities
1	-	X	-	-	10
2	X	X	-	-	30
3	-	-	X	X	30
4	-	-	X	-	40
Distribution	20	20	40	30	
Processable	20	40	70	30	

TAB. 3.6 – An example with two unique subsets.

Since tools A and B are only qualified for recipes 1 and 2 (that are only qualified on tools A and B), they represent a unique subset, that can be called U_1. Similarly tools C and D are only qualified for recipes 3 and 4 (that are only qualified on tools C and D), and thus represent another subset, called U_2. The ideal WIP ratios for U_1 is $\sum_{t \in U_1} WIP(t)/|U_1| = (10+30)/2 = 20$, and $\sum_{t \in U_2} WIP(t)/|U_2| = (30+40)/2 = 35$ for U2. In Table 3.6 it can be seen that $\sum_{t \in U_1} WIP(t)$ can be perfectly balanced within U_1, but that $\sum_{t \in U_2} WIP(t)$ cannot be perfectly balanced within U_2. Furthermore it can be seen that Tool C is the most charged tool of U_2 and the whole toolset. Tool D is the least charged tool in U_2, but not in the whole toolset since the tool charge is lower in U_1. Tools A and B are both the most charged tools and the least charged tools in U_1.

Lemma 3.7. *If $\sum_{t \in U_i} WIP(t)$ can be perfectly balanced on the tools in a unique subset U_i, the workload $WIP(t)$ of each tool $t \in U_i$ is equal to $\sum_{t \in U_i} WIP(t)/|U_i|$.*

Démonstration. If the total WIP quantity $\sum_{t \in U_i} WIP(t)$ of a unique subset U_i is perfectly balanced, the workload on all tools in U_i is equal to $\sum_{t \in U_i} WIP(t)/|U_i|$. This can only be obtained if the WIP quantity of each tool in U_i is equal to the total WIP quantity $\sum_{t \in U_i} WIP(t)$ divided by the number of tools $|U_i|$, which will be equal to $\sum_{t \in U_i} WIP(t)/|U_i|$. □

Lemma 3.8. *For a data set with two or more unique subsets, $F^{WIP} = 1$ is true only if the ideal WIP ratios $\sum_{t \in U_i} WIP(t)/|U_i|$ for each unique subset U_i are exactly the same and the WIP quantities can be perfectly balanced within all unique subsets.*

Démonstration. If the total WIP quantity of recipes associated to a unique subset cannot be perfectly balanced, the WIP quantities cannot be perfectly balanced on all tools. Therefore $F^{WIP} \neq 1$ according to Lemma 3.5. Furthermore, if $WIP(t_1) \neq WIP(t_2)$ where $t_1 \in U_1$ and $t_2 \in U_2$, the toolset is not perfectly balanced and, according to Lemma 3.5, $F^{WIP} \neq 1$. □

Lemma 3.9. *In order to perfectly balance the total WIP quantity within a unique subset U_i, each tool t must be able to process at least $\sum_{t \in U_i} WIP(t)/|U_i|$.*

Démonstration. If a tool t cannot process at least $\sum_{t \in U_i} WIP(t)/|U_i|$, then at least one of the other tools in the unique subset U_i processes more than $\sum_{t \in U_i} WIP(t)/|U_i|$ in order to satisfy (3.15). Hence, the workload cannot be perfectly balanced in U_i.

$$WIP(t) = \sum_{t \in U_i} \frac{WIP_t}{|U_i|} \quad \forall t \in U_i \tag{3.15}$$

□

Lemma 3.10. *Qualifying an additional recipe r on the least charged tool $\underline{t}(U_i)$ (See Definition 3.6) of a unique subset of tools U_i, always increases F^{WIP}.*

Démonstration. Qualifying a recipe r on a least charged tool $\underline{t}(U_i)$ such that

$$\underline{t}(U_i) = \arg_{t=1,..,T} \min \{WIP(t) \quad \forall t \in U_i\}$$

allows the WIP quantity WIP_r of a recipe r to still be distributed on the tools that was already qualified before the new qualification, but also to be distributed on $\underline{t}(U_i)$. Let t be a tool from where WIP quantities can only be redistributed to $\underline{t}(U_i)$ when the qualification of r on $\underline{t}(U_i)$ has been conducted. The redistributed WIP quantity is denoted by $\epsilon > 0$. It can be proved that (3.16) is true :

$$WIP(\underline{t}(U_i))^\gamma + WIP(t)^\gamma > (WIP(\underline{t}(U_i)) + \epsilon)^\gamma + (WIP(t) - \epsilon)^\gamma \tag{3.16}$$

The left hand side of the expression corresponds to the distribution of WIP quantities before the redistribution and, on the right hand side, after the redistribution.

If this is true, then the denominator of F^{WIP} decreases after the redistribution, and thus the WIP flexibility F^{WIP} increases. It can be shown that (3.16) is true since :

$$(WIP(\underline{t}(U_i)) + \epsilon)^\gamma = WIP(\underline{t}(U_i))^\gamma + WIP(\underline{t}(U_i))^{\gamma-1}\epsilon + ... + WIP(\underline{t}(U_i))\epsilon^{\gamma-1} + \epsilon^\gamma \tag{3.17}$$

and

$$(WIP(t) + \epsilon)^\gamma = WIP(t)^\gamma - WIP(t)^{\gamma-1}\epsilon + ... + WIP(t)(-\epsilon)^{\gamma-1}(-\epsilon)^\gamma \tag{3.18}$$

where the binomial coefficients are omitted. Since the negative terms on the right hand side (3.17) is larger than (3.17), (3.16) is true. □

Lemma 3.11. *Qualifying an additional recipe r on a most charged tool $\bar{t}$ (See Definition 3.7) never increases F^{WIP}.*

Démonstration. According to Definition 3.7, a most charged tool is a tool $\bar{t}$ such that

$$\bar{t} = \arg_{t=1,..,T} \max \{WIP(t) \quad \forall t = 1, .., T\}$$

As shown in Lemma 3.10 : qualifying a recipe r on a tool t allows the WIP quantity $WIP_{r,t}$ to be distributed on the tools where r was qualified before the new qualification $Q_{r,t}$. But qualifying an additional recipe on a most charged tool $\bar{t}$ only allows more WIP quantity to be transferred to that tool. Hence, the balancing algorithm in Section 2.4.1 finds the same distribution of the WIP quantities than before the qualification, and hence the value of F^{WIP} remains the same. □

Lemma 3.12. *By qualifying a recipe r on a tool t' for which WIP quantity WIP_r is distributed on a most charged tool $\bar{t}$, F^{WIP} increases if t is not a most charged tool.*

Démonstration. By qualifying a recipe r on a tool t' so that WIP_r is already distributed on $\bar{t}$, a WIP quantity ϵ ($\epsilon > 0$ and $\epsilon \leq WIP_r$) can be moved from $\bar{t}$ to t' such that :

$$WIP_{after}(t') = WIP_{before}(t') + \epsilon$$
$$WIP_{after}(\bar{t}) = WIP_{before}(\bar{t}) - \epsilon$$
$$WIP_{after}(t') \leq WIP_{after}(\bar{t})$$

WIP_{before} is the WIP quantity before the qualification and WIP_{after} after the qualification. Let us show that $F^{WIP}_{after} > F^{WIP}_{before}$. Using the conditions above and because $WIP_{after}(t) = WIP_{before}(t)$, $\forall t \in \{1,..,T\} - \{\bar{t}, t'\}$, $F^{WIP}_{after} > F^{WIP}_{before}$ becomes :

$$\frac{T \times \left(\sum_{t=1}^{T} (WIP_{after}(t))/T\right)^{\gamma}}{\sum_{t=1;t\neq t',t\neq \bar{t}}^{T} WIP_{before}(t)^{\gamma} + (WIP_{before}(t') + \epsilon)^{\gamma} + (WIP_{before}(\bar{t}) - \epsilon)^{\gamma}} > \frac{T \times \left(\sum_{t=1}^{T} (WIP_{before}(t))/T\right)^{\gamma}}{\sum_{t=1}^{T} WIP_{before}(t)^{\gamma}} \quad (3.19)$$

Because $\sum_{t=1}^{T} WIP_{after}(t)$ has a constant value ($\sum_{r=1}^{R} WIP_r$) and at the same time is equal to $\sum_{t=1}^{T} WIP_{before}(t)$, (3.19) can be simplified :

$$WIP_{before}(t')^{\gamma} + WIP_{before}(\bar{t})^{\gamma} > (WIP_{before}(t') + \epsilon)^{\gamma} + (WIP_{before}(\bar{t}) - \epsilon)^{\gamma} \quad (3.20)$$

And (3.20) is true because $WIP_{before}(t') < WIP_{before}(\bar{t})$ ($\bar{t}$ is a most charged tool and not t').

□

Analyzing the properties of the WIP flexibility measure F^{WIP}

In order to calculate F^{WIP} faster, the definitions of unique subset, processable WIP quantity, ideal WIP ratio can be implemented. These values are easy to calculate compared to F^{WIP}. In spite of the fact that they can reduce computational times, they cannot assure that the qualifications which optimize F^{WIP} are found.

3.2.3 Properties of the time flexibility measure F^{time}

Finding properties for the time flexibility measure F^{time} is much harder than for the WIP flexibility measure F^{WIP}, since the different throughput times make it hard to predict where to qualify. Moreover it can be chosen to either stress the importance of balancing the production times or minimizing the total production time. One property can still be stated.

Lemma 3.13. *A new qualification cannot strictly decrease the value of* F^{time}.

Démonstration. It was shown in Section 2.4.2 that the time balancing algorithm always finds the distribution of the WIP quantities which optimizes F^{time}. If the WIP quantities cannot be better distributed by an additional qualification, the balancing algorithm will still find the same distribution and F^{time} remains the same. □

In general it can be said that a qualification should be performed on a tool which is both fast and with a current small assigned WIP quantity. However, in many cases, these hypotheses are not satisfied simultaneously. A common case is that, when one tool is faster than the others for all recipes, tool engineers tend to qualify as many recipes as possible on this tool.

A way to use the information on throughput times could be to implement heuristics which do not consider all possible qualifications. For recipe qualifications on fast tools or with small WIP quantities, additional qualifications can be tested with a higher likelihood to lead to an optimal solution than other qualifications. Although this may not guarantee a better solution, computational times will decrease.

3.2.4 Properties of the system flexibility measure F^{SYS}

The system flexibility measure F^{SYS} depends on the other flexibility measures and hence its properties directly depend on the properties of the other measures. When one or more of the values a, b and c in the system flexibility is set to 0, the corresponding measures have no influence on the system flexibility.

Lemma 3.14. *A new qualification cannot strictly decrease the value of* F^{SYS}.

Démonstration. The system flexibility measure F^{SYS} is the addition of F^{TS}, F^{WIP} and F^{time}. Because F^{TS} always increases with a qualification (Lemma 3.1) and F^{WIP} and F^{time} cannot strictly decrease(Lemma 3.6 and Lemma 3.13), F^{SYS} cannot strictly decrease. □

3.3 Heuristics

A heuristic is an approximate method which does not guarantee to determine an optimal solution. For the problem of finding the combination of qualifications which maximizes the flexibility measures, heuristics which reduce the number of tested combinations have been implemented. These are expected to decrease the computation times drastically, but optimality cannot be guaranteed.

Two problems which are related to the problem of finding k qualifications that optimize the flexibility are the p-center [57] and p-median [58] problems. The p-center problem is a location problem where p optimal sites (for example fire departments) should be located within an area (for example a town), so that the maximum distance from a site to the clients in the area is minimized. In the closely related p-median problem, the sum of all distances between the sites and the clients is minimized. Both these problems have been proved to be NP-hard [67]. Therefore heuristics have been developed to solve the p-center by Mladenović *et al.* [69] and the p-median problem by Hansen and Mladenović [42].

In [69], a greedy heuristic and two local search heuristics were used as references in order to test the performance of a tabu search and a variable neighborhood search heuristics for the p-center problem. Similarly, in this section, a greedy heuristic and two local search heuristics will be developed in order to compare it with a tabu search approach for the problem of finding the combination of k qualifications which optimizes the flexibility measures.

3.3.1 Greedy heuristic

A greedy heuristic has been implemented to determine the combination of qualifications that optimizes the selected flexibility measure. At each step, the greedy algorithm chooses the single qualification which optimizes the selected flexibility measure.

Finding a combination of k qualifications can be done in $N + (N - 1) + (N - 2) + \cdots \approx k \times N$ times. Finding a combination of 5 qualifications out of 100 possible qualifications takes less than a second to calculate for F^{time}. The computing times are acceptable but the question is if the solutions are sufficiently close to an optimal solution. The quality and the computing times of the greedy algorithm will be evaluated together with the other heuristics later in this chapter (Section 3.3.5).

Pseudo-code for the greedy heuristic

The following pseudo-code describes how the greedy heuristic finds k qualifications.

For $j = 1$ to k
  Determine (r^*, t^*) that solves (3.1) optimally for $k = 1$.
  $Q_{r^*,t^*} = 1$
Next j

3.3.2 Local search heuristic 1

The first local search heuristic uses the solution determined by the greedy algorithm at each step. Each time the greedy heuristic finds a new solution, the suggested qualifications are removed one by one in order to see if better qualifications can be found. At first the greedy heuristic finds a solution with two qualifications. The first qualification is removed, and it is checked if any other qualification could improve the flexibility. If this is the case, the new qualification is chosen and the second qualification is removed in order to see if there is a better substitute for that qualification. This is repeated until the two qualifications remains the same. Then the greedy heuristic finds a third qualification. The local search heuristic is run again for these three qualifications. The method is continued until the desired number of qualifications k is reached.

Pseudo-code for local search heuristic 1

For the pseudo-code of local search heuristic 1, a vector $BestQual(j)$ has been defined which contains recipe r and tool t for the optimal qualification $Q_{r,t}$ decided in (3.1) at step j. The variable *oldFlex* keeps the value of the previous flexibility measure, to compare with the selected flexibility measure F^*.

```
For j = 1 to k
  Determine (r*, t*) that solves (3.1) optimally
  Q_{r*,t*} = 1
  BestQual(j) = (r*, t*)
  if j ≥ 2 Do
    i = 1
    oldFlex=0
    While F >oldFlex
      oldFlex=F*
      Q_{BestQual(i)} = 0
      Determine (r*, t*) that solves (3.1) optimally for k = 1
      Q_{r*,t*} = 1
      BestQual(j) = (r*, t*)
      If i < k then
        i = i + 1
      Else
        i = 1
      End if
    End While
  End if
Next j
```

3.3.3 Local search heuristic 2

Another local search heuristic has also been developed. In local search heuristic 2, the greedy heuristic finds k qualifications, before the last step of local search heuristic 1 is performed. Local search heuristic 2 removes one qualification at the time and tries to replace it with a better one. When no qualifications can be replaced by better ones, the heuristic is stopped.

Pseudo-code for local search heuristic 2

The following pseudo-code describes local search heuristic 2 using the same parameters as for local search heuristic 1.

Let the greedy heuristic in Section 3.3.1 find $BestQual(j) \quad \forall j$
$oldFlex = 0$
$j = 1$
While $F^* > oldFlex$
  $oldFlex = F^*$
  $Q_{BestQual}(j)) = 0$
  Determine (r^*, t^*) that solves (3.1) optimally for $k = 1$
  $Q_{r^*,t^*} = 1$
  $BestQual(j) = (r^*, t^*)$
  If $j < k$ then
    $j = j + 1$
  Else
    1. $j = 1$
  End if
End While

3.3.4 Tabu search

Mladenović *et al.* [69] developed a tabu search and a variable neighborhood search metaheuristics to solve the p-center problem. Thus the idea was given to derive a tabu search method for the problem of determining k optimal qualifications.

Tabu search is a local search algorithm that requires a *starting solution* and a *neighborhood structure*. Tabu search proceeds by transiting from solution to solution, using the transitions defined by the neighborhood structure, in a systematic way until some stopping criterion is met. Tabu search examines all of the neighbors of the current solution and selects the best admissible move (note that this might be a degrading one). An admissible move is a move which is not on the *tabu list*, where the tabu list is a list containing forbidden moves. For a general description of tabu search see *e.g.* Glover [32, 33] and Glover *et al.* [34].

Only the basic ideas of tabu search have been implemented, i.e. a list of tabu moves to prevent local search from cycling. The procedure can be extended (and most likely improved) by integrating some of the more complex aspects of tabu search, *e.g.* aspiration level criterion, long-term memory and strategic oscillation. However, the primary objective is to see if the tabu search method provides a powerful heuristic for the problem of determining k optimal qualifications.

The greedy heuristic has been used to generate a starting solution. For the obtained solution, neighborhood solutions are defined. Neighbors are defined as solutions where all qualifications except one is kept from the previous solution.

Two variants of the tabu search have been implemented. Both variants use a fixed number of iterations (50) and fixed length of the tabu list (20 items). The difference between the variants is the way the neighborhood search is performed. In *Tabu search 1*, all neighbors, except the ones in the tabu list, are checked. In *Tabu search 2*, only changes of qualifications to neighbors that are conducted on either the same tool or the same recipe as the previously deleted solution are checked.

Below, an overview of the proposed tabu search procedure for the problem of finding flexible qualification is given, using the elements detailed in the previous sections. *MaxIter* corresponds to the maximum number of iterations the tabu search is run.

- Find a starting solution using the greedy heuristic.
- Repeat the following until *MaxIter* iterations without improvement on the makespan have been performed.
  - Remove the oldest solution in the tabu list, and replace it with the new solution.
  - Search the neighborhood of the current solution using one of the two search methods previously explained to find the best non-tabu move.

– Restore the best solution and stop.

3.3.5 Numerical experiments

Several numerical experiments have been conducted in order to test the performance of the heuristics. First initial tests on real fab data have been performed, and thereafter further tests been performed on data where qualifications, WIP quantities and throughput times have been randomly generated. The test instances have been solved with the flexibility software presented in Appendix C, implemented with MS Excel VBA on a PC with a Pentium 4 processor.

Using a complete search, where the flexibility is calculated for all possible combinations of qualifications, it is possible to compare the solutions of the heuristics with the optimal solutions.

3.3.5.1 Tests on real fab data

For a photolithography workshop at the Crolles300 site, tests have been run in order to see if the heuristics find the qualifications which optimize F^{time} and how long time it takes for the heuristics to compute the solutions. The photolithography workshop for which the numerical experiments have been performed consists of six tools and ten recipes with WIP quantities varying between 1 and 300 wafers. The throughput times vary between 75 and 130 wafers per hour.

Table 3.7 shows how F^{time} evolves (in percentage) as additional qualifications proposed by the different heuristics are performed. Also the computing times for the heuristics are shown.

In the original set, one of the recipe was not qualified on any tool and hence F^{time} was close to 0%. After qualifying one tool for this recipe, F^{time} increased to 0.545 (or 54.5%). The greedy heuristic is faster than the other heuristics but does only find the optimal solution for the two qualifications. The two local search heuristics are both quite fast and find the optimal solution more cases than the greedy heuristic, when they find the same solutions as the greedy heursitc. For five qualifications Local Search Heuristic 1 finds the optimal solution which is not found by Local Search Heuristic 2. The computational time for tabu search is longer but

Number of qualifications		2	3	4	5	6	7
Complete search	(time)	4s	1min	9min	>1h	>7h	>38h
	(flexibility)	**74.0**	**81.3**	**84.5**	**88.7**	**91.7**	**92.7**
Greedy heuristic	(time)	0s	1s	1s	2s	3s	3s
	(flexibility)	**74.0**	80.2	83.8	85.6	90.5	91.1
Local search heuristic 1	(time)	1s	2s	3s	7s	8s	12s
	(flexibility)	**74.0**	**81.3**	83.8	**88.7**	**91.7**	**92.7**
Local search heuristic 2	(time)	1s	2s	2s	4s	5s	6s
	(flexibility)	**74.0**	**81.3**	83.8	85.6	**91.7**	**92.7**
Tabu search	(time)	1s	14s	20s	31s	65s	83s
	(flexibility)	**74.0**	**81.3**	**84.5**	87.8	**91.7**	**92.7**

Tab. 3.7 – Example : Evaluation of the heuristics in a photolithography workshop.

the optimal solution is found in all but one case. This case is also better than the greedy algorithm.

A similar test has been performed for an implant workshop. The test series for this workshop is much larger than for the photolithography workshop with 274 recipes and seven tools. Instead of directly considering the WIP quantities which are currently waiting in the workshop, the average values of moves per day and recipes have been considered. The implant area is also characterized by the fact that recipes can have different throughput times on different tools, and that a tool which produces one recipe fast, produces another recipe slow, whereas it is the opposite for another tool for the same recipes. It is hence interesting to use F^{time} to determine which qualifications should be performed. Since the complete search is too slow (already finding two qualifications takes about 16 hours), the complete search has been excluded. The solutions are shown in Table 3.8.

It can be seen that all heuristics find the same solution in all cases (without knowing for certain that it is the optimal solution). Additionally the computations take much longer times for these large instances, especially for the tabu search. Determining five qualifications with tabu search takes almost four hours.

Further work can be done to reduce the computational times. *E.g.* other heuristics could be developed, other ways of finding starting solutions could be studied and another balancing method than the active set method might be proposed.

Number of qualifications		2	3	4	5	6
Greedy heuristic	(time)	144s	250s	330s	472s	537s
	(flexibility)	**92.2**	**94.7**	**96.9**	**97.5**	**97.9**
Local search heuristic 1	(time)	312s	682s	1061s	1762s	2289s
	(flexibility)	**92.2**	**94.7**	**96.9**	**97.5**	**97.9**
Local search heuristic 2	(time)	311s	514s	640s	867s	988s
	(flexibility)	**92.2**	**94.7**	**96.9**	**97.5**	**97.9**
Tabu search	(time)	5506s	9355s	12132s	15322s	17830s
	(flexibility)	**92.2**	**94.7**	**96.9**	**97.5**	**97.9**

Tab. 3.8 – Example : Evaluation of the heuristics in an implant workshop.

3.3.5.2 Generation of test instances

Since it takes quite long time to download and prepare real fab data, test instances with qualifications in workshops have been run to achieve and calculate instances as fast as possible. The instances have been generated with MS Excel using VBA which is the environment used for prototypes at the participating company.

All in all, 1146 test instances with different qualification settings in workshops have been simulated. Workshops have been generated to have between 3 and 10 tools and 3 and 10 recipes. Most of these tests (424) have been generated for a workshop with 7 recipes and 7 tools. Initially, a recipe has been set to be qualified on a tool with 25% probability in most of the instances (1018 cases) and with 35% probability in some case (128 cases). The recipes have been set to have WIP quantities between 50 and 100 wafers, and the throughput times have been set to lie between 50 and 100 wafers per hour. This are typical values in many of the workshops in wafer fabs, e.g. photolithography.

Solving the test instances

It is checked how often the heuristics find a set of qualifications which result in the optimal value of F^{WIP}. Moreover the optimal frequency in percentage is noted. The average deviation in percentage from the optimal solution and the worse case are provided. At last the computational times are also given.

The tests are presented separately for the WIP flexibility measure F^{WIP} and the time flexibility measure F^{time}.

Testing heuristics for the WIP flexibility measure F^{WIP}

For F^{WIP} the heuristics have been tested for two to seven qualifications. The results from 280 instances for a workshop with seven tools and seven recipes are displayed in Table 3.9. First a complete search has been performed, which is guaranted to find the optimal solution. Afterwards the heuristics are run on the same problem, to see if they also find the optimal solution.

It can be seen that the greedy heuristic finds the optimal solutions for most instances where only two or three qualifications need to be found. For three to five qualifications, the greedy heuristic does not find the optimal solutions as often. Although after six and seven qualifications it seems that it starts to determine more optimal solutions again. This mainly has to do with the fact that, after six or seven qualifications, quite often a solution with $F^{WIP} = 1$ can be found.

The local search heuristics find the optimal solution more often. Local search heuristic 1 finds the optimal solution a little more often than local search heuristic 2 for five qualifications. The results of both methods are the very close.

The tabu search heuristics find the optimal solutions in almost all cases, and both variants find the optimal solutions with the same frequency.

In Table 3.10, a qualitative estimation of the heuristics is presented. For all cases when the optimal solution has not been found, the average deviation has been calculated. The maximum deviation (worst case) is also displayed.

The obtained solutions for the greedy heuristic are sometimes very far from the optimal solutions, especially for three, four and five qualifications. In one case for four qualifications, the solution is 49% away from the optimal solution, which should not be regarded as acceptable solutions.

The worst case solutions for the local search heuristics are not as bad as for the greedy algorithm. Only for three qualifications, the obtained solutions are more than 6 % lower than the optimal solution, which could be considered as a bad solution.

For the tabu search methods, where only one non-optimal solution for each method was found, the solutions are very close to an optimal solutions.

Number of qualifications		2	3	4	5	6	7
Method	Number of instances :	74	47	47	47	54	11
Greedy heuristic							
	Optimal solutions	**74**	37	38	41	52	**11**
	Optimality frequency (%)	100	79	81	67	96	100
Local search heuristic 1							
	Optimal solutions	**74**	45	46	44	54	**11**
	Optimality frequency (%)	100	96	98	94	100	100
Local search heuristic 2							
	Optimal solutions	**74**	45	46	43	**54**	**11**
	Optimality frequency (%)	100	96	98	91	100	100
Tabu search 1							
	Optimal solutions	**74**	**47**	**47**	**46**	**54**	**11**
	Optimality frequency (%)	100	100	100	98	100	100
Tabu search 2							
	Optimal solutions	**74**	**47**	**47**	**46**	**54**	**11**
	Optimality frequency (%)	100	100	100	98	100	100

TAB. 3.9 – Frequency of optimality for F^{WIP}

In Table 3.11, the average of the computational times for the heuristics are shown. Also the computational times for the complete search are displayed, and it can be seen how the computing times increase exponentially as the number of qualifications grow until the search of a set of seven qualifications. A stop function was implemented in the program, so that the search stops when $F^{WIP} = 1$. This happens quite often for sets of seven qualifications, and hence the methods do not need so much time in these cases.

The average computational times for the greedy heuristic stay under half a second for all cases. The two local search heuristics are also considered to be very fast. The computational times are below two seconds for local search heuristic 1 and below one second for local search heuristic 2.

The tabu search methods take longer time, both the computational times do not grow too much. The longest average computational times occur for both methods for six qualifications : 21 seconds for Tabu search 1 and 8.4 seconds for Tabu search 2.

Number of qualifications		2	3	4	5	6	7
Method	Number of instances :	74	47	47	47	54	11
Greedy heuristic							
	Average deviation (%)	0	8.3	11	5.5	0.5	0
	Worst case (%)	0	21	49	23	0.5	0
Local search heuristic 1							
	Average deviation (%)	0	6.1	0.5	0.7	0	0
	Worst case (%)	0	6.5	0.5	1.0	0	0
Local search heuristic 2							
	Average deviation (%)	0	6.1	0.5	0.7	0	0
	Worst case (%)	0	6.5	0.5	1.0	0	0
Tabu search 1							
	Average deviation (%)	0	0	0	0.3	0	0
	Worst case (%)	0	0	0	0.3	0	0
Tabu search 2							
	Average deviation (%)	0	0	0	0.2	0	0
	Worst case (%)	0	0	0	0.2	0	0

TAB. 3.10 – Optimality deviation for F^{WIP}

Testing heuristics for the time flexibility F^{time}

A similar series of tests with 144 instances (seven tools and seven recipes) have been performed for optimizing F^{time}. Since the computational times are much longer for F^{time} than for F^{WIP} it was decided to only calculate the solutions for up to five qualifications.

Table 3.12 shows how often the heuristics find the optimal solutions. The optimal solutions for F^{time} are not found as often as for F^{WIP}. For F^{WIP}, many different qualifications may lead to an optimal solution, which is not the case for F^{time}. This can be seen on the results for the greedy heuristic, which only finds the optimal solutions in 50% of the cases for five qualifications. For F^{time}, it is also possible to see a clear difference between the two local search heuristics. Local search heuristic 1 finds the optimal solutions for five qualifications in all cases, whereas local search heuristic 2 only finds the optimal solution in 81 % of the cases. The tabu search heuristics still find the optimal solutions in most cases.

In Table 3.13 it can be seen how the non-optimal solutions deviate from the

Number of qualifications		2	3	4	5	6	7
Method	Number of instances :	74	47	47	47	54	11
Complete search							
Computational time (s)		0.9	17	151	1162	7050	4015
Greedy heuristic							
Computational time (s)		0.1	0.2	0.5	0.4	0.3	0.4
Local search heuristic 1							
Computational time (s)		0.3	0.5	1.4	1.4	1.9	1
Local search heuristic 2							
Computational time (s)		0.2	0.5	0.8	0.9	0.6	0.5
Tabu search 1							
Computational time (s)		3.8	8.3	13	17	21	10
Tabu search 2							
Computational time (s)		1.4	3.0	5.2	6.3	8.4	4.1

TAB. 3.11 – Average computational times for F^{WIP}

optimal solutions. The non-optimal solutions for the greedy heuristic are on average quite far away from the optimal solutions. For three qualifications, the non-optimal solutions are on average 16% lower than the optimal solutions. The worst solutions for the greedy heuristic are also quite bad, but can be improved by the local search heuristics. Only for one case does local search heuristic 2 find a solution which is 12% lower than the optimal solution. For most other cases, the local search heuristics find solutions which are less than 5% away from the optimal solution. However, compared to the solutions for F^{WIP}, these results are worse. The two tabu search methods find the optimal solution in all cases except one, when the solution value is 4 % less than the optimal one .

As already mentioned, calculating F^{time} takes longer time than F^{WIP}. In Table 3.14, the computational times are displayed. Particularly for the tabu search, the computational times are long. Optimization for four and five qualifications with tabu search 1 takes more than one and half minutes. The computational times for the greedy and local search heuristics are, however, still small (a couple of seconds at most).

Number of qualifications		2	3	4	5
Method	Number of instances :	47	47	34	16
Greedy heuristic					
	Optimal solutions	46	44	26	8
	Optimality frequency (%)	98	94	76	50
Local search heuristic 1					
	Optimal solutions	46	**47**	32	**16**
	Optimality frequency (%)	98	100	94	100
Local search heuristic 2					
	Optimal solutions	46	**47**	32	13
	Optimality frequency (%)	98	100	94	81
Tabu search 1					
	Optimal solutions	**47**	**47**	33	**16**
	Optimality frequency (%)	100	100	97	100
Tabu search 2					
	Optimal solutions	**47**	**47**	33	**16**
	Optimality frequency (%)	100	100	97	100

TAB. 3.12 – Frequency of optimality for F^{time}

3.3.5.3 Test summary

Considering the 1146 instances, optimal solutions were found in 78% of the cases for the greedy heuristic, 91% of the cases for the first local search heuristic, 90% of the cases for the second local search heuristic, 99.8% of the cases for the first tabu search heuristic and 99.2% for the second tabu search heuristic.

If the fab engineers want to prioritize fast solutions probably one of the local search heuristics should still be used since there solutions value are quite much better than for the greedy heuristic. If the fab engineers prefer which more probably are optimal but can wait a little longer for a solution the tabu search should probably be used. It can be seen that the heuristics are slightly better for the generated tests instances, than for the test gotten from real instances. The instances are generated in a quite realistic way, but their results should not totaly be trusted. On the other hand the real case tests are few and does not show so significant results.

Number of qualifications		2	3	4	5
Method	Number of instances :	47	47	34	16
Greedy heuristic					
Average deviation (%)		2.0	16	8.0	7.2
Worst case (%)		2.0	19	11	17
Local search heuristic 1					
Average deviation (%)		2.0	0	3.4	0
Worst case (%)		2.0	0	4.5	0
Local search heuristic 2					
Average deviation (%)		2.0	0	3.4	7.3
Worst case (%)		2.0	0	4.5	12
Tabu search 1					
Average deviation (%)		0	0	3.9	0
Worst case (%)		0	0	3.9	0
Tabu search 2					
Average deviation (%)		0	0	3.9	0
Worst case (%)		0	0	3.9	0

TAB. 3.13 – Optimality deviation for F^{time}

3.4 Conclusions

In this chapter, determining qualifications which optimize the values of the flexibility measures have been studied. Properties of the flexibility measures have also been studied in order to find means to reduce computing times. It has been shown that finding the combination of qualifications which optimizes the toolset flexibility measure can be quickly conducted for any number of possible qualifications. Moreover it has been shown that optimizing F^{WIP} and F^{time} are NP-hard problems in the strong sense, which makes the problem difficult to solve. Therefore heuristics have been developed in order to find close to optimal solutions in reasonable computing times. It has been seen that tabu search heuristics can find optimal solutions in most cases.

The greedy heuristic finds optimal solutions for two qualifications for most cases. Results are worse for more qualifications but, at least for F^{WIP}, it seems like it starts to find optimal solutions more often as six or seven solutions are determined. The most important reason for using the greedy heuristic is that the computing times

Number of qualifications		2	3	4	5
Method	Number of instances :	47	47	34	16
Complete search					
Computational time (s)		7.3	90	1115	10360
Greedy heuristic					
Computational time (s)		0.7	0.7	1.4	1.7
Local search heuristic 1					
Computational time (s)		1.5	2.5	6.1	8
Local search heuristic 2					
Computational time (s)		1.4	1.8	4.0	4.4
Tabu search 1					
Computational time (s)		29	45	99	107
Tabu search 2					
Computational time (s)		7.6	6.0	28	35

TAB. 3.14 – Average computational times for F^{time}

are much smaller than for the other heuristics.

Both local search heuristics find optimal solutions more often, while the computing times stay quite small. Local search heuristic 2 is a little bit faster than local search heuristic 1. However, local search heuristic 1 finds an optimal solution one more time than the local search heuristic 2. The tabu search heuristics were run with a tabu list with 20 elements over 50 iterations. Although the computing times cannot be compared with the complete search, they are quite small. It can also be seen that the tabu search methods do not always find the optimal flexibility.

Further extensions for the tabu search could be implemented. The computational times could be improved by using dynamic or adaptive sizes of the tabu list. Also other heuristics could possibly improve computational times, optimality frequence and/or deviation. However, the methods that have been developed have been proven to work quite well for the problems studied in this chapter.

In the next chapter, extensions to the approach are presented in order to improve the functionality of the model and to propose a more realistic model.

Chapitre 4

Extensions

To qualify tools for recipes while only considering the current situation in the workshop, with all qualification considered as equally easy to qualify, is a limited view. Therefore, other factors have been regarded in order to achieve a more realistic approach. Information on how WIP quantities are changing in the future can be included in the model. Furthermore, factors which affect how easy or difficult is a qualification can also be considered. A recipe which can be qualified very fast on a tool may be preferred to a recipe which would take longer time to qualify even if the latter would increase the flexibility more. Also groups of similar recipes can be taken into account, since qualifying one recipe from one group on a tool, may make it easier to qualify other recipes of the same group on that tool. Finally, statistics on the availability of the recipes on the tools can be considered in the model.

4.1 Anticipating Dynamic WIP Quantities

Semiconductor manufacturing is exposed to outer and inner changes. Outer changes occur when demands from customers are changing, products are renewed, and new technologies must be started. From the inside, equipment status changes may occur due for example to downtimes and also that different jobs occur irregularly. These changes need to be handled with caution. Actions, such as qualifications, should not be taken while only considering the current status of the fab but should also take into account future plans. Therefore dynamic approaches considering how the WIP quantities change over several periods have been proposed.

4.1.1 Periodical weights

After discussing with production managers, it became clear that WIP quantities for different periods cannot be valued equally. The later a period, the more uncertain is the forecast of WIP quantities. It also became clear that it is normally too late to take actions for WIP quantities that are placed in front of the tools. It has hence been decided to give each periods $p \in \{1, .., P\}$) a weight, d_p, such that later periods are considered less important than earlier periods, except for the current period which also should be considered to have a relatively small importance, because of the time it takes to qualify recipes. The periodical weights have been defined such that $\sum_{p=1}^{P} d_p = 1$.

Planning qualifications in advance for future periods was also considered using variable $OQ_{p,r,t}$ which is equal to 1 if recipe r should be qualified on tool t at the beginning of period p. This approach was not pursued in this thesis since it did not seem to be of practical use at the time. However, it might an interesting topic for future research.

4.1.2 Formulating dynamic flexibility

Two formulations for anticipating dynamic WIP quantities for the flexibility measures are proposed from Formulation (3.1). In the first formulation (4.1), the weight d_p is multiplied with the WIP quantities $WIP_{r,p}$ of each period. The products $d_pWIP_{r,p}$ for all periods are added together to determine WIP_r for each recipe r. With these new WIP quantities, it is possible to use one of the flexibility measures from Chapter 2. In Formulation (4.1), the system flexibility measure F^{SYS} is used, since it is the most general of the flexibility measures which can be transformed to any of the other measures by setting two of the parameters a, b or c to 0. Furthermore recall that $OQ_{r,t}$ is equal to 1 if recipe r should be qualified on tool t, and 0 otherwise, and that the maximum number of qualifications that can be proposed is k.

$$
\begin{aligned}
&\max F^{SYS} \\
&WIP_r = \sum_{p=1}^{P} d_p WIP_{r,p} && r = 1,..,R \\
&\sum_{t=1}^{T} WIP_{r,t,p} = WIP_{r,p} && r = 1,..,R \quad p = 1,..,P \\
&WIP_{r,t,p} \leq (Q_{r,t} + OQ_{r,t})\, WIP_{r,p} && r = 1,..,R \quad t = 1,..,T \quad p = 1,..,P \\
&\sum_{r=1}^{R}\sum_{t=1}^{T} OQ_{r,t} = k \\
&OQ_{r,t} \in \{0,1\} && r = 1,..,R \quad t = 1,..,T \\
&WIP_{r,t,p} \geq 0 && r = 1,..,R \quad t = 1,..,T \quad p = 1,..,P
\end{aligned}
\tag{4.1}
$$

The second way to consider the dynamic changes is to calculate the system flexibility measure F_p^{SYS} for each period as in Formulation (4.2). The periodical weights are multiplied with the flexibility measures at each period and added together.

$$
\begin{aligned}
&\max \sum_{p=1}^{P} d_p F_p^{SYS} \\
&\sum_{t=1}^{T} WIP_{r,t,p} = WIP_{r,p} && r = 1,..,R \quad p = 1,..,P \\
&WIP_{r,t,p} \leq (Q_{r,t} + OQ_{r,t})\, WIP_{r,p} && r = 1,..,R \quad t = 1,..,T \quad p = 1,..,P \\
&\sum_{r=1}^{R}\sum_{t=1}^{T} OQ_{r,t} = k \\
&OQ_{r,t} \in \{0,1\} && r = 1,..,R \quad t = 1,..,T \\
&WIP_{r,t,p} \geq 0 && r = 1,..,R \quad t = 1,..,T \quad p = 1,..,P
\end{aligned}
\tag{4.2}
$$

4.1.3 Example

In Table 4.1, an example with WIP quantities of three recipes for four periods is shown. The WIP quantities which currently have to be processed on the tools are multiplied with a periodical weight $d_0 = 1/8$. The WIP quantities that will arrive in the next 24 hours are given a periodical weight of $d_1 = 1/2$, from 24 to 48 hours, the weight is $d_2 = 1/4$ and, from 48 to 72 hours, the weight is $d_3 = 1/8$.

Period	now	-24h	-48h	-72h	$\sum WIP_p d_p$
d_p	1/8	1/2	1/4	1/8	
$WIP_{r=1}$	200	50	100	65	83
$WIP_{r=2}$	0	100	50	100	75
$WIP_{r=3}$	100	60	80	50	69
F^{SYS} for $\sum d_p WIP_p$					0.65
$\sum d_p F_p$	0.78	0.53	0.63	0.48	0.58

TAB. 4.1 – Example of dynamic WIP quantities

In Table 4.1, the WIP quantities for each recipe of period p are multiplied with the periodical weights d_p and added together in the last column. The first dynamic flexibility model considers only the added WIP quantities to calculate F^{SYS}. In the second model, the flexibility measure is first calculated for the WIP quantities of each period and the flexibility measures are multiplied with the periodical weights and added together.

4.1.3.1 Time constraint

To make the model even more realistic, a time constraint could be added to ensure that a tool cannot be loaded more than a maximum WIP quantity per period. If the length of a period is P time units, the following constraint should then be added :

$$\sum_{r=1}^{R} \frac{WIP_{r,t,p}}{TP_{r,t}} \leq P \qquad t = 1,..,T \quad p = 1,..,P \tag{4.3}$$

Analyzing whether this constraint is really relevant in practice, and considering it in the optimization has been left for future research

4.1.4 Numerical experiments

The current and future WIP quantities in a photolithography area have been considered for the next 36 hours. The WIP quantities have been divided such that the WIP quantities which currently await to be processed are in Period P_0, the WIP quantities that arrive in the next 12 hours belong to Period P_1, the WIP quantities arriving from 12 hours until 24 hours belong to period P_2, and the WIP quantities arriving from 24 hours until 36 hours belong to Period P_3. The WIP quantities for ten recipes are displayed in Table 4.2. The weights of the periods are set to $d_0 = 0.125$, $d_1 = 0.5$, $d_2 = 0.25$ and $d_3 = 0.125$.

	Periods			
Recipes	P_0	P_1	P_2	P_3
1	100	0	175	125
2	75	125	75	150
3	175	300	125	175
4	95	250	250	100
5	150	125	175	225
6	0	200	50	175
7	50	100	75	0
8	25	175	100	175
9	0	300	164	98
10	75	200	275	200

TAB. 4.2 – WIP quantities over four periods

	Tools					
Recipes	A	B	C	D	E	F
1	-	-	X	-	-	-
2	-	-	-	-	X	-
3	-	-	-	X	-	X
4	-	-	X	X	-	-
5	-	-	-	-	X	X
6	-	-	-	-	X	-
7	-	X	X	-	-	-
8	X	-	-	-	-	-
9	-	-	-	-	X	X
10	X	-	X	-	-	-

TAB. 4.3 – Initial qualifications of the experiment.

The initial qualifications of the toolset are displayed in Table 4.3. The WIP flexibility measure F^{WIP} has been used to calculated which qualifications should be conducted. Using Model (4.1), $F^{WIP} = 0.909$ and, using Model (4.2), $F^{WIP} = 0.901$, for the current setting. If two qualifications are considered, the first measure recommends to qualify Recipe 9 on Tool B and Recipe 2 on Tool B. This increases F^{WIP} to 1.0, i.e. total flexibility. The second measure recommends to qualify Recipe 5 on Tool B and Recipe 2 on Tool B, which results in $F^{WIP} = 0.996$.

It can be noted that the first method recommends a qualification on a tool which has a high workload in Period P_1, whereas the second method recommends a qualifications on a tool which has a quite high workload in all periods.

4.2 Further Extensions

In order to increase the functionality of the flexibility model, further extensions have been introduced.

4.2.1 Recipe hold types

There are different reasons why a recipe is not qualified on a tool. In the input data it is noted if the recipe has been qualified but has later been put temporarily on hold. There are several reasons why recipes are put on hold. A list of different hold constraints used at the STMicroelectronics Crolles300 site is displayed in the list below. When a hold constraint is enabled for a recipe, it means that production of that recipe or capability cannot continue on a chamber or a tool. Capability specifications are put on lots defining quality criteria, so that tools which cannot fulfill these criteria cannot process these lots.

- Hold PPID - A PPID lists all chamber combinations that a recipe can use on cluster tools. If a chamber becomes unavailable on a tool, or if the chamber does not fulfill the quality requirements of the recipe, the PPIDs including that chamber are put on hold.
- Hold Recipe - If it is seen that a tool cannot fullfil the quality requirements, the recipe is put on hold for the entire tool.

– Hold Recipe group - All recipes belonging to a group of recipes with the same characteristics are put on hold.
– Hold Capability - A capability is put on hold.
– Unknown - Often no reason is given why the recipe is not processable. In most cases, it means that the recipe has never been qualified on that tool.

The hold constraints have been implemented in the model such that it is possible to choose which kind of the hold types will be considered for qualifications. The process engineer might only want to consider qualifications for recipes with the hold capability constraint. The rest of the hold constraints are excluded from the calculations.

In the program a hold constraint $h_{r,t}$ has the boolean value *true* if recipe r has the constraint on tool t and the user has enabled to display the constraint type and *false* otherwise. The program will then calculate the chosen flexibility measure for the case when r would be qualified on t.

4.2.2 Easiness levels of qualifications

Qualifying a recipe on a tool takes time. Tests need to be run in order to see if the tool can fulfill all the requirements of the recipe. Also parameter settings might need to be installed on a computer. This might require manpower and trained personnel. In other cases, qualifications of recipes on tools might be rather easy to perform by the operator just before the process starts. Hence some recipes are considered to be easier to qualify than others. Therefore, qualifications are sorted into different levels depending on how easy they are to qualify. For example, three easiness levels can be defined such that :
– "1" means easy to qualify,
– "2" means medium hard to qualify,
– "3" means hard to qualify.

The number of qualifications that are allowed for each easiness level is specified. For example three easy qualifications, one medium and one hard. Table 4.4 shows three recipes and three tools with different difficulties to qualify. Qualifying recipe 1 on tool A is considered to be easy, whereas qualifying recipe 2 on tool C is hard.

	Tools		
Recipes	A	B	C
1	1	3	1
2	2	2	3
3	1	1	2

TAB. 4.4 – Examples of qualifications with different easiness levels.

The easiness levels have been implemented such that the user states how many qualifications k_i^{max} are allowed for each easiness level i. The program then allows a qualification of recipe r on a tool t belonging to easiness level i if the number of qualifications k_i already conducted for i is smaller than k_i^{max}.

4.2.3 Recipe groups

Recipes of similar type form groups. They share similar characteristics and hence similar setup tests need to be performed in order to qualify recipes from the same group. Therefore, when a recipe is qualified on a tool, it will be easier to qualify the other recipes from the same group on the same tool, since it is not needed to perform the same tests again.

This has been regarded such that, in order to qualify a recipe, it is first needed to qualify the group of the recipe on the tool. Once the group has been qualified on a tool, all recipes of the group may be qualified. As well as it is possible to state how many recipe qualification that are allowed, it is also possible to state how many groups can be qualified.

A qualified group for a recipe r on a tool t is defined as $g_{r,t} = 1$ and $g_{r,t} = 0$ if it is not qualified. Moreover it is checked if the number of groups *groupIter* is smaller than the number of allowed group qualifications *maxGroups*. Recipe r may be qualified on a tool t if its group already has been qualified on tool t, i.e. $g_{r,t} = 1$ or if *groupIter* is strictly smaller than *maxGroups*.

The recipe groups have been implemented as a standard in the developed software. If the user does not want to include recipe groups in the calculations, all recipes belong to one group that is qualified on all tools.

4.2.4 Cluster tools

On cluster tools, like in the etch area, parallel processing of wafers from one lot can be performed simultaneously on several chambers. Figure 4.1 shows a sketch of a typical cluster tool for 300mm semiconductor manufacturing. The tool consists of three load ports (LP1, LP2 and LP3), where the lots are deposed. From the load port,wafers are unloaded from the lot, during the time the wafers are processed until the wafers are loaded back into the lots. Since there are only three load ports, the wafers from no more than three lots can be handled simultaneously. When a wafer is unloaded from the lot, it is transported into one of the two load locks (LL1 and LL2). In the locks, the atmosphere is adjusted such that it has the same characteristics than the atmosphere in the processing chambers. There is only place for one wafer in the load locks, which means that the load locks sometimes can be bottlenecks for the processing. The wafers are transfered from the load locks into one of the four processing chambers (A, B, C and D) by a robot. Many new cluster tools come with a robot with two arms in order to speed up the transport process inside the tool, such that, when one wafer is taken out of the chamber, it can quickly be replaced by another wafer. When the process of a wafer is finished, the wafers are transported back via the load locks into their original lot.

On cluster tools, it is considered better to qualify a recipe on a chamber for a tool where another chamber has already been qualified, than to qualify the recipe on a chamber for a tool where no other chamber has been qualified. The reason is that the wafers of a lot can be processed faster. This helps the load ports of the tool to be available earlier for other lots. Also the load locks of the tools can be used more effectively if more chambers are used for processing the wafers in a lot. Moreover, the robot arm, which transports the wafers between the load locks and the chambers, cannot work efficiently if recipes with different throughput rates are processed simultaneously on a tool.

At the STMicroelectronics Rousset site, it has been estimated that a cluster tool for parallel processing is losing about 20% of its throughput rate when different recipes are processed on the tool simultaneously.

In order to cope with this, it has been implemented that it is first needed for a tool to be qualified for a recipe before a chamber is qualified. A tool where a

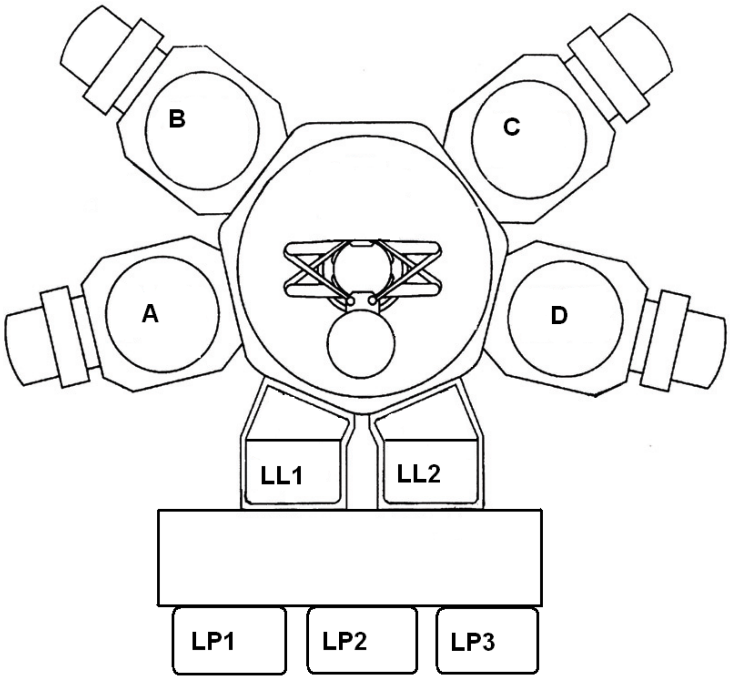


FIG. 4.1 – A cluster tool

recipe already has been qualified on a chamber is considered to be already qualified. I.e., for processing a recipe both the tool and the chamber must be qualified. By allowing fewer qualification of tools than chambers, recipes are more often qualified on chambers of tools on which they are already qualified.

In the implementation of chamber tools, the program initially checks if any chamber c of a tool t has been qualified for a recipe r. If this is the case, *i.e.* $Qc_{r,c} = 1$, the tool t is qualified $Q_{r,t} = 1$ otherwise $Q_{r,t} = 0$. For each qualification of a recipe r on a chamber c, it is checked if the tool t of chamber c is already qualified. If $Q_{r,t} = 1$, chamber c may be qualified. If $Q_{r,t} = 0$, it is checked if tool t can be qualified. Tool can be qualified if the number of qualifications k^t is smaller than the maximum number of allowed qualifications of tools k^t_{max}.

The model of chamber tools is implemented as a standard for calculations within the developed program. For non cluster tools, all tools belong to one fictitious chamber tool which is qualified for all recipes.

4.2.5 Recipe/tool availability

Fab statistics is kept on how often a recipe is put on hold for all tools. Hence, it is also known how often the recipe is available for processing on a tool. I.e., if a recipe has been on hold on a tool 20% of the time during a time period, the recipe availability is 80% on the tool. A small availability rate indicates that it might be hard to process the recipe on that tool and, in that case, it would be better to process the recipe on a tool with higher availability.

This information can be integrated in the flexibility measures ; an operator rather wants to qualify a recipe where he knows that the availability is high. This is why the availability information has been integrated in the toolset flexibility measure.

The toolset flexibility measure (2.2) from Chapter 2 is recalled below :

$$F^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} (WIP_r / NQT_r)}$$

An availability parameter $\alpha_{r,t} \in [0,1]$, which is equal to 0.8 if the availability of recipe r on tool t is 80%, can be integrated in the toolset flexibility measure as in (4.4). The parameter $Q_{r,t}$ is used, which is equal to 1 if the recipe r is processable on tool t and 0 otherwise.

$$F_{\alpha}^{TS} = \frac{\sum_{r=1}^{R} WIP_r}{T \times \sum_{r=1}^{R} \frac{WIP_r}{\sum_{t=1}^{T} \alpha_{r,t} \cdot Q_{r,t}}} \tag{4.4}$$

The toolset flexibility measure F^{TS} can only be used to specify which recipe should be qualified. Using the measure F_{α}^{TS}, it is also possible to have information on which tool the recipe should be qualified on. It is better to qualify a recipe on a tool which is available more often.

In the developed software, the toolset flexibility measure F^{TS} presented in Chapter 2 has been replaced by the measure F_{α}^{TS} presented in this section. Though, by default, the availability parameter is set to 1 for all qualifications so that the two measures are equal. However, a user of the program can at anytime change the values of the availability parameters.

4.3 Conclusions

In order to increase the practical usability of the flexibility measures and the approaches, various extensions have been taken in to account in order to optimize the recommendations of which recipes to qualify on which tools.

Two different ways of anticipating WIP quantities that are arriving in future time periods have been developed. The first method is faster, while the second method finds a more robust solution which optimizes the flexibility for all periods.

There are various reasons why recipes have not yet been qualified on a tool. Different so called *hold* conditions have already been defined in the input data. An extension to the model has been implemented such that it is possible to select which hold conditions should be considered.

Furthermore it is possible to define and sort qualifications in different easiness levels, depending on how easy they are to conduct. It can be chosen how many qualifications from a certain easiness level may be performed and the system will find the optimal qualifications for each level.

Moreover groups have been considered where recipes cannot be qualified on a tool before the recipe group has been qualified on the tool.

Cluster tools where processing on several chambers can be performed in parallel have been studied. It is considered better to qualify a recipe on chambers for tools which already have other qualified chambers, than to qualify a chamber on a tool where no chambers have been qualified for the recipe. An extension has been implemented, which requires that both a tool and a chamber must be qualified before the recipe can be processed. If the number of tools that can be qualified is smaller than the number of chambers, the program proposes to qualify several chambers on the cluster tools.

Finally the availability of recipes on tools has been considered by keeping statistics on how often the recipes are put on hold on the tools. This has been implemented in the toolset flexibility measure such that it is not only important to decide the recipe that should be qualified but also which tool is preferable for the recipe.

Chapitre 5

Impact of Qualification Management on Scheduling

In the previous chapters the concept of qualification management was developed. Measures were defined in order to see which qualifications increase the flexibility of workshops in wafer fabs the most. The flexible qualification model have been developed and extended in order to have a more realistic model. We are now ready to tests the outcome of the model. To see which impact qualifications based on flexibility measures have on fab performance (see Figure 5.1), tests with the two scheduling simulators and one have been conducted.

After studying the literature on scheduling, three schedulers for different workshops are presented. The scheduler for photolithography area in [113] is recalled. Thereafter the main characteristics of the etch workshop are described. This information is used for deriving a new scheduling simulator for a etch workshop. Furthermore the a batch optimization solver for the diffusion area developed in [112] is described.

These simulators are used to tests the impact of qualifications that gradually increase the flexibility measures have been implemented. With these tests it is possible to see how effective the flexibility measures really are.

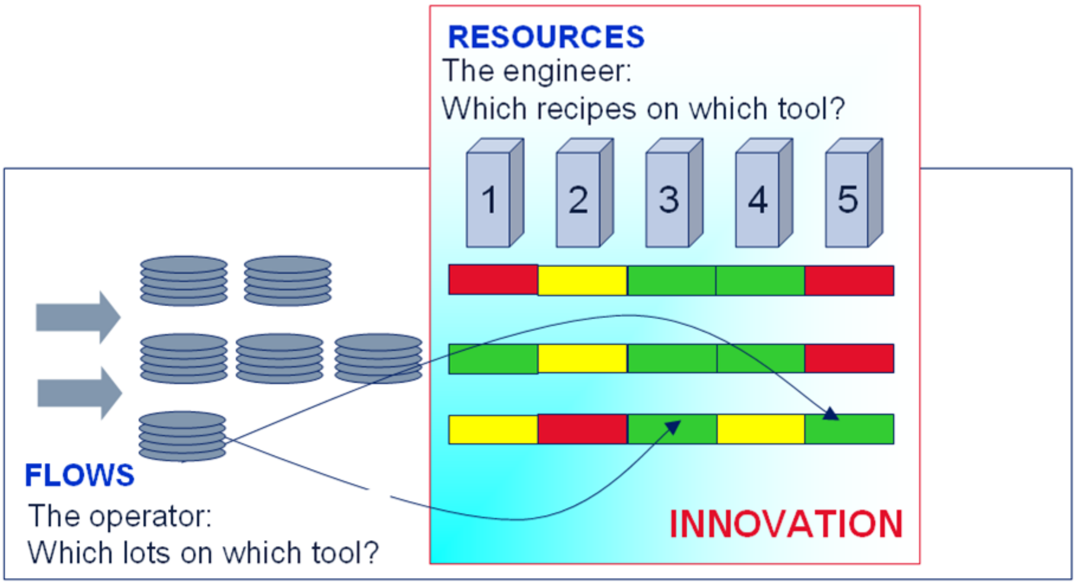


FIG. 5.1 – By qualifying new resources, engineers can help operators to improve production flows.

5.1 Scheduling

Scheduling is an important area for semiconductor manufacturing. However, previous works often include complex models which require long computing times or are too simple and fails to achieve a well functioning schedule. In this chapter a scheduler simulator based on simple rules for the etch area is presented. The scheduler simulator resembles the scheduler simulator for the photolithography area described by Yugma *et al.* [113]. In [113] dispatching rules are used to schedule lots on tools. However, the etch area has different characteristics than the photolithography area, and the rules have been modified.

5.1.1 Literature on scheduling

An exhaustive reference for theory of scheduling and its applications has been written by Błażewicz *et al.* [16]. They differentiate between three different kinds of scheduling in workshops :

– Flow-shop - All jobs have the same fixed processing order (route) on the tools.
– Job-shop - Arbitrary fixed processing order.
– Open-shop - The processing order for each of the tools much be determined.

It has been shown that scheduling lots in these workshop types is difficult. Flow-shop and job-shop scheduling problems [30, 64] are NP-hard and the open-shop scheduling problem is an "especially hard problem" [18, 38].

5.1.1.1 Dispatching and scheduling

To avoid confusion, scheduling and dispatching are first defined. Scheduling in an etch workshop has things in common with scheduling on parallel computing systems, where several jobs can be executed simultaneously. Therefore definitions of scheduling and dispatching from [102] are used. Below the definitions are stated for scheduling and dispatching in a wafer fab workshop :

– Long-term scheduling (or admission) – Authorization, prohibition or delay of jobs which can or cannot be scheduled.
– Mid-term scheduling – Arranging the order of the jobs to obtain an optimal sequence of the schedule.
– Short-term scheduling (or **dispatching**) – Deciding which of the ready-to-be-processed jobs should be executed next and where it should be executed.

In this chapter, the three definitions of scheduling are used. If lots can be processed, then they will be admitted (long-term) depending on their priority and cycle time, their sequence order will be decided (mid-term) and thereafter each lot can be dispatched one by one on the tools (short-term).

5.1.1.2 Scheduling in wafer fabs

Scheduling differs much between different wafer fabs depending on the characteristics of the fab. Sloan [99] compares shop-floor scheduling in 28 wafer fabs with different technologies and performance objectives and finds big differences between how scheduling was regarded and performed. Including all details leads to very complex simulation models [7].

Mönch *et al.* [72] have described decision-making on three levels in wafer fabs for production control which is an hierarchical structure which could also be considered for scheduling :

1. Work center level (also called workshops) - A single group of parallel tools.
2. Work area level - A group of workshops completing an operation.
3. Wafer fab level - All work areas together in the fab.

Additionally internal scheduling of tools must sometimes be considered. Internal scheduling is especially interesting for scheduling wafers inside cluster tools. This has been regarded in several studies both for parallel and sequential processing [81], [105], [26], [111], [101].

5.1.1.3 Scheduling in workshops

Scheduling lots in wafer fab workshops is often a complex job-shop scheduling problem. Since it is a NP-hard problem [30, 64], various approximate approaches to solve the problem have been developed.

A common way to solve job shop scheduling problems is to use a shifting bottleneck (SB) heuristic [3]. Different modifications of the shifting bottleneck heuristic have been developed for complex job-shop scheduling problems [66, 71, 21, 10]

Oechsner and Rose [83, 82] describe how scheduling is performed on one single cluster tool by using filtered beam search and recipe comparison on the chambers of a single tool.

5.1.1.4 Scheduling at wafer area level

Different shifting bottleneck heuristics have been used for decision making for minimizing the total weighted tardiness (TWT) of macro-operations in work areas [70]. In work areas several processes constitute an operation together. These operations either serves to deposing material, removing, modifying or creating patterns on the wafers. Examples of work areas are diffusion (deposing), etch (removing), implantation (modification) and photolithography (patterning). For more information of the fabrication in work areas see Chapter 1.

5.1.1.5 Scheduling on wafer fab level

At wafer fab level, the wafers routes through the fab are considered. This is complex problem since semiconductor manufacturing is signified by its reentrancy flow where lots passes the work areas several time during its way through the fab. A way to model the routes in a wafer fab is presented in [103]. The complex reentrancy flow makes it especially important to have knowledge when lots are completed at one work area and is ready for an operation at the next. A way to minimize the deviation of the completion time of lots is presented by [39].

5.1.1.6 Scheduling in three different workshops

In this thesis, scheduling tools for three different workshops are used :

- Photolithography workshop - Sequential processing on parallel tools.
- Etch area workshop - Cluster tools with parallel processing on different chambers of the tools.
- Diffusion workshop - Parallel batching tools.

For the photolithography workshop a scheduling simulator described in [113] is used. The simulator was developed based on some simple principles [106] ; by first sorting the lots based on their priorities and thereafter dispatching them one by one on the tools to obtain a fully scheduled workshop.

Scheduling for the etch area have been implemented in a simulator using similar ideas as for the photolithography scheduling simulator. The scheduling simulator for the etch area is described in Section 5.1.3.

In the diffusion area, batch size optimization is of great importance. This has been treated by Rulkens *et al.* [94]. In this chapter the approach, which both optimizes batch sizes and schedules the batches, described by Yugma *et al.* [112] has been used.

5.1.2 A scheduling simulator for a photolithography workshop

Before describing the new scheduler simulator for the etch area, the characteristics of the scheduling simulator for a photolithography area is recalled [113]. A scheduling simulator that considers a set of lots with wafers that are ready to be processed in a workshop of photolithography tools.

Scheduling is performed in two steps. In the first step, it is decided in which order the lots should be scheduled (scheduling). In the second step, it is decided on which tools lots are processed (dispatching). This is done by first calculating a global weight for the lots, and then, for each lot, a tool rank which specifies on which tool a lot should be dispatched.

5.1.2.1 Global weights

Lots have different priorities, which specify how important the customer planning considers the job of the lot. Priorities are calculated considering customer service, line balancing and on-time delivery.

The DAO of the lot is also considered. The DAO (Day-At-Operation) of a lot is a cycle time measure, which corresponds to the time the lot stays at an operation. The customer planning wants to minimize the cycle times of the lots and hence the lots should not be idle too long in the workshop. Therefore, lots with higher DAO should normally be processed first.

The scheduling simulator first sorts lots using their global weights Wglobal which are based on the priority and DAO of the lots. The lots are then dispatched on the tools one by one ; the lot which has the highest global weight is dispatched first, then the one with the second highest global weight and so on.

5.1.2.2 Tool rank

Then it is decided on which tool each lot is dispatched. For this a tool rank, W_{tool}, is computed for the lot. The tool rank depends on four local rules :

– Tool availability,

– Tool load,
– Mask location,
– Batch configuration.

The scheduling can be performed over several time periods. The tool availability rule checks if it is still possible to schedule the lot within the current time period on a tool. If it is not possible to dispatch the lot on any tool during the current time period, the lot will be dispatched the next time period. The tool loading favors scheduling on the tool where currently not so many lots have been scheduled. On photolithography tools, a mask (or reticle) is used to make patterns on the micro-chips. The mask location rule favors a tool if the mask is already on the tool. A penalty is given if the mask is to be found stored in the stockers next to the photo-lithography workshop. An additional penalty is given if the mask is on another tool than the one that is considered. Finally the tool rank favors a tool when the batch configuration does not need to be changed from the previous lot on the tool.

5.1.3 A scheduling simulator for a etch workshop

Scheduling for the etch area is conducted in a similar way than for the photolithography area, but with somewhat different rules due to the different characteristics of the etch tools. Before explaining how the scheduling simulator is built, these characteristics are explained in detail.

5.1.3.1 Characteristics of etch tools

Tools in the etch area are different from those in the photolithography area in many ways. Photolithography tools process one wafer at a time on several process steps in a sequence :

– Preparation (temperature, gas, liquid setups),
– Photo resist application (a reticle is put on the wafer),
– Ultraviolet light exposure,
– Development of photo resist.

In a dry etch tool, several wafers can be processed simultaneously in parallel chambers. A chamber may process a wafer from one lot whereas another chamber may simultaneously process a wafer from another lot. The wafers of one lot may also be simultaneously processed on different chambers of the tool. The chamber sequence that is chosen for processing the wafers of a lot is called PPID (Process Program IDentification). For a tool with two chambers (A and B) three PPIDs would be possible : "A", or "B", or "A *and* B".

Figure 5.2 shows a typical cluster tool in the etch area. The tool has three load ports (LP1, LP2 and LP3) where lots are placed. The wafers in the lots are unloaded to one of the two load locks (LL1 and LL2). In the load locks the gas pressure is adjusted for processing module. The transport robot, in the middle of the chamber module, takes out the wafer from the load lock when the right gas pressure is achieved and places the wafer on one of the four chambers (A, B, C or D) which is available. When the wafer is processed the transport robot places the wafer back in the load lock which then goes back to the lot.

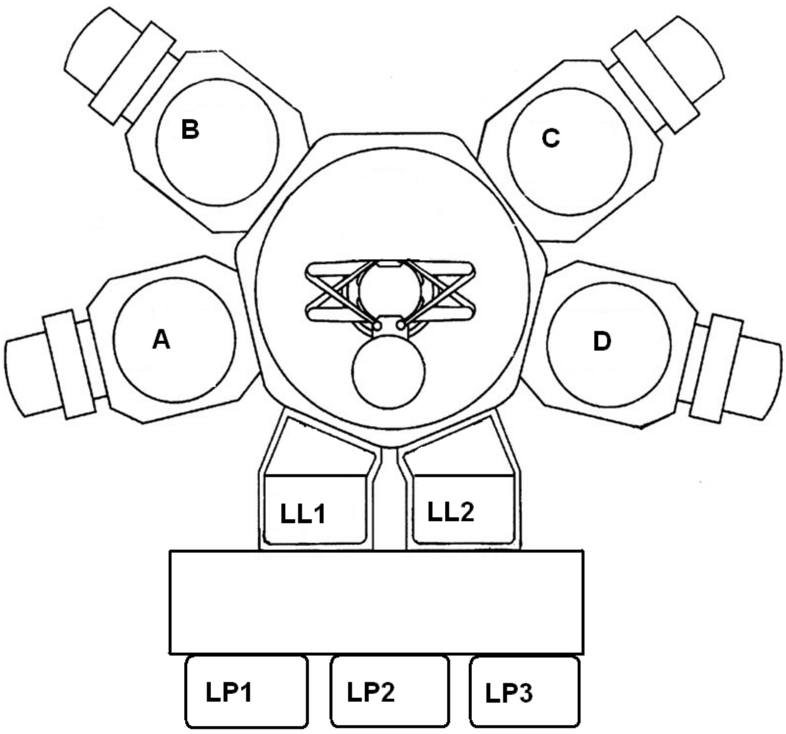


FIG. 5.2 – A cluster tool

5.1.3.2 Scheduling rules

Scheduling rules, considering the characteristics of the tools in the etch area, have been implemented in the scheduling simulator.

Admission

Not all jobs can be processed and hence, lots that cannot be processed should be excluded from the scheduling. As mentioned in Chapter 1, a job may only be processed on a chamber or a tool if the recipe of the process has been qualified on the chamber, otherwise the job will not be authorized on the chamber. Similarly, a lot can be excluded from a tool if its recipe has not been qualified on any of the tool's chambers. The lot can also be totally removed from the dispatching if there is no tool that may process the lot.

Load port rule

It has been decided to add a load port rule. The tools in the considered etch workshop are equipped with two or three load ports. This constraint implies that it is not possible to process wafers from more lots than there are load ports. The constraint forces to only dispatch a lot to a tool when a load port is free, and hence a load port rule has been implemented. The implemented load port rule specifies that, when a load port becomes available, next lot is dispatched on the tool.

Global rule

As with the scheduler simulator for the photolithography area, a global weight is calculated for each lot depending on its priority and DAO. The original global weight of a lot is calculated as in 5.1.

$$W_{global} = \frac{a + PIT \cdot (b + DAO)}{b} = \frac{a}{b} + PIT + \frac{PIT \cdot DAO}{b} \tag{5.1}$$

Where
– PIT is the priority of the lot,

– DAO (Day(s)-At-Operation) is a cycle time measure which corresponds to the time the lot has been staying in the work area,
– a and b are constant terms which specify how big the value of the weight should be.

Since the parameter a obviously does not affect neither the PIT value nor the PIT-weighted DAO value, and that there is no way to regulate the PIT value, a new definition of the global weight is derived in 5.2 :

$$W_{global} = a \cdot PIT + b \cdot PIT \cdot DAO \tag{5.2}$$

The global weights are sufficient to determine in which order the lots should be scheduled, but no information specifies on which tools and chambers they should be dispatched on.

Local rules

Similarly as with the scheduling simulator for photolithography area, the scheduling simulator has local rules. However, since for the etch tools, the load port rule decides on which tool the next lot should be dispatched on, local rules for the etch area scheduling have been designed to evaluate which PPID is best for a lot.

The workshop that has been studied contains two different tool types. One of the tool types handles the set up of capabilities differently than the other tool types. Therefore some of the rules only apply one of the tool types. Tools are denoted *V-tools* which do not need setup times for changing capability type. Tools that need setup times are denoted *K-tools*.

Before stating the local rules, some definitions are needed.

Definition 5.1. ***Desired chambers*** - *The set of chambers on a tool where processing will end first.*

Example 5.1. *A tool has three chambers A, B and C. Wafers from two lots L1 and L2 are ready for processing on the tool. Lot L1 uses all chambers A, B and C for processing and it finishes its processing before processing the wafers of lot L2, whose wafers will be processed on chambers B and C. Since chamber A will be ready for the* *next lot before chambers B and C, it is preferred to process the lot which will use at least chamber A. Chamber A is hence the "desired chamber".*

Definition 5.2. ***Desired capability*** *- The type of capability used by the previous process on the desired chambers.*

Example 5.2. *Considering Example 5.1 suppose that, if the wafers of lot L1 used capability C1 and lot L2 was using capability C2. Since lot L1 will finish first, the desired capability is C1. Changing capabilities requires additional setup, and therefore it is preferred to use the same capability of the chamber for the last lot. From the previous example, it would hence be preferable to choose a lot which needs capability C1 and can be processed on chamber A.*

Definition 5.3. ***SameCapa chambers*** *- All chambers on a tool, where the previous processes have the same capability as the lot that must be processed.*

Example 5.3. *Often no lot which is ready to be processed has the desired capability. However, it will still be better to process a lot which needs the same capability as the last lot being processed. Considering the previous examples, if there are two remaining lots L1 and L2 which can be processed on all three chambers (which are not desired chambers) ; L1 needs capability C2 and L2 needs capability C3. In this case it would be preferable to process L1 since two of the chambers (B and C) used capability C2 for the last process. Capability C3 of L2 needs additional setup and would then not be chosen.*

Definition 5.4. ***DifferentCapa chambers*** *- Chambers that meet the following criteria :*

- *The chamber is not desired,*
- *The capability of the previous lot is different than the one of the lot the must be processed next,*
- *There is another lot, which requires this capability and is processable on the chambers.*

Example 5.4. *From Example 5.3, L2 satisfies all the criteria above for chambers B and C. Chamber A would have been desired if the capability of L2 would be capability C1, but this is not the case. Neither chamber B nor chamber C have the capability of*

L2. Furthermore, there exists another lot, L1, which requires the same capability that was used last on chambers B and C. Hence, chambers B and C are "DifferentCapa chambers". Chamber A is not a "DifferentCapa chamber" since there is no other lot that requires the capability which was last processed on this chamber.

Definition 5.5. ***Capability Constraint Limit (CCL) (or Capability Contention Limit)*** *is a constraint which assures that a capability is not used on too many chambers at the same time.*

Example 5.5. *If the same capability is used on nearly all chambers, it is sure that setup time is required when a lot requiring another capability is dispatched on a tool. Therefore a penalty is added if dispatching a lot on a chamber would mean that a certain capability would be used on more chambers that the CCL-rules have been defined for. Hence, rather than dispatching this lot on the chamber, the lot should wait until a chamber using this capability becomes available.*

Using these definitions, the dispatching rules can be defined.

Maximum series length avoidance

Add a penalty to a PPID if the lot does not have the desired capability : *0* if the capability is the desired and *1* otherwise (valid only for K-tools) :

$$W_1 = w_1 \cdot \{0, 1\} \tag{5.3}$$

This rule avoids to process a lot which is not desired on the chambers.

Full utilization of chambers

Add a bonus to the PPIDs that maximize the usage of desired chambers :

$$W_2 = w_2 \cdot \frac{\#\ desired\ chambers\ used\ by\ PPID}{\#\ desired\ chambers} \tag{5.4}$$

This rule aims at maximizing the usage of desired chambers.

Lot cycle time optimization

a) Add a bonus for the PPIDs that maximize the use of SameCapa chambers (valid only for K-tools) :

$$W_{3a} = w_{3a} \cdot \frac{\#\ SameCapa\ chambers\ used\ by\ PPID}{\#\ SameCapa\ chambers} \tag{5.5}$$

b) Add a penalty for the PPIDs that minimize the use of DifferentCapa chambers (valid only for K-tools) :

$$W_{3b} = w_{3b} \cdot \frac{\#\ DifferentCapa\ chambers\ used\ by\ PPID}{\#\ DifferentCapa\ chambers} \tag{5.6}$$

c) Add a bonus for the PPIDs that maximize the number of used chamber

$$W_{3c} = w_{3c} \cdot \frac{\#\ chambers\ used\ by\ PPID}{\#\ chambers} \tag{5.7}$$

Rule a) helps to avoid additional setup, by only using chambers with the same capability if it is possible. If this is not possible, rule b) makes sure that a chamber is not used if it is desired by another lot. Rule c) aims at maximizing the number of chambers that are used, so the process will end earlier.

Capability contention limit

Add a penalty if the capability is used on more chambers than the CCL.

$$W_4 = w_4 \cdot \{0, 1\} \tag{5.8}$$

In order to avoid having the same capability on all tools, and hence let different capabilities be used simultaneously, a limited number is set for the number of chambers which uses a capability simultaneously.

Summarized local weight

The local rules can now be summarized as the local weight expressed in 5.9.

$$W_{local} = -W_1 + W_2 + W_{3a} - W_{3b} + W_{3c} - W_4 \tag{5.9}$$

A combined rule

In [113] the scheduling is performed in two steps : the lots are first ordered based on their global weights, and then the lots are dispatched on the tools one by one based on the tool rank.

In the photolithography area, local rules are applied on a lot to select the best tool. However, in the etch area local rules are used to find the best lot (and PPID) for the tool. This makes it possible to combine the global rule and the local rules into one single measure. Adding the global weight and the local tool ranks would result in a weight measure as displayed in Expression 5.10.

$$W = W_{local} + W_{global} \tag{5.10}$$

The advantage of this measure is that it is not required to schedule the lots in the order given by the global rule. For example, it may be possible that the lot with the largest global weight is not at all suited to be dispatched on the tool which is currently regarded or worse, that the recipe of the lot is not qualified on the tool. Instead the (lot,PPID) couple, that suits the tool the best, is chosen. A disadvantage could be that longer computing times are expected. However, the computing times which are experienced from the implementation of the algorithm are moderate.

5.1.3.3 Scheduling algorithm

The algorithm starts by considering which lots can be processed, and hence should be scheduled, and which cannot be processed, and hence should not be scheduled. A tool can also be excluded from the schedule when there is no more lots that can be processed on the tool.

Then lots are dispatched one by one on the tools as long as there is still a free load port on at least one of the tools. The lot/PPID-couple which has the highest weight (5.10) is dispatched at each step. When all load ports are occupied, the next lot is dispatched first when a lot exits a tool and a new load port becomes available. The schedule is completed when all authorized lots are dispatched.

5.1.3.4 Example

Let us use an example in which a load port becomes available on a tool, and there is a short WIP list containing only two lots (L1 and L2). Lot L1 can be processed on chambers A and/or B on the tool and lot L2 can be processed on chambers B and/or C. In Table 5.1 all possible (lot,PPID) couples are listed together with a fictitious weight associated the couple. The idea is that, once all the weights are calculated, the couple with the highest weight should be dispatched on the tool.

Lot	I	I	I	II	II	II
PPID	A	B	AB	B	C	BC
Weight	0.23	0.06	0.15	0.16	0.22	0.25

TAB. 5.1 – Example with weights for (lot,PPID) couples.

The best combination would in this case be to process lot L2 on chambers B and C. As illustrated in Table 5.1 for lot L1, it is not always preferable to process a lot on all its possible chambers. In the example the PPID AB has a lower weight for lot L1 than PPID A. This could be due to the fact that chamber B is a so called DifferentCapa chamber or that the *capability contention limit* is achieved.

5.1.4 Batch optimization solver for a diffusion area

With batching is meant that several lots are jointed together and processed simultaneously on a tool. This allows decreasing cycle time of lots in a workshop, but it also increases the complexity of the scheduling in the workshop, since another factor needs to be considered.

In the diffusion area, batching and scheduling of lots are performed on two different types of equipment : cleaning tools and furnaces [112]. One operation of one or several lots, which is defined by a recipe, can be performed on different tools in a workshop.

There are two types of constraints in the diffusion area : tool constraints and process constraints and line management constraints. The constraints are further detailed below :

1. Tool constraints
   - *Dedicated tools* : A tool can only process a limited number of processes.
   - *Maximum batch size* : The size of a batch is limited on the tools.
   - *Loading and unloading* : It takes time to load and unload a batch on a tool.
   - *Unavailability periods* : Tools need maintenance and to be repaired.
   - *Furnace inspection* : Furnaces may need to be inspected before a new batch can be processed.
2. Process constraints
   - *Preceding lots* : An operation of a lot cannot be processed until the previous operation of a lot is completed.
   - *Material handling time* : The transport time for a lot between two operations is considered.
   - *Release dates* : Lots cannot be processed before they arrive in the workshop.
   - *Recipe* : All lots in a batch must have the same recipe.
   - *Process time* : The time it takes to process a batch depends on the recipe.ends on the recipe.
   - *Maximum time lag* : A maximum time given to the process of two successive operations for a given lot.

These constraints have been implemented in an optimization-based software called Batch Optimization Solver (BOS) as described by Yugma *et al.* [112] and implemented in a wafer fab. The solver constructs a schedule by simultaneously making the following decisions :

1. Sorting lots into batches,
2. Select which tool should process which batch,
3. Order the batches on the tools,
4. Decide the start time for each batch.

The tools in the tests are considered as *identical* according to the definition of [90] : the process times of a recipe are the same on all tools in the workshop. Different recipes are, however, processed with different throughput times.

5.2 Impact of Qualifications on Scheduling in a Photolithography area

In order to see the impact of qualifications, it has been decided to test different qualification sets on the scheduling simulator recalled in Section 5.1.2. Input data for the tests have been obtained from the STMicroelectronics 300mm front-end wafer fab of Crolles. The tests have been performed for a toolset with six tools and lots with different WIP quantities [54].

Tools A, B and F have the same process times. Also, the throughput times for Tool D are the same as for Tool E, whereas the process times for Tool C are independent from the other tools. In the initial set, no lot could be processed on Tool D since none of the recipes were qualified on this tool.

The batch-train (also called block) configurations are decided by the recipe and the photo resist that is used. If the configuration of the previous lot is different than the current one the configuration setup is set to five minutes. The setup of a new reticule on a tool takes one minute. For all tests using the WIP flexibility measure or the time flexibility measure, the parameter γ is equal to 4.

5.2.1 Performance measures

The performance measures that have been used for the tests are the pre-defined measures used by STMicroelectronics at their 300mm site in Crolles, and not the ones studied in Chapter 1. Nevertheless, the performance metric also measures : cycle time, variability and throughput.

- DAO – Day(s)-At-Operation - How long time the lots stay on average in the work area.
- σDAO – The standard deviation of the DAO.

- 8th decile DAO – The DAO for the lot where 80 percent of the lots have lower DAO.
- Max tool charge – The total WIP quantity of the tool with the largest total WIP quantity.
- Number of dispatched lots – How many lots can be dispatched.

Since fab managers want to minimize the cycle times for the lots, it is important to decrease the average DAO. The standard deviation of the DAO and the 8th decile of the DAO indicate the variability of the cycle time. The max tool charge indicates if the lots can be well spread on the tools. A large max tool charge can be used as a throughput measure and indicates that it takes long time until the last lot of the workshop is processed.

By considering the number of dispatched lots, it can be seen if the set of qualifications do not allow the lots to be dispatched anywhere. The number of dispatched hot lots is also considered. Hot lots (or ultra handy carry lots) are lots with very high priority. Since these lots are very important for the company, it is considered very bad if any of the hot lots cannot be dispatched on a tool because the required recipe is not qualified anywhere or if it is only qualified on one tool or chamber and this tool or chamber becomes unavailable for production.

Note that all qualifications increase the toolset flexibility measure (and therefore also the system flexibility) but not necessarily the WIP flexibility measure or the time flexibility measure. The different flexibility types will be analyzed separately in the sequel.

5.2.2 Numerical Experiments

Tests have be performed with qualifications that increase WIP flexibility, time flexibility, toolset flexibility and system flexibility on the photolithography workshop describe above. Additionally it has been tested if flexible qualifications can reduce the negative impacts induced by the unavailability of tools.

5.2.2.1 WIP Flexibility Qualifications

In this first series of tests, qualifications that optimize the WIP flexibility have been chosen. A test series with 239 lots and 4397 wafers are selected. The results of the qualifications for the photolithography area can be seen in Table 5.2.

Nb of additional qualifications	0	1	2
F^{WIP} (%)	**46.1**	**89.6**	**100**
F^{time} (%)	35.1	54.7	55.0
F^{TS} (%)	46.8	49.0	60.3
DAO	32.9	31.0	30.9
σDAO	24.1	24.8	24.9
DAO 8^{th} decile	46.1	46.1	46.1
Max tool charge (WIP)	1182	1177	1167
Max tool charge (h)	23.0	17.2	16.6

TAB. 5.2 – Qualifications in the photolithography area based on WIP flexibility.

Since a toolset with high WIP flexibility allows WIP quantities to be well distributed on the tools, it seems logical that the max tool charge decreases, which is confirmed by the tests. The tests with increasing WIP flexibility shows also that the DAO value decreases quite much until the WIP flexibility also becomes 100 %. The values of σDAO indicate that the variability of the DAO increases a little as WIP flexibility increases. This is due to the fact that no means in the scheduler has been implemented in order to decrease variability. However, the 8^{th} decile remains the same, i.e. jobs with high cycle times are not further delayed.

In order to see that performance improvements are not always obtained with additional qualifications, qualifications which do not increase the WIP flexibility measure have been performed (Table 5.3). For the first qualification nothing happens. When the second qualification is performed, the DAO slightly decreases but on the other hand the max tool charge slightly increases.

Nb of additional qualifications	0	1	2
F^{WIP} (%)	**46.1**	**46.1**	**46.1**
F^{time} (%)	35.1	35.1	35.1
F^{TS} (%)	46.8	46.8	46.9
DAO	32.9	32.9	32.8
σDAO	24.1	24.1	24.1
DAO 8^{th} decile	46.1	46.1	46.1
Max tool charge (WIP)	1182	1182	1142
Max tool charge (h)	23.0	23.0	23.9

Tab. 5.3 – Qualifications in the photolithography area which do not improve WIP flexibility.

5.2.2.2 Time Flexibility Qualifications

Starting from the same qualification set as in the previous example, qualifications based on the time flexibility measure have been conducted (Table 5.4).

Nb of additional qualifications	0	1	2
F^{WIP} (%)	46.1	89.6	89.6
F^{time} (%)	**35.1**	**73.6**	**80.0**
F^{TS} (%)	46.8	57.1	60.3
DAO	32.9	30.3	30.1
σDAO	24.1	25.0	25.1
DAO 8^{th} decile	46.1	45.8	43.4
Max tool charge (WIP)	1182	1167	1167
Max tool charge (h)	23.0	18.7	18.8

Tab. 5.4 – Qualifications in the photolithography area based on time flexibility.

The qualifications based on time flexibility indicates an even larger decrease of DAO than the qualifications based on WIP flexibility. In this case, the variability increases a little. On the other hand, the 8^{th} decile of the DAO decreases, which indicates that cycle times decrease for lots with large DAO. The max tool charge measured in WIP quantity is larger than for the qualifications based on WIP flexibility, whereas the value of max tool charge measured in process time is lower. This is interesting since the scheduling simulator does not explicitly try to minimize the process times. Hence, this effect is only due to the qualifications which improve the time flexibility. However, it can be seen that the max tool charge time of a tool is slightly larger after two qualifications than after one qualification. If the throughput times of the tools are further studied in Table 5.5, it can be seen that the throughput times are better balanced on the tools and that the total production time of the tools is smaller.

Tool	A	B	C	D	E	F	Total TP
1 qualification	12.9	13.3	13.5	14.7	18.7	11.4	84.5
2 qualifications	13.0	12.9	13.0	14.7	18.8	11.4	83.8

Tab. 5.5 – Throughput times on tools.

Hence, it could be concluded that the qualifications based on time flexibility have a considerable effect both on cycle times (DAO) and throughput, compared with qualifications based on WIP flexibility.

In order to make sure that the effects depend on the fact that the selected qualifications increase time flexibility, a series of tests with qualifications that do not increase time flexibility have been performed (see Table 5.6). It can be seen that the DAO remains the same for all qualifications. However, for the first qualification, the max tool charge time is much lower, but increases again with the second qualification. This is because of the number of setups for batch-train configurations. In the initial case, 30 batch-train configurations are performed. After the first qualification, there are 23 and after the second there are 34.

5.2.2.3 Toolset and System Flexibility Qualifications

The toolset flexibility measure only indicates which recipe needs more capacity and not on which tool a qualification should be conducted. Therefore different possibilities for qualification strategies for the tools are possible. Table 5.7 shows qualifications on tools for recipes that will increase the toolset flexibility the most without increasing the WIP flexibility and time flexibility measures too much.

Nb of additional qualifications	0	1	2
F^{WIP} (%)	46.1	46.5	46.5
F^{time} (%)	**35.1**	**35.1**	**35.1**
F^{TS} (%)	46.8	57.1	60.3
DAO	32.9	31.8	31.8
σDAO	24.1	24.4	24.4
DAO 8^{th} decile	46.1	46.1	46.1
Max tool charge (WIP)	1182	1177	1036
Max tool charge (h)	23.0	17.2	20.1

TAB. 5.6 – Qualifications in the photolithography area which do not increase time flexibility.

Nb of additional qualifications	0	1	2
F^{WIP}(%)	46.1	46.1	57.9
F^{time}(%)	35.1	35.1	37.0
F^{TS} (%)	**46.8**	**49.0**	**51.4**
DAO	32.9	31.0	31.0
σDAO	24.1	24.8	24.9
DAO 8th decile	46.1	46.1	46.1
Max tool charge (WIP)	1182	1177	1167
Max tool charge (h)	23.0	17.2	16.6

TAB. 5.7 – Qualifications based only on toolset flexibility.

It can be seen that the first qualification leads to a decrease of the DAO which is larger than for the first qualification based on WIP flexibility, but not as much as for the first qualification based on time flexibility. Also the max tool charge time is lower than for the qualifications based on WIP flexibility and time flexibility. This is because of the number of batch-train configuration changes. For the first qualification for toolset flexibility, the scheduling simulator finds a solution where setups for changing batch-train configuration is only needed 24 times compared to 29 times for the first qualification based on WIP flexibility. For the first qualification for time flexibility, 30 batch-train configurations are needed, but still the DAO is considerably lower than for the toolset flexibility. In order to see if it is only the low number of batch-trains configuration that impacts the low DAO times, a new series of tests have been made where 29 (resp. 27) batch setups have been made for one (resp. two) qualification(s) while still maximizing the toolset flexibility and keeping the WIP flexibility and time flexibility low. Table 5.8 shows the result of these tests.

Nb of additional qualifications	0	1	2
F^{WIP}(%)	46.1	46.5	57.9
F^{time}(%)	35.1	35.1	37.0
F^{TS} (%)	**46.8**	**49.0**	**51.4**
DAO	32.9	31.8	32.4
σDAO	24.1	24.4	24.2
DAO 8th decile	46.1	46.1	46.1
Max tool charge (WIP)	1182	1012	1090
Max tool charge (h)	23.0	20.9	20.9

TAB. 5.8 – Qualifications based only on toolset flexibility, and many batch-train configurations.

For the qualifications when the number of batch-configuration does not decrease too much, the DAO and tool charge remain quite high. The DAO even increases a little for the second qualification. On the other hand, the variability of the DAO has slightly decreased after the second qualification.

In Table 5.9, the same recipes as in Tables 5.7 and 5.8 have been qualified, but on tools which also increase WIP flexibility and time flexibility. These are the qualifications that are recommended by the system flexibility measure.

In spite that the number of batch-train configurations remains quite large for these qualifications (28 configurations for both qualifications), the results are very good. The DAO is still slightly larger than for the qualifications based on only time flexibility but the max tool charge is even lower.

The tests indicate that qualifications only based on toolset flexibility do not necessarily improve the performance if they do not also increase WIP flexibility and time flexibility.

Nb of additional qualifications	0	1	2
F^{WIP}(%)	46.1	79.2	79.2
F^{time}(%)	35.1	56.6	64.6
F^{TS} (%)	**46.8**	**49.0**	**51.4**
DAO	32.9	30.9	30.8
σDAO	24.1	25.0	25.0
DAO 8th decile	46.1	46.1	46.1
Max tool charge (WIP)	1181	1006	1050
Max tool charge (h)	23.0	16.4	15.8

TAB. 5.9 – Qualifications based on toolset flexibility and system flexibility.

5.2.2.4 Tool Availability

To see what kind of qualification policies reduce the effects when tools are unavailable (e.g. breakdowns or maintenance), tests have been performed with scenarios where different tools are simulated as being unavailable one by one. In Table 5.10, it can be seen what happens if each of the six tools becomes unavailable separately.

Tool unavailable	0	A	B	C	D	E	F
Nb of dispatched lots	239	210	229	206	239	211	204
Nb of dispatched hot lots	11	11	11	11	11	11	8
DAO	32.9	35.2	34.6	35.9	32.9	35.7	35.9
σDAO	24.1	23.1	23.4	24.8	24.1	23.3	25.9
DAO 8th decile	46.1	47.0	47.0	51.5	46.1	47.0	51.7
Max tool charge (WIP)	1182	1201	1342	1299	1182	1282	1157
Max tool charge (h)	23.0	24.1	24.0	24.3	23.0	24.0	23.5

TAB. 5.10 – Scenarios with unavailable tools.

The unavailability of a tool, may impact the productivity in the workshop quite much. The only case with no effect is when Tool D breaks down since no recipe was initially qualified on this tool. For the other cases, it is not possible to process all lots, the DAO increases and the max tool charge may increase a lot. When Tool F becomes unavailable, three so called hot lots cannot be processed. Hot lots are the lots with the highest priority in the fab, and it is thus rather critical if these lots cannot be processed.

The question is whether additional qualifications based on the flexibility measures could have reduced these effects ? In order to test this, two series of qualifications have been performed. The first series S_{time} mainly increases the time flexibility whereas, for the second series S_{ts}, qualifications which mainly improve the toolset flexibility have been performed. For both series, five qualifications have been performed. In Table 5.11, the flexibility measures for different qualification series are shown.

Flexibility measure	Previous value	S_{time} series	S_{ts} series
F^{WIP} (%)	46.1	100	92.5
F^{time} (%)	35.1	**86.1**	58.0
F^{TS} (%)	46.8	53.0	**70.4**

TAB. 5.11 – Flexibility values for the qualification series.

In Table 5.12, it can be seen how the performances are impacted if S_{time} would have been performed prior the tools becoming unavailable.

Tool unavailable	0	A	B	C	D	E	F
Nb of dispatched lots	239	239	239	222	239	239	204
Nb of dispatched hot lots	11	11	11	11	11	11	8
DAO	30.6	31.5	31.2	32.7	32.2	32.3	33.6
σDAO	25.2	24.8	24.8	26.5	24.3	24.3	27.2
DAO 8th decile	46.6	47.0	46.0	51.4	46.2	46.9	49.5
Max tool charge (WIP)	853	1167	1083	1232	1027	1077	904
Max tool charge (h)	20.4	21.8	21.8	22.4	22.4	21.9	21.3

TAB. 5.12 – Impact of tool unavailability after performing qualifications for series S_{time}.

It can be seen that the performance is significantly improved when all tools are available. The qualifications have also improved the possibility of processing lots when the tools become unavailable. The DAO and the max tool charge remain quite low in spite of the unavailable tools. However, when Tool C and Tool F break down, there is no possibility to process all lots. Additionally, the qualifications have not improved the situation for the hot lots.

In Table 5.13 tests for series S_{ts}, which mainly improves the toolset flexibility, can be studied.

Tool unavailable	0	A	B	C	D	E	F
Nb of dispatched lots	239	239	239	228	239	239	235
Nb of dispatched hot lots	11	11	11	11	11	11	11
DAO	30.8	32.8	32.3	33.1	32.1	32.9	32.2
σDAO	24.9	24.5	24.5	25.0	24.3	24.3	24.9
DAO 8th decile	45.4	45.9	46.8	50.7	45.4	47.2	46.1
Max tool charge (WIP)	881	995	1077	990	957	1034	997
Max tool charge (h)	18.2	21.0	19.5	20.4	22.2	21.6	19.2

TAB. 5.13 – Impact of tool unavailability after performing qualifications for series S_{ts}.

The DAO is slightly larger for these series, but the max work charge is lower in most cases. It is also possible to process all or almost all lots for all scenarios. Only for Tool C and for Tool F there are 11 (resp. 4 lots) which cannot be processed and, in all cases, it is possible to process all hot lots.

In order to better see the effect of unavailable tools, the two cases which are effected the most are shown in Table 5.14 (for Tool C) and Table 5.15 (for Tool F).

Test series	0	S_{time}	S_{ts}
Nb of dispatched lots	206	222	228
Nb of dispatched hot lots	11	11	11
DAO	35.9	32.7	33.1
σDAO	24.8	26.5	25.0
DAO 8th decile	51.5	51.4	50.7
Max tool charge (WIP)	1299	1232	990
Max tool charge (h)	24.3	22.4	20.4

TAB. 5.14 – Results when Tool C is unavailable.

In Table 5.14 it can be seen how both series of tests improve the situation from the initial qualification set. The DAO is slightly better for the qualifications based on the time flexibility (S_{time}), which seems logical since the time flexibility measure has been partly implemented for this purpose. On the other hand, the max tool charge is better for the qualification based on the toolset flexibility measure (S_{ts}). Also the variability is better for the second test series. The main remark is, however, that more lots can be processed. For this scenario the difference is not so large, but if Tool F becomes unavailable, as seen in Table 5.15, the difference is even larger.

Test series	0	S_{time}	S_{ts}
Nb of dispatched lots	204	222	235
Nb of dispatched hot lots	8	8	11
DAO	35.9	33.6	32.7
σDAO	25.9	27.2	24.9
DAO 8th decile	51.7	49.5	46.1
Max tool charge (WIP)	1157	904	997
Max tool charge (h)	23.5	21.3	19.2

TAB. 5.15 – Results when Tool F is unavailable.

When Tool F becomes unavailable, only 222 lots can be processed if the qualifications based on time flexibility (S_{time}) is implemented. For the series S_{ts}, the result is better ; there are only four lots that cannot be processed, and all of the hot lots can be processed. For the series S_{time}, there are three hot lots that cannot be processed, which is a major issue for a wafer fab. Furthermore, the series S_{ts} improves both the DAO and the max tool charge more than the series S_{time}.

5.2.2.5 Conclusions for the photolithography area tests

Experiments show that, the WIP flexibility measure can be used in order to recommend qualifications that improve performance in a photolithography workshop. However, tests show the qualifications based on the time flexibility measure can improve performance even more. Qualifications based on toolset flexibility do not improve the performance as much as the other measures, but reduce the negative impacts of unavailable tools more than qualifications based on time flexibility.

The tests show the importance of all measures : time flexibility improves cycle time and throughput, and more effectively than WIP flexibility (but WIP flexibility can be computed much faster), and toolset flexibility helps to make the production much more robust in case of unavailable tools. Combining two or all three measures into the system flexibility measure, makes it possible to have one measure which helps to improve all aspects.

One effect which has not been considered by the flexibility measures is set up time. It seems like the number of configurations that leads to additional setups can influence performance in a way that is not directly controlled by the flexibility measures.

5.3 Impact of Qualifications on Scheduling in an Etch Area

Similar tests to the photolithography area have also been performed for the etch area, to see how flexible qualifications impact scheduling in a workshop with cluster tools for parallel processing, numerical experiments have been used on data from the ST Crolles300 site. 108 lots with 2063 wafers have been used. The process of the wafers consists of 14 recipes and five capabilities which can be processed on five different tools (A, B, C, D, E). Tool B has two process chambers and the other tools have four process chambers.

5.3.1 Performance measures

The same performance measures as for the photolithography area (Section 5.2) has been used for the etch area. A priority weighted DAO has also been used.

- DAO – Day(s)-At-Operation - How long time the lots stay on average in the work area.
- σDAO – The standard deviation of the DAO.
- 8th decile DAO – The DAO for the lot where 80% of the lots have lower DAO.
- Max tool charge – The total WIP quantity of the tool with the largest total WIP quantity.
- Number of dispatched lots – How many lots can be dispatched.

In the etch area the max tool charge is measured in how many minutes it takes to process all lots on the chamber which has the highest workload. The DAO in the etch area is measured in number of days the lot has been in the work area (in the photolithography area, it was measured in hours).

5.3.2 Numerical experiments

Test series with qualifications which are based on the different flexibility measures have been performed.

5.3.2.1 WIP flexibility qualifications

From the original qualifications in the etch area, tests have been conducted where additional qualifications which increases F^{WIP} have been conducted (Table 5.16).

Nb of additional qualifications	0	1	2	3
F^{WIP} (%)	57.4	68.8	80.9	91.7
DAO	3.880	3.891	3.883	3.837
σDAO	11.23	11.23	11.23	11.24
DAO 8^{th} decile	1.91	1.91	1.91	1.91
Max tool charge	1557	1515	1643	1373

Tab. 5.16 – Qualifications in the etch area based on WIP flexibility.

After the first qualification, the DAO actually increases, but after further qualifications the DAO decreases again. The max tool charge decreases after the first qualification but after the second qualification it increases.

The worse performance is explained by an industrial engineer at the ST Rousset site : qualifications on tools for which no chamber is already qualified can worsen the condition of qualifications. The tool is occupied by different lots at the same time, instead of producing one lot very fast. The load ports and load locks may become the bottleneck of the tool, and further lot may not be processed although chambers are available. Therefore it is preferable to first qualify chambers on tools where the recipe is already qualified on other chambers (see discussion on cluster tools in Section 4.2.4). In fact, in Table 5.16, all three qualifications are performed on chambers on tools where chambers have not been qualified before. Instead a series of qualifications were performed (Table 5.17) on chambers of tools where other chambers were already qualified for the recipe.

Nb of additional qualifications	0	1	2	3
F^{WIP} (%)	57.4	68.8	69.4	70.6
DAO	3.880	3.874	3.857	3.857
σDAO	11.23	11.23	11.24	11.24
DAO 8^{th} decile	1.91	1.91	1.91	1.91
Max tool charge	1557	1560	1561	1561

Tab. 5.17 – Qualifications on tools with chambers already qualified for the recipe in the etch area.

The DAO for the first and second qualifications decreases much more than for the first series of tests. It is, however, not possible to increase F^{WIP} as much when qualifications may only be performed on tools with already qualified chambers. The qualifications do no longer decrease the DAO for the third qualification and the max tool charge cannot be decreased. Hence it is not sufficient to only qualify recipes on tools where chambers are already qualified on other chambers for the tool. A third test series with qualifications was performed in Table 5.18 where qualifications on one new tool and three chambers are allowed. For this test series, both DAO and tool charge is minimized.

Nb of additional qualifications	0	1	2	3
F^{WIP} (%)	57.4	68.8	79.4	84.3
DAO	3.880	3.891	3.872	3.827
σDAO	11.23	11.23	11.23	11.24
DAO 8^{th} decile	1.91	1.91	1.91	1.91
Max tool charge	1557	1515	1589	1373

Tab. 5.18 – Qualifications in the etch area based on WIP flexibility.

5.3.2.2 Time flexibility qualifications

Similar tests have been performed in Table 5.19 with qualifications which increase the time flexibility measure F^{time}.

Nb of additional qualifications	0	1	2a	2b
F^{time} (%)	51.6	62.7	79.4	63.0
DAO	3.880	3.859	3.875	3.853
σDAO	11.23	11.23	11.23	11.24
DAO 8^{th} decile	1.91	1.91	1.91	1.91
Max tool charge	1557	1617	1847	1617

TAB. 5.19 – Qualifications in the etch area based on F^{time}.

The DAO increases more after the first qualification in this case than it was possible for the qualification based on F^{WIP}. However, after the second qualification, the DAO increases. In fact the second recipe qualification (2a) is conducted on a new tool where the recipe previously was not qualified. If, instead, the second qualification would be made for a recipe on a tool with chambers (2b), the DAO would continue to decrease to 3.853 days, which is lower than the second qualification based on F^{WIP} (3.857). The max tool charge actually increases for additional qualifications.

5.3.2.3 Unavailable chambers

Tests have been performed where it has been simulated that different chambers are unavailable for processing. For these tests, three different qualification types have been used : the original qualification set (0), a set with four additional qualifications which increase the WIP flexibility (WIP), and a set with four qualifications which increase the toolset flexibility (TS).

In the etch workshop used in the numerical experiments, there was redundancy in the qualifications, i.e. almost all recipes were qualified on at least two chambers. Only one recipe was only qualified on one chamber and, for this recipe, there was only one lot to consider. However, there are some other differences in solutions depending on the solution values. The average DAO is first studied when the chambers become unavailable in Table 5.20.

Unavailable				
tool	chamber	0	TS	WIP
A	i	3.887	3.826	3.854
A	ii	3.891	3.826	3.847
A	iii	3.885	3.818	3.846
A	iv	3.878	3.811	3.847
B	i	3.895	3.846	3.870
B	ii	3.880	3.882	*3.847*
C	i	3.921	3.856	3.886
C	ii	3.921	3.856	3.886
C	iii	3.904	3.847	3.878
C	iv	3.909	3.858	3.886
D	i	3.911	3.840	3.876
D	ii	3.880	3.876	*3.847*
D	iii	3.919	3.820	3.867
D	iv	3.937	3.837	3.914
E	i	3.909	3.832	3.868
E	ii	3.909	3.851	3.881
E	iii	3.914	3.861	3.887
E	iv	3.911	3.861	3.865

TAB. 5.20 – DAO for unavailable tools.

Qualifications based on the toolset flexibility measure seem to improve the DAO much more than the qualifications based on the WIP flexibility measure. Only in two cases the toolset flexibility was better. Also the maximum tool charges were compared for the same qualification sets (Table 5.21).

Even though the WIP flexibility measure is meant to decrease the maximum tool charge, when a chamber gets unavailable and in many cases, the qualifications based on toolset flexibility are better.

5.3.2.4 Conclusions for etch area tests

Experiments show that qualifications based on WIP flexibility improves significantly both cycle time and throughput of scheduling in an etch workshop. Qualifi-

Unavailable				
tool	chamber	0	TS	WIP
A	i	1814	1333	1752
A	ii	1661	1333	1444
A	iii	1864	1261	1545
A	iv	1547	1634	1422
B	i	1540	1739	1474
B	ii	1557	1651	1474
C	i	1907	1600	1479
C	ii	1907	1600	1479
C	iii	1722	1488	1738
C	iv	1663	1438	1479
D	i	1664	1509	1469
D	ii	1557	1355	1469
D	iii	1946	1794	1559
D	iv	1731	1533	1765
E	i	1738	1501	1524
E	ii	1716	1682	1539
E	iii	1748	1832	1560
E	iv	1700	1832	1514

Tab. 5.21 – Maximum tool charge for unavailable tools.

cations based on time flexibility improve the results even more. The only thing that requires additional consideration is the cluster tool effect. Rather than qualifying a recipe on a chamber for a tool where the recipe is not qualified on any other chamber, the recipe should be qualified on a chamber of a tool with a chamber already qualified. According to the extension for cluster tools mentioned in Section 4.2.4 this can easily be implemented.

Finally, although the effects of unavailable chambers are not so severe in the etch area as for the photolithography area, qualifications based on toolset flexibility also reduce the negative impacts of unavailable chambers.

5.4 Impact of Qualifications on Scheduling in a Diffusion Area

Ten different data instances have been used for the diffusion area. Tools in the diffusion area are identical, i.e. recipes have same throughput on all tools, but two different recipes do not necessarily have the same throughput times. Therefore only qualifications which optimize F^{WIP} have been performed. It is, however, not certain that this will lead to the same solution as when $F^(time)$ is optimized. This is because if all recipes with long process times are qualified on one tool and all recipes with short process times are qualified on another tool, the workload might be different. The fastest recipe to process, for the test instances, takes 3 hours per batch, and the slowest takes 6 hours per batch.

5.4.1 Performance measures

The implemented objectives are the following :

- *Maximization of the number of moves* f_{moves}, i.e. the number of completed operations in number of wafers during the scheduling horizon. f_{moves} is calculated in (5.11) for a set of lots $\mathcal{J} = \{J_i|\ i = 1,..,n\}$ started in the scheduling horizon T, where ω_i is the number of wafers in lot J_i and $\theta_i \in [0, 1]$ is the completion ratio of J_i before the end of the scheduling horizon, i.e. $\theta_i = 1$ if J_i is completed before T.

$$f_{moves} = \sum_{J_i \in \mathcal{J}} \theta_i \omega_i \tag{5.11}$$

- *Maximization of the batching coefficient* f_{batch}, where the batching coefficient is calculated as the total number of moves divided by the sum of the number of completed batches for each tool times the maximum capacity of that tool. The batching coefficient is calculated in (5.12), where $\mathcal{B}^T$ denotes the set of batches completed before the end of the scheduling horizon T, $\mathcal{B}_t^T \subseteq \mathcal{B}^T$ the subset of batches in $\mathcal{B}^T$ sequenced on tool t, $|B|$ the number of lots in batch

B and R_t the maximal number of lots in a batch on tool t.

$$f_{batch} = \frac{\sum_{t=1}^{T} \sum_{B \in \mathcal{B}_t^T} \frac{|B|}{R_t}}{|\mathcal{B}^T|} \tag{5.12}$$

– *Minimization of the X-factor* $f_{X\text{-}factor}$, is a cycle time measure which is equal to the sum of the total time that each lot stays in the work area divided by its process time. The average X-factor of each lot J_i weighted by the priority c_i of J_i is calculated in (5.13) where $\mathcal{J}^T$ is the set of lots completed before the end of the scheduling horizon T, S_i is the start time of J_i and r_i is its arrival time.

$$f_{X\text{-}factor} = \frac{\sum_{J_i \in \mathcal{J}^T} c_i \left(S_i - r_i\right)}{|\mathcal{J}^T|} \tag{5.13}$$

These objectives have been used as performance measures for the numerical experiments.

5.4.2 Numerical experiments

Test instances from ten different time periods have been extracted from data of a real fab. Together with the initial qualifications, tests were run where the qualification (i) optimizes the WIP flexibility has been performed, (ii) a not so good qualification has been performed, (iii) a recipe has been disqualified on a tool and (iv) all possible qualifications have been performed.

5.4.2.1 Number of moves

Table 5.22, shows how the number of moves changes for the different qualifications. Most of the qualifications improve the performance, but the optimal qualification does not always lead to the best results. In one case, Instance 3, it is even better to disqualify a recipe than to qualify the best one. However, on average it is better to qualify a recipe which optimize the WIP flexibility (1.4 % increase on average)

		(i)	(ii)	(iii)	(iv)
Qualification	Initial	optimal	non-optimal	disqualification	all qualified
Instance 1	14769	14780	*14988*	14549	*13822*
Instance 2	11442	11442	11442	11292	11585
Instance 3	14264	*14339*	*14489*	*14414*	*13817*
Instance 4	17498	18119	*18142*	*18058*	18155
Instance 5	17983	18458	18033	17920	18878
Instance 6	17555	17656	17605	*17631*	18655
Instance 7	18987	19041	*18888*	18643	19669
Instance 8	20398	20845	*20876*	19905	*20150*
Instance 9	16120	16367	16145	15770	*16200*
Instance 10	18223	18770	18273	17652	*18320*

TAB. 5.22 – Number of moves for test instances in a diffusion area.

than a qualification which does not optimize the WIP flexibility (0.9 % increase on average) or to disqualify a recipe (0.9 % decrease on average).

Qualifying all recipes does not always seem to be a good solution. In three of the cases, the output is worse when all recipes are qualified on all tools than for the initial case. One possible explanation of that is that the heuristic in the batch optimization solver does not always provide a very good solution. In general, because many recipes only consist of one or a few batches, it is not certain that qualifications on additional tools can improve the performance so much. Moreover, in some cases, it could be that the scheduling tool considers that optimizing the batching coefficient or optimizing the X-factor is more important than minimizing the number of moves.

5.4.2.2 Batching coefficient

In order to analyze the correlation between WIP flexibility and the batching coefficient the value of the batching coefficient is shown in Table 5.23 for the same test instances.

It can be noted that the batching coefficient only decreases for two of the instances when a disqualification is performed. This indicates that the heuristic does not find the optimal solution, since for the initial case the same solution can be used as for when a disqualification has been performed. The optimal qualification (i) is

		(i)	(ii)	(iii)	(iv)
Qualification	Initial	optimal	non-optimal	disqualification	all qualified
Instance 1	0.784	*0.781*	0.783	*0.788*	0.808
Instance 2	0.753	0.753	0.753	0.749	0.762
Instance 3	0.832	0.832	0.832	0.832	*0.820*
Instance 4	0.826	0.838	0.829	*0.832*	*0.833*
Instance 5	0.833	0.833	0.833	*0.839*	0.835
Instance 6	0.846	0.850	0.846	*0.853*	0.850
Instance 7	0.857	0.860	0.857	*0.863*	0.863
Instance 8	0.864	0.867	*0.870*	*0.870*	0.863
Instance 9	0.801	*0.799*	0.801	0.794	0.789
Instance 10	0.812	0.823	0.812	0,812	0.823

TAB. 5.23 – Value of the batching coefficient for the test instances in a diffusion area.

still often the optimal batching coefficient (0.33% on average).

5.4.2.3 X-factor

Finally, a last study regarding the X-factor is performed in Table 5.24. The X-factor is minimized and, for the optimal qualification, the X-factor only increases for Instance 1. The disqualification decreases the X-factor in half of the cases. When qualifying recipes on all tools, in the first instance, the X-factor increases by 10.5% (sic !). This is hard to explain since, for the same instance, the number of moves is decreasing. For Instance 3, the result for all performance measures are worse when qualifying all recipes on all tools than the initial case. It can, however, be observed that a optimal qualification is, in most cases and on average, better than a not so good qualification or a disqualification.

5.4.2.4 Conclusions for tests in the diffusion area

The effects of qualifying recipes on tools based on the flexibility measures for the diffusion area are not always as positive as for the photolithography and etch area. This might be due to the fact that, for the diffusion area, the number of batches

		(i)	(ii)	(iii)	(iv)
Qualification	Initial	optimal	non-optimal	disqualification	all qualified
Instance 1	2.90	*2.92*	2.88	2.93	*3.24*
Instance 2	3.01	3.01	*3.02*	*2.98*	*3.24*
Instance 3	3.56	3.56	3.53	*3.54*	*3.71*
Instance 4	3.89	3.80	3.80	*3.74*	3.79
Instance 5	6.86	6.85	6.86	*6.72*	6.74
Instance 6	3.79	3.72	*3.79*	*3.69*	3.73
Instance 7	6.25	6.19	6.25	6.28	6.14
Instance 8	4.16	4.10	4.15	4.20	4.08
Instance 9	3.87	3.80	3.87	3.97	3.74
Instance 10	2.79	2.78	*2.83*	2.85	*2.91*

TAB. 5.24 – Value of the X-factor for test instances in a diffusion area.

needs to be considered and not the number of wafers. Since the number of batches is small for many recipes, making a qualification for a recipe which only has, for example, five lots may have a negative effect, since the batch scheduler might try to split the batches even if it is not necessary. It should be pointed out the batch optimization solver is based on a heuristic which cannot look through all possible solutions, and hence in general the optimal solution is not found. I.e. even though an additional qualification should at least provide the same solution than without the qualification, the result can be worse in some cases.

5.5 Conclusions

It has been shown that performing just a few additional qualifications may improve scheduling performances. This is possible when recipes are qualified on tools that increase the flexibility of capacity allocation.

We have seen how tool charge, DAO and other performance measures are improved with new qualifications until the WIP flexibility or time flexibility measures are close to 100 percent. Additional qualifications often do not lead to more improvements and might even worsen the performance measures a little. This might happen when scheduling tries to compensate for the different local rules.

A workshop with high toolset flexibility does not necessarily increase the performance of the workshop. It was, however, shown that, even when the time flexibility or WIP flexibility measures are close to 100 percent, increasing the toolset flexibility may improve the robustness of a workshop when tools become unavailable.

Increasing the flexibility measures by performing qualifications clearly improves performances very well for the photolithography and etch areas. For the diffusion area, the impact performances is not as positive. It might depend on the scheduling heuristic which not only tries to optimize the number of moves, but also the number of batches and the X-factor. Theoretically, after an additional qualification, the scheduling tool can still find the same solution than before the qualification, or a better schedule. As observed in the test results this is not always the case. The scheduling heuristic could probably be further improved. Additionally, the number of batches is considered and not the number of wafers (which is the case for the photolithography and etch areas). Since the number of batches is often small, a qualification which increases the flexibility might also negatively impact the schedule of all batches.

It should be noted that the qualification management software does not explicitly consider detailed scheduling considerations such as batch configurations, mask trains and lot priorities. This may significantly influence the scheduling in some cases. One of the goals is to study whether and how these elements should be included when proposing new optimal qualifications.

Different models on how several qualifications can be conducted optimally have been developed. They should be tested to see which of these methods is the most suitable. A relevant method should provide sufficiently good solutions with reasonable computing times.

Also, it may be relevant to adjust the scheduler simulator for photolithography. Our tests have indicated that the scheduling rules play an important role in the results. In particular, the trade-off between the batch configuration rule and the tool charge rule should be tested. It should also be analyzed whether process times should be considered by the tool charge rule.

Chapitre 6

Conclusions and Perspectives

In this last chapter conclusions based on the studies in thesis are presented. From these conclusions perspectives beyond the scoop of this thesis are outlined. The sections in this chapter are structured such that the different parts of the thesis first are summarized in order to better conclude the results of theme and thereafter see if there are perspectives for further research within these areas.

Scientific Context

Firstly, the scientific context of the thesis was introduced. Industrial engineering and semiconductor manufacturing were presented. It was motivated why qualifications in semiconductor manufacturing need to be well managed. Qualifications management (QM) was presented as a way of optimizing capacity allocation. QM has rarely been considered in the literature and it is usually only discussed in general terms without pointing out possible directions for further research. Therefore QM was thoroughly defined, and different challenges within QM were presented. It has been motivated how qualifications of recipes on tools can ease the production flow in the fabrication. In order to find the right qualifications to conduct, flexibility models have been developed. The idea is to increase the flexibility for the operators in semiconductor fabs such that they are able to chose optimal ways to allocate the capacity in the workshops. To do that flexibility measures were defined, that can measure how flexible a system of recipes qualified on a toolset currently is and more importantly how further qualifications can increase the flexibility of the toolset.

It has been shown that QM is an excellent tool for improving flexibility for semi-conductor manufacturing. However, as mentioned, flexibility is only one of several aspects that can be improved by QM. Other areas for example process control could possibly be improved by studying how different qualifications effect the quality of the products. Also further studies should be done in order to see how qualifications and disqualifications can reduce the degree of reentrancy in wafer fabs ; this is so far only a hypothesis, which would probably be mostly useful for fabs where wafers are processed in high volume and in a low mix such that tools can be dedicated to production lines.

Flexibility measures

Four flexibility measures were defined :

- Toolset flexibility
- WIP flexibility
- Time flexibility
- System flexibility

The toolset flexibility measure stresses the importance of having much capacity and therefore, qualifications are favored for recipes where only a few tools already have been qualified and for which have high WIP quantity. The WIP flexibility measure on the other hand considers qualifications of recipes that enables the WIP to be well-balanced on the tools to be important. The time flexibility measure works similarly. However, instead of the actual WIP quantities, it is considered important to have the production times well-balanced on the tools. For the time flexibility measure it also important that the production times are kept minimized. Finally a system flexibility measure has been developed which consider the effects from the other measures. Additionally to the original measures also alternative measures have been developed. Comparing tests were performed in order to understand which of the measures are the most suitable for measuring flexibility.

In the literature some other types of flexibility measures have been considered. These have not explicitly been developed for workshops. Nevertheless other ways to measure flexibility of qualifications in a toolset could still be developed.

Workload balancing

In order to calculate the WIP flexibility measure and the time flexibility measure the optimal way to distribute the WIP quantities on the tools needs to be found. Which way is the optimal for the WIP distribution differs between the two measures. This is due to the fact that the total WIP quantity which is used for the WIP flexibility is constant, whereas the sum of the production times on the tools is variable. Hence, the production times do not only need to be well-balanced on the tools, but they also need to be minimized. For the WIP flexibility measure a new workload balancing method was developed. The method gradually distribute the WIP quantity of a recipe at the time on the tool or tools that currently has the lowest WIP quantity. It has been proved that this method distribute the WIP quantity such that the WIP flexibility measure is optimized. For the time flexibility measure an active set method has been used for the distribution of the WIP such that the time flexibility measure is optimized. At each step, the WIP is redistributed in a optimal *direction*. With direction, in this case, is meant that as much WIP that is added to some of the tools, as much needs to be removed on other equipments. How long step in the direction that should be taken (or how much workload should be redistributed is regulated by the Wolfe condition). Both the methods have been proved to be optimal.

The WIP balancing procedures have been proved to balance the workload in a way such that the maximal values of the WIP flexibility measures respectively the time flexibility measure are found. The solutions are, however, relaxed as a non-integer problem ; i.e. a part of wafer/lot/batch is processed on one tool, at the same time as it is partly processed on another tool. This is a theoretical approximate, but according to the tests in most cases a good practical solution. It should remember that the balancing algorithms are made to find the optimal values for the measures, and that the measures are developed in order to model flexibility for capacity allocation and not to perform dispatching. Nevertheless new balancing algorithms could be developed which optimize the integer balancing problem for whole wafers, lots or batches.

Qualification of several recipes

When more than two qualifications are considered, it is not sure that a greedy approach is optimal for the WIP flexibility or the time flexibility – i.e. to always qualify the recipe which increases the flexibility the most without considering what the following qualifications might be. At the same time to do a complete search of all combination of possible qualifications takes long time. Therefore heuristics have been developed : greedy, local search (two methods), tabu search (two methods) .

The local search methods find the optimal solution more often than the greedy, and when they do not, their solutions are often better than the greedy. The tabu search methods find the optimal solutions in almost all cases. On the other hand the tabu search methods are not as fast as the local search methods.

Furthermore, properties of the flexibilities have been thoroughly studied in order to see what can be further conducted in order to decrease computational times. For the toolset flexibility it has been that it can be very rapidly discovered which recipe needs to be qualified in order to increase the flexibility the most. For the WIP flexibility it has been studied which qualifications increase the flexibility measure. For the time flexibility it is a little bit harder to make predictions which qualifications increase the flexibility. It can, however, be stated that a qualification on tool which is not so heavily charged, on a recipe which throughput time is high on the tool, will increase the flexibility.

The computational times for the tabu search method could probably be improved further by for example implementing tabu lists with varying sizes. Also other heuristics such as the variable neighborhood heuristic could maybe improve the computational performance. Furthermore it is up to the fab management to decide which method they want use. Depending on how long the process engineer can wait for having a solution. Work can be done to reduce the computational times. *E.g.* heuristics can be developed, other ways of finding starting solutions could be studied and and another balancing method than the active set method could reduce the computational times.

Dynamic methods

In a low volume/high mix fab the product mix change rapidly. In order to create sustainable qualifications, the WIP quantities for the coming periods need to be anticipated. Two models, which consider how the WIP quantities change over time have been developed.

In both models weights for the different periods are given. Periods that are considered to be much affected by the qualification are weighed with higher values than less influential periods.

For the first model the WIP quantities of the different periods are weighed by the weights and than added together. The new WIP quantities for the recipes are thereafter used to calculate the flexibility as in the original model.

In the second model the flexibility measures for each period is calculated. Thereafter the flexibility measures are weighed with their weights and added together as one measure.

The first measure has the advantage of being rapidly calculated, whereas the second measure could said to be more robust by considering the flexibility for all periods.

Theoretically the models also could suggest additional qualifications at the beginning of each period. However, it has been decided to only consider qualifications at the current period, since the WIP quantities anyway changes so much over time.

Extensions

In reality some qualifications are harder to conduct than others. A recipe might just temporarily be down and a simple execution in a computer program could be enough to re-qualify the recipe in the instant of a second. For another recipe hours of tests need to be conducted before the recipe can be qualified. For yet other recipes reconfigurations on the tools need to be performed – changes of metal compositions and gas pressure – and the recipe might not be qualified until a week later. Thus recipes could be arranged into different easiness levels. If the qualifications are sorted in these different levels the model can decide to perform a certain number of qualifications from each levels.

When a recipe is qualified on a tool other it will be easier to qualify recipes belonging to the same group of recipes additional tests which are the same for both recipes do not needed to be performed. The recipe groups have been considered in the model such that before a recipe is qualified the group need to be qualified. As with the easiness levels, also the number of qualifications of groups are limited.

Also the characteristics of cluster tools have been considered. It is considered better to qualify a recipe on chamber on a tool where already other chambers have been qualified on the tool. Therefore an extension has been implemented, where tools need to be qualified.

Statistics on recipe availability on the tools are kept by the wafer fabs. Since it is better to qualify a recipe on tool where it is more likely to stay available this statistic has been considered by the toolset flexibility. With the new toolset flexibility measure it is not only possible to see which recipe should be qualified in order to increase the flexibility the most, but also on which tool the qualification should be carried out on.

In order to use the extensions to the model, still some definitions have to be made and statistic information need to be retrieved ; a project needs to be initiated to define if a qualification is easy, medium hard or hard to be conducted, or if further levels are needed. It also need to be studied what recipes can be grouped together. It also needs to be studied if statistics on the tool available can be obtained for the recipes. As long as the information is no available, the model considers all recipes to be easy to qualify, that they belong to one group and that the availability is at 100 per cent.

The model does not consider that different jobs need different setup times depending on what job was processed before on the tool. Further studies could consider how setup times affect flexibility and if setup times could be integrated in the model.

Scheduling

Three scheduling simulators for workshops in different work areas have been used in order to see what impact qualifications might have on scheduling : Photolithography area, etch area, diffusion area. The tests with the simulators have proved that qualifications based on the WIP flexibility measure and the time flexibility measure decrease production times and cycle times well in the photolithography and etch area.

If instead the toolset flexibility measure is used for recommending further qualifications the toolset get more robust concerning the times when a tool becomes unavailable for processing. With the system flexibility, measure it is possible to consider both these effects.

The tests for the photolithography and etch are have shown very good results, and it seems that the measures in most cases serves their purpose. It is thus possible to perform optimal qualifications which optimize the performance in the workshops (and possible the whole fab) and make the production less sensitive when tool becomes unavailable.

In the diffusion area the average results of several instances is quite good. But in many of the instances an optimal solution cannot be assured but a qualification which optimizes the flexibility. The batch optimizing solver for the diffusion is based on a heuristic and it is not sure that it finds the optimal solution in all cases.

Tests have so far only considered how the performance can be improved within individual workshops. But since local optimization not always is optimal on global level, it can be researched how flexible qualifications in the workshops effect the total factory output, by simulating scheduling for a whole fab.

The tests have shown that the flexibility is a very good mean to improve performance in wafer fab workshops such the cycly time, througput, but also in order to make workshops more robust cornening unavailability of tools. However, it should be remember that flexibility is not an exhaustive tool. The tests have also shown that flexibility does not cover all the complexity of semiconductor manufacturing. Batch optimization, global optimization are not directly concerned by flexibility. As discussed in Chapter 1 other objectives such as improving APC and reducing reentrancy could also be the objective of qualification management.

Implementation

A software has been developed, for the Crolles 300-site, that constantly updates process/tool statuses. Using this program, the fab management could have better control of the qualified units in the fab, engineers could see where further qualifications need to be conducted and operators could clearly visualize where processes can be carried out. At the moment no decision has been taken to use the software in production at the Crolles300 site. Nevertheless tests have been started in order to see if the flexibility model could be used at ST's Rousset site, also ST's Crolles200 site has shown interest in implementing the software in their fab.

Annexe A

Dictionary

A.1 Dictionary

Below a small dictionary for expressions used in this thesis be found. For a more extensive dictionary on terms and expressions used in the semiconductor industry internet site *The Semiconductor Glossary*[1] should be refered to.

- batch – Lots requiring the same recipe, that are grouped together, such that they can be processed at the same time.
- block – A block or batch train is lots, requiring the same recipe, that are grouped together and thereafter processed directly after one another in order to reduce the set up before the process of each lot.
- bottleneck – A place in the production chain where the capacity is limited such that the capacity is reduced in the whole production chain.
- cluster tool – a tool with several processing chambers. On dry etch cluster tools the chambers can be used for parallel processing. In diffusion cluster tools are used for sequential processing.
- cycle time – The time a wafer or a lot stays in a work area or the entire fab.
- etch – A process where photolithography patterns are removed (dry etch) or bulk is removed (wet etch) from the wafers. Throughout this thesis dry etch tools have been regarded. Dry etch tools are usually cluster tools.

[1] http ://www.semiconductorglossary.com/

– fab - A semiconductor *fab*rication plant is the factory where the integrated circuit are produced on silicon wafers.
– hold constraint – The reason why a qualification is set to be non processable : The initial tests have not been performed, the tool or a chamber cannot fulfill the required quality of the process etc.
– integrated circuit (IC) – An electronic curcuit built on a single piece of substrate (typically silicon).
– lot - A lot contains places for 25 wafers (for 300mm wafer fabrication), which are used to transport the wafers inside the fab.
– photolithography – A method using UV radiation which prints patterns on the integrated circuits. The photolithography tools are very expensive, and the fab managements tries to minimize the volume of these tools, which often results in bottle necks around the photolithography workshops.
– qualification - To have a recipe qualified on a tool means that all instructions and settings for the process have been defined and approved such that the recipe can be used for processing wafers on the tool.
– recipe – The instructions and settings required to be defined for a process on a tool. In order to perform a process on a tool, its recipe need to be qualified on the tool.
– tool – Most of the times tools, equipments and machines are used as synonyms in the semiconductor industry. What is meant in all cases is the unit which performs processes.
– toolset – A group of tools in a workshop that can perform the same or similar kinds of recipes.
– throughput time - The production speed (of a recipe) on a tool. Normally measured in WIP quantity per hour.
– wafer – A thin circular plate on which the intergrated circuits are produced on.
– WIP – Work-in-progress or work-in-process – the jobs that are awaiting to be processed.
– WIP quantity – The amont of work that awaits to be process. In semiconductor manufacturing this amont is mostly calculated in numbers of wafers.
– work area – An area consisting of different workshops which together cover an

operation.

- workshop – The tools which are used for conducting a certain step for the production.

Annexe B

Pseudo-Codes

B.1 Pseudo-code : Active Set method

1. Let x_k be a feasible solution
2. If $Z^T \nabla \left(T\gamma\left(x_k\right)\right) = 0$ Then
   (a) If $\bar{\lambda} \geq 0$ Then goto 1. for recipe $r+1$
   (b) Else for $j = argmin\left\{\lambda_i\right\}$ drop constraint j
3. $p = -Z\left(Z^T \nabla^2 \left(T^\gamma\right) Z\right)^{-1} Z^T \nabla \left(T^\gamma\right)$
4. $\alpha_{min} = min\left\{1, \frac{WIP_i}{-p_i} : p_i < 0\right\} i\forall$ inactive constraints
5. $x_{k+1} = x_k + \alpha_{min} p$
6. If $\alpha_{min} < 1$ activate constraint $j = argmin\left\{\alpha_j\right\}$
   (a) If $\alpha_{min} = 0$ add all constraints t s.t. $WIP_t = 0$.
7. Goto step 2.

Annexe C

Implementations

To practically perform the tests and simulations of the models in this thesis two software have been programmed. The software are also programmed such that they also could be used by operators and engineers for optimizing the production. The first software is the QM software which proposes on which tools recipes should be qualified on in a toolset in order to increase the flexibility of the workshop. The second software is the scheduling simulator for the etch area, where the rules described in Chapter 5 have been implemented.

C.1 The Qualification Management Software

First, a software was programmed that shows the process statuses of recipe on the tools in a work area. The software has been programmed in VBA (Visual Basics Applications) and its output is displayed in MS Excel. VBA for MS Excel was chosen as software by the company, since it has an integrated development environment which can be easily managed directly from MS Excel. Furthermore, most users are familiar with the MS Excel environment. A first version of the software which only showed the process statues of the recipes on the tools where developed during two months in the begining of the thesis. When this version was put in the used for the production, it was decided that a new version in Java should be developed.

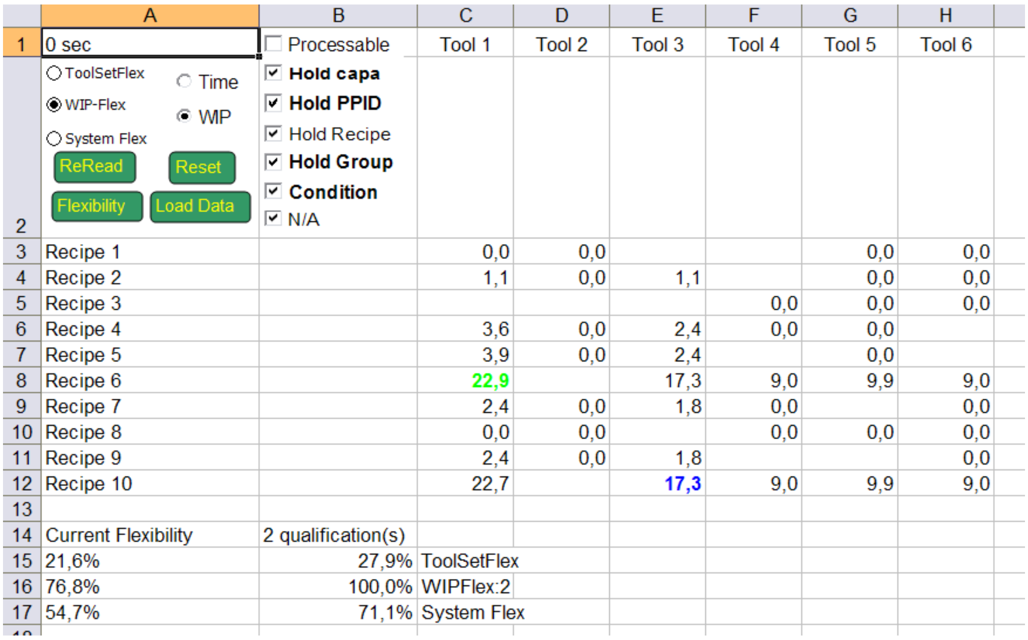
A screen shot of the software is shown in Figure C.1. The recipe names are displayed in the right column, and the tool names are shown in the first row. Below the recipes the current values of the flexibility measures are displayed. For the example in the figure the toolset flexibility is 21.6%, the WIP flexibility is 76.8% and the system flexibility is 54.7%. Directly to the left of the current flexibility values the flexibility measures after k number of qualifications can be seen. How many qualifications k that are recommended, can be chosen by the user. In the figure, two qualifications have been chosen. The flexibility measures would increase to 27.9% for the toolset flexibility, 100% for the WIP flexibility and 71.1% for the system flexibility. In the upper left corner of the figure it can be seen how the user can choose if the qualifications that increase the toolset flexibility, the WIP flexibility, the time flexibility or system flexibility should be chosen. In the example WIP flexibility has been chosen.

In the intersection of recipe row and tool column different values are displayed. These values indicate how much a qualification of the recipe on this tool would increase the flexibility (from the flexibility type that has been chosen). For the case displayed in the figure, qualifying recipe 2 on tool 1 increases the flexibility with 1.1 percentage points. Qualifications of recipes on tools that have already been performed are left empty in the figure. It is, however, possible for the user to display how much a potential disqualification would decrease the flexibility on these places.

Two of the qualifications are highlighted : recipe 6 on tool 1 is highlighted in green, and recipe 12 on tool 3 is highlighted in blue. The qualification highlighted in green signifies that this is the qualification which will currently increase the flexibility measure the most. Qualifications highlighted in blue are the x qualifications that together will increase the flexibility the most. If a qualification that is already highlighted in green, also is included among the x qualifications that increases the flexibility the most, the color green is kept in favor of the blue color.

C.1.1 Extensions

It is possible to chose if the software should consider qualifications with a certain process status or not. For example it is possible to exclude all the possible qualifications of recipes on tools where the status is *N/A*. If the option box for processable



	A	B	C	D	E	F	G	H
1	0 sec	Processable	Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6
2	ToolSetFlex, Time, WIP-Flex, WIP, System Flex, ReRead, Reset, Flexibility, Load Data	Hold capa, Hold PPID, Hold Recipe, Hold Group, Condition, N/A						
3	Recipe 1		0,0	0,0			0,0	0,0
4	Recipe 2		1,1	0,0	1,1		0,0	0,0
5	Recipe 3					0,0	0,0	0,0
6	Recipe 4		3,6	0,0	2,4	0,0	0,0	
7	Recipe 5		3,9	0,0	2,4		0,0	
8	Recipe 6		22,9		17,3	9,0	9,9	9,0
9	Recipe 7		2,4	0,0	1,8	0,0		0,0
10	Recipe 8		0,0	0,0		0,0	0,0	0,0
11	Recipe 9		2,4	0,0	1,8			0,0
12	Recipe 10		22,7		17,3	9,0	9,9	9,0
13								
14	Current Flexibility	2 qualification(s)						
15	21,6%	27,9%	ToolSetFlex					
16	76,8%	100,0%	WIPFlex:2					
17	54,7%	71,1%	System Flex					

FIG. C.1 – The flexibility software

recipes is deselected only possible qualifications would be considered and no disqualifications would be considered. Figure C.2 shows a picture of the option check boxes where process statuses can be selected or deselected.

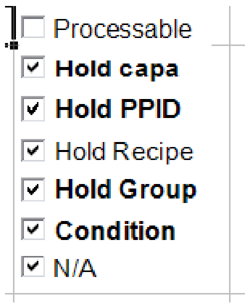


FIG. C.2 – Enabling process statuses

The qualifications can be categorized in different easiness levels (e.g. easy, medium, hard) as mentioned in Chapter 4. This has to be entered by the operator before the calculations are effectuated. It is possible for the operator to choose the number of optimal qualifications from each level that should be found. For example the operator could tell the software to find three easy qualifications, one medium qualifications and one hard qualification that will increase the flexibility the most.

Recipes belong to different groups with similar characteristics. In reality when one recipe of the group has been qualified on a tool it is much easier to qualify the rest of the recipes of the group on that tool. In the software it has been implemented such that in order to qualify a recipe the underlying group first need to be qualified.

It is possible to retrieve information about lots which will arrive at the workshops during the following days and their estimated arrival. This is used for the dynamic models described in Chapter 4.

With statistics from the availability of the recipes on the tools, the software may also use the toolset flexibility model with integrated recipe/tool availability described in Chapter 4.

The information about easiness levels, groups and recipe/tool availability is not available at the fab today. Until this information can be obtained all qualifications are considered to be easy.

In Figure C.3, the parameter sheet of the software is shown. On the first row it is possible it is displayed if the time flexibility or WIP flexibility measure has been chosen. *Faux* (false in French) siginifies that the WIP flexibility measure has been chosen. *Vrai* (true in French) signigies that the time flexibility has been chosen. On the second row the number of units in the system is stated. A tool without chambers is counted as one unit, whereas for a tool with processing chambers, only the chambers are counted as units. On row 3 the number of recipes are stated. On row 4 it is shown which flexiblity type has been chosen : toolset flexibility, time flexibiltiy, WIP flexibility and system flexibility On rows 7 and 8, it is stated which recipe statuses that will be displayed. The number of time periods can be chosen on row 9. On row 10 and 11 it is possible to choose how many qualifications that should be considered. It is possible to decide which easiness levels the qualifications should be made at, and how many groups that can be qualified. On row 12 it is possible to decide which value γ will have in the calculations of the WIP flexibility and time flexibility measures.

The value on row 13 allows to decide how much of the toolset flexibility measure respectively the WIP flexiblity/time flexibility measure should be included in the system flexibility measure. The value 0.4 indicates that $F^{sys} = 0.6 \cdot F^{TS} + 0.4 \cdot F^{y}$ where y indicates if the time flexibility or the WIP flexibility has been chosen on row 1.

The parameter on row 15 allows to chose if the active set method will be used or not, and on row 16 a parameter allows to adjust how exact the active set method should be. On row 22 it is possible to chose which heuristic should be used. On row 24 the type dynamic method can be chosen. On rows 25, 26 and 27 it is possible to control how the tabu search methods should be run.

1	Time/WIP-Button	FAUX					
2	Number of Units	6					
3	Number of Recipes	10					
4	FlexChoice	2					
5							
6							
7	Processable	Capa	PPID	Recipe	Group	Condition	N/A
8	FAUX	VRAI	VRAI	VRAI	VRAI	VRAI	VRAI
9	Number of Periods	4					
10		Easy	Medium	Hard	Group		
11	Nb of Qualifications	2	0	0	5		
12	gamma (>1.1)	2	(High value will make the balance more important / low value will decrease total throughput time)				
13	flexWeight [0,1]	0,4	(Low value will increase the importance of the Toolset Flexibility)				
14							
15	Active Set	VRAI					
16	alpha_max	20					
17							
18							
19							
20							
21							
22	Method	4	(1=Greedy, 2=Alternate, 3=Local Step, 4=InterChange, 5=Tabu1, 6=Tabu2)				
23		1					
24	MultiPeriods	1	(0=none, 1=sumWip, 2=sumFlex)				
25	Tabu iterations	50					
26	Tabu list size	20					
27	Dynamic list size	5	FAUX				
28							

FIG. C.3 – The Software's parameter sheet

C.1.2 Calculating the flexibility measures

Knowing the recipe statuses and their WIP quantities it is possible for the software to calculate the flexiblity measures. It is considered if a recipe is processable on a unit or not and the WIP quantities for each recipes. As a unit is considered the chambers of the cluster tools and the tool it self if the tool does not have any chambers. This information is sufficient for the software in order to calculate the toolset flexibility measure and the WIP flexibility measure (and the system flexibility measure in cases where the time flexibility measure is not considered). In order to calculate the time flexibility measure throughput times for each recipe on each unit is needed. This information is received from an external file. However, throughput times are not available for all recipes on all units. When this is the case the recipe for this unit is considered as non-processable, and is not considered when calculating the the time flexibility.

To calculate the WIP flexibility measure and the time flexibility measure the software uses the two workload balancing procedures described in Chapter 2. As mentioned the flexibility measures for the current setting with processable and non-processable units for recipes are calculated and displayed by the software. The software also calculates the flexibility for all cases if an additionally qualification of a non-processable recipe on a unit would be conducted. Also the disqualification of a processable to a non-processable recipe may be calculated. On the screen (see Figure C.1) it is possible to see the difference of the current flexibility measure and the flexibility measure if the recipe would be qualified on the unit.

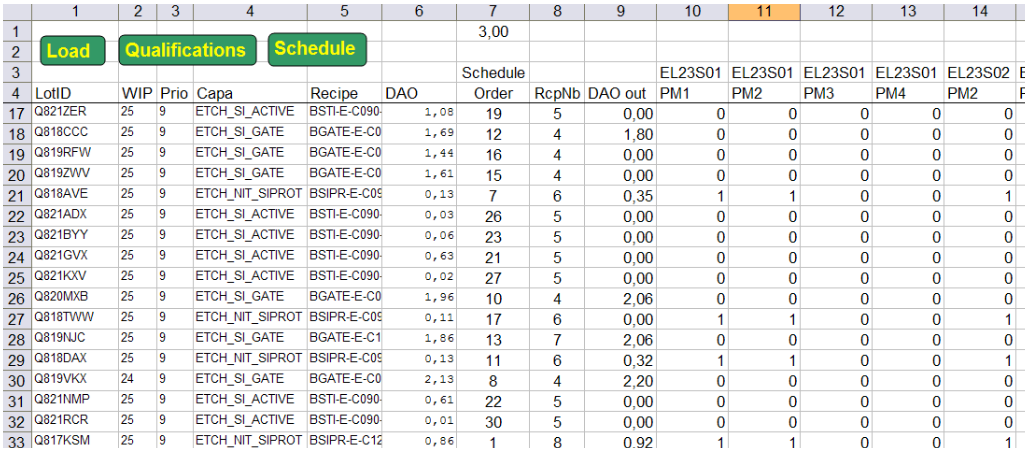
Additionally it is possible for the user to set the number of optimal qualifications that the software has to calculate. The software then finds the qualifications that together will increase the flexibility the most according to one of the heuristics mentioned in Chapter 3. Depending on which heuristic is chosen the likelihood of finding the qualifications which are really optimal and the computing times differs.

C.2 Scheduling Simulator for the Etch Area

Also a software which simulates scheduling for an etch work area have been implemented. With this software it is possible to see if additional qualifications in the etch area will impact the scheduling and in the end improve the performance in the etch area. The etch tools are characterized as cluster tools where parallel processing may occur simultaneously on the different chambers on a tool. The simulator has been programmed in VBA and the output is displayed in MS Excel.

C.2.1 Input

As input the software takes the WIP list for the toolset, i.e. all lots that are ready to be processed in the toolset. For each lot the WIP list contains data about number of wafers (WIP quantity), PIT, DAO, required recipe and required capability. Thereafter a second input file is ready with information on what tools the recipes are processable. It is possible for the user to simulate a qualification of a recipe on a chamber by hand so that the recipe is regarded as processable on that tool. It is also possible to do disqualifications so that the recipes are no longer processable on a tool. In that way it is possible to see how qualifications may impact the scheduling. Also the throughput times of the recipes on the tools are loaded from an external file. Figure C.4 shows a picture on the sheet where the input data has been loaded.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1							3,00								
2	Load	Qualifications			Schedule										
3							Schedule			EL23S01	EL23S01	EL23S01	EL23S01	EL23S02	E
4	LotID	WIP	Prio	Capa	Recipe	DAO	Order	RcpNb	DAO out	PM1	PM2	PM3	PM4	PM2	F
17	Q821ZER	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	1,08	19	5	0,00	0	0	0	0	0	
18	Q818CCC	25	9	ETCH_SI_GATE	BGATE-E-C0	1,69	12	4	1,80	0	0	0	0	0	
19	Q819RFW	25	9	ETCH_SI_GATE	BGATE-E-C0	1,44	16	4	0,00	0	0	0	0	0	
20	Q819ZWV	25	9	ETCH_SI_GATE	BGATE-E-C0	1,61	15	4	0,00	0	0	0	0	0	
21	Q818AVE	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,13	7	6	0,35	1	1	0	0	1	
22	Q821ADX	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,03	26	5	0,00	0	0	0	0	0	
23	Q821BYY	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,06	23	5	0,00	0	0	0	0	0	
24	Q821GVX	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,63	21	5	0,00	0	0	0	0	0	
25	Q821KXV	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,02	27	5	0,00	0	0	0	0	0	
26	Q820MXB	25	9	ETCH_SI_GATE	BGATE-E-C0	1,96	10	4	2,06	0	0	0	0	0	
27	Q818TWW	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,11	17	6	0,00	1	1	0	0	1	
28	Q819NJC	25	9	ETCH_SI_GATE	BGATE-E-C1	1,86	13	7	2,06	0	0	0	0	0	
29	Q818DAX	25	9	ETCH_NIT_SIPROT	BSIPR-E-C09	0,13	11	6	0,32	1	1	0	0	1	
30	Q819VKX	24	9	ETCH_SI_GATE	BGATE-E-C0	2,13	8	4	2,20	0	0	0	0	0	
31	Q821NMP	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,61	22	5	0,00	0	0	0	0	0	
32	Q821RCR	25	9	ETCH_SI_ACTIVE	BSTI-E-C090-	0,01	30	5	0,00	0	0	0	0	0	
33	Q817KSM	25	9	ETCH_NIT_SIPROT	BSIPR-E-C12	0,86	1	8	0.92	1	1	0	0	1	

Fig. C.4 – Input data sheet for the simulator

By entering values on the data sheet in the MS Excel user interface it is possible to adjust the values of the scheduling weight parameters and decide during how long the scheduling should take place.

C.2.2 Scheduling

The scheduling is started by checking which lots are processable on the toolset. All lots that are not processable are excluded from the simulation. For the remag lots. A new lot is dispatch to a tool when at least one load port of the tool is available.

The rules described in Chapter 5 decide which of the lots should be dispatch and on what chambers (read PPID) should be used for the processing. When all the tools have been dispatched a full schedule for the toolset is achieved.

For the internal scheduling inside the tool the load locks and the handler have been ignored. Instead a wafer at the time is processed on the chamber until there is no more wafer to process in the lot. The wafers of another lot may not be processed on a chamber until the lot which was dispatched before has stopped using the chamber for processing.

C.2.3 Output

The output (Figure C.5) shows which order the lots are dispatch on each tool and which chambers they use. It is possible to see how long each process have lasted, the identification code of the lot or when the lot will be ready on the tool. In Figure C.5 the first lot on the first tool uses chambers PM1 and PM2 and will be ready after 95 resp. 90 minutes. The second lot on the same tool will be ready by 185 minutes. Lastly the values of performance measures are displayed on a separate worksheet.

	EL23S_1				EL23S_2		EL23S_3				EL23S_4				EL23S_5			
Lot order	PM1	PM2	PM3	PM4	PM2	PM3	PM1	PM2	PM3	PM4	PM1	PM2	PM3	PM4	PM1	PM2	PM3	PM4
1	95	90			155		45	40		40			95	90			40	35
2	185	185			310		100	100	70	95			185	185			130	130
3	280	275					160	160	135	155			280	275			225	220
4																		
5																		
6																		
7																		

FIG. C.5 – A ready schedule

Bibliographie

[1] Aardal, K., Pochet, Y., and Wolsey, L. Capacitated facility location : Valid inequalities and facets. *Mathematics of Operations Research 20* (August 1995), 562–582.

[2] Achacoso, J., and Pisapia, G. Partnering for the integration of equipment recipe services into an automated factory. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (Cambridge, MA, USA, November 1995), pp. 101–104.

[3] Adams, J., Balas, E., and Zawack, D. The shifting bottleneck procedure for job shop scheduling. *Manufacturing Science 34* (1988), 391–401.

[4] Akşin, Z., Karaesmen, F., and Örmeci, L. On the interaction between resource flexibility and flexibility structures. In *Fifth international conference on analysis of manufacturing systems* (Greece, May 2005).

[5] Arfken, G. *Lagrange Multipliers.* Academic Press, Orlando, FL, USA, 1985, ch. 17.6, pp. 945–950.

[6] Armijo, L. Minimization of function having Lipschitz continuous first partial derivates. *Pacific Journal of Mathematics 16* (1966), 1–3.

[7] Artigues, C., Dauzère-Pérès, S., Derreumaux, A., Sibille, O., and Yugma, C. A batch optimization solver for diffusion area scheduling in semiconductor manufacturing. In *IFAC Symposium on Information Control Problems in Manufacturing* (Saint-Etienne, France, May 2006), vol. 2, pp. 733–738.

[8] Aubry, A., Espinouse, M.-L., Jacomino, M., and Raison, B. About configuration under uncertainty of a power distribution network. In *INCOM' 2006* (Saint-Etienne, France, May 2006), pp. 383–388.

[9] Aubry, A., Rossi, A., Espinouse, M.-L., and Jacomino. Minimizing setup costs for parallel multi-purpose machines under load-balancing constraint. *European Journal of Operational Research* (2008), 1115–1125.

[10] Balas, E., Lenstra, E., and Vazacopoulos, A. The one machine problem with delayed precedence constraints and tis use in job shop scheduling. *Management Science 41* (1995), 94–109.

[11] Barna, G. APC in the semiconductor industry, history and near term prognosis. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (November 1996), pp. 364–369.

[12] Ben Tal, A., and Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming 88* (2000), 411–424.

[13] Ben Tal, A., and Nemirovski, A. Robust optimization - methodoly and applications. *Mathematical Programming 92* (2002), 453–480.

[14] Bertsimas, D., and Sim, M. The price of robustness. *Operations Research 52* (2002), 35–53.

[15] Bertsimas, D., and Sim, M. Robust optimization - methodoly and applications. *Mathematical Programming 98* (2003), 49–71.

[16] Błażewicz, J., Ecker, K., Pesch, E., Schmidt, G., and Węglarz, J. *Handbook on scheduling.* Springer, Heidelberg, Germany, 2007.

[17] Browne, J. Classification of flexible manufacturing systems. *The FMS Magazine* (1984), 114–117.

[18] Brucker, P., Hurink, J., Jurisch, B., and Wöstmann. A branch and bound algorithm for the open shop problem. *Discrete Applied Mathematics 17* (1997), 43–59.

[19] Bureau, M., Dauzère-Pérès, S., Yugma, C., Vermarien, L., and Maria, J.-B. Simulation results and formalism for global-local scheduling in se-

miconductor manufacturing facilities. In *2007 Winter Simulation Conference* (Washington, DC, USA, 2007), pp. 1768–1773.

[20] Das, S. The measurement of flexibility in manufacturing systems. *The International Journal of Flexible Manufacturing Systems* (1996), 67–93.

[21] Dauzère-Pérès, S., and Lasserre, J.-B. A modified shifting bottleneck procedure for job shop scheduling. *International Journal of Production Research* (1993), 923–932.

[22] de Ron, A., and Rooda, J. Equipment effectiveness : OEE revisited. *IEEE Transaction on semiconductor manufacturing 18* (February 2005), 190–196.

[23] De Toni, A., and Tonchia, S. Manufacturing flexibility : a literature review. *International Journal of Production Research* (1998), 1587–1617.

[24] De Toni, A., and Tonchia, S. Performance measurement systems : Models, characteristics and measures. *International Journal of Operations & Production Management* (2001), 46–71.

[25] Deffree, S. July chip sales grow as 300mm crossover continues. *Electronics Design, Strategy, News* (September 2008).

[26] Ding, S., Yi, J., and Zhang, M. Multicluster tools scheduling : An integrated event graph and network model approach. *IEEE Transactions on Semiconductor Manufacturing 19* (2006), 339–351.

[27] Ford, H. *My life and work.* Deitroit, MI, USA, 1922.

[28] Fuchs, M., Muscogiuri, C., Niederée, C., and Hemmje, M. An open framework for integrated qualification management portals. In *13th International Workshop on Database and Expert Systems Applications* (2002), pp. 1–5.

[29] Garey, M., and Johnson, D. *Computers and Intractability : A guide to the Theory of NP-completeness.* Freeman, San Francisco, CA, USA, 1979.

[30] Garey, M., Johnson, D., and Sethi, R. The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research 1* (1976), 117–129.

[31] Giachetti, E., Martinez, L., Sáenz, O., and Chen, C.-S. Analysis of the structural measures of flexibility and agility using a measurement theoretical framework. *International Journal of Production Economics 1* (2002), 47–62.

[32] Glover, F. Tabu search – part I. *ORSA Journal on Computing 1* (1989), 190–206.

[33] Glover, F. Tabu search – part II. *ORSA Journal on Computing 1* (1990), 4–32.

[34] Glover, F., Taillard, E., and de Werra, D. A user's guide to tabu search. *Annals of Operations Research* (1993), 3–28.

[35] Goodwin, F. Trends in the costs of photolithography development and outlook for the future. Tech. rep., Infineon, Albany, USA, October 2005.

[36] Gorton, W. The genesis of the transistor. *Proceedings of the IEEE 86* (1998), 50–52.

[37] Graves, S., and Tomlin, B. Process flexibility in supply chains. *Management Science 7* (2003), 907–919.

[38] Gueret, C., Jussien, N., and Prins, C. Using intellegent backtracking to improve branch and bound methods : An application to open shop problems. *European Journal of Operational Research 127* (2000), 344–354.

[39] Habenicht, I., and Mönch, L. A finite-capacity beam-search algorithm for production scheduling in semiconductor manufacturing. In *Proceedings of the 2002 Winter Simulation Conference* (San Diego, CA, USA, 2002), pp. 1406–1413.

[40] Hachman, M. Intel, Samsung, TSMC plan shift to 450 mm wafers. *PC Magazine* (May 2008).

[41] Hand, A. ISMI updates goals, challenges of 450 mm wafers. *Semiconductor International* (January 2008).

[42] Hansen, P., and Mladenović, N. Variable neighborhood search for the p-median. *Location Science* (1997), 207–226.

[43] Hofkamp, A., and Rooda, J. Chi reference manual. Tech. rep., TU Eindhoven, November 2002.

[44] Hopp, W., and Spearman, M. *Factory Physics : Foundations of Manufacturing Management.* McGraw-Hill, London, UK, 2001.

[45] Hoyer, W. Variants of the reduced Newton method for nonlinear equality constrained optimiwation problems. *Optimization 17* (1986), 757–774.

[46] Ignizio, J. The exploitation of manufacturing's 3rd and 4th dimensions. In *Third ISMI Symposium on Manufacturing Effectiveness* (Austin, USA, October 2006).

[47] Imai, M. *Kaizen : The key to Japan's competetive success.* McGraw-Hill, New York, NY, USA, 1986.

[48] Jacobs, J., Etman, L., van Campen, E., and Rooda, J. Characterization of operational time variability using effective process time. *IEEE Transactions on Semiconductor manufacturing 16* (2003), 511–520.

[49] Johnzén, C. Robust optimisation of the capcitated location problem. Master's thesis, KTH, Stockholm, Sweden, September 2005.

[50] Johnzén, C., Dauzère-Pérès, S., and Vialletelle. Flexibility measures for qualifications management in wafer fabs. *Production Planning & Control* (2009), in revision.

[51] Johnzén, C., Dauzère-Pérès, S., and Vialletelle, P. Flexibility measures for machine qualification management. In *Roadef* (Grenoble, France, February 2007), pp. 249–250.

[52] Johnzén, C., Dauzère-Pérès, S., and Vialletelle, P. Flexibility models for capacity allocation in semiconductor fabrication. In *MOSIM'08* (Paris, France, April 2008).

[53] Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., and Yugma, C. Importance of qualification management for wafer fabs. In *Advanced Semiconductor Manufacturing Conference* (Stresa, Italy, June 2007), pp. 166–169.

[54] Johnzén, C., Dauzère-Pérès, S., Vialletelle, P., Yugma, C., and Derreumaux, A. Impact of qualification management on scheduling in semiconductor manufacturing. In *Winter Simulation Conference* (Miami, FL, USA, December 2008).

[55] Jordan, W., and Graves, S. Principles on the benefits of manufacturing process flexibility. *Management Sciences 41* (April 1995), 577–594.

[56] Kanellos, M. New life for Moore's law. *c/net* (April 2005).

[57] Kariv, O., and Hakimi, S. An algorithmic approach to network location problems. part 1 : The p-centers. *SIAM Journal on Applied Mathematics* (1979), 513–538.

[58] Kariv, O., and Hakimi, S. An algorithmic approach to network location problems. part 2 : The p-medians. *SIAM Journal on Applied Mathematics* (1979), 539–560.

[59] Karmakar, N. A new polynomial time algorithm for linear programming. *Combinatorica 4* (1984), 373–395.

[60] Kouvelis, P., and Yu, G. *Robust Discrete Optimization and its Applications.* Springer, 1997.

[61] Lai, S.-M., and Hui, C.-W. Measurement of plant flexibility. In *17th European Symposium on Computer Aided Process Engineering* (Bucharest, Romania, 2007).

[62] Lawler, E., and Labetoulle, J. On preemptive scheduling of unrelated parallel processors by linear programming. *Journal of the Association for Computing Machinery 42* (1978), 48–64.

[63] Leachman, R., and Hodges, D. Benchmarking semiconductor manufacturing. *IEEE International Integrated Reliability Workshop Final Reports* (October 1997), 1–6.

[64] Lenstra, J., and Rinnooy Kan, A. Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics 4* (1979), 121–140.

[65] Mahoney, R. *High-mix low-volume manufacturing.* Prentice Hall, Upper Saddle River, USA, 1997.

[66] Mason, S., Fowler, J., and Carlyle, W. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling 5* (2002).

[67] Megiddo, N., and Supowit, K. On the complexity of some common geometric location problems. *SIAM Journal on Computing 13* (February 1984), 182–196.

[68] Mehrotra, S. "on the implementation of a primal-dual interior point method. *SIAM Journal on Optimization 2* (1992), 575–601.

[69] Mladenović, N., Labbé, M., and Hansen, P. Solving the p-center problem with tabu search and variable neighborhood search. *Networks* (2003), 207–226.

[70] Mönch, L., and Driessel, R. A distributed shifting bottleneck heuristic for complex job shops. *Computers and Industrial Engineering 49* (2005), 672–680.

[71] Mönch, L., Schabacker, R., Pabst, D., and Fowler, J. Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. *European Journal of Operational Research* (2007), 2100–2118.

[72] Mönch, L., Stehli, M., Zimmermann, J., and Habenicht, I. The FABMAS multi-agent-system prototype for production control of wafer fabs : design, implementation and performance assessment. *Production Planning & Control 17* (October 2006), 701–716.

[73] Montoya Torres, J. *Transport automatisé dans les systèmes de fabricatin de semi-conducteurs.* PhD thesis, EMSE, Gardanne, France, November 2005.

[74] Montoya Torres, J. A literature survey on design approaches and operational issues of automated wafer-transport systems for wafer fabs. *Production, Planning & Control 17* (October 2006), 648–663.

[75] Montoya Torres, J. Manufacturing performance evaluation in wafer semiconductor factories. *International Journal of Productivity and Performance Management 55* (2006), 300–310.

[76] Moore, G. Cramming more components onto integrated circuits. *Electronics 38* (1965).

[77] Moyne, J., del Castillo, E., and Hurwitz, A. *Run-to-Run Control in Semiconductor Manufacturing.* CRC Press, Boca Raton, FA, USA, 2001.

[78] Moyne, J., Solakhian, V., Yershow, A., Anderson, M., and Mockler Hebert, D. Development and deployment of a multi-component advanced process control system for an epitaxy tool. In *ASMC* (Boston, MA, USA, April/May 2002), pp. 125–130.

[79] Nakajima, S. *Introduction to TPM : Total Productive Maintenance.* Productivity Press, Cambridge, MA, USA, 1988.

[80] Nash, S., and Sofer, A. *Linear and Nonlinear Programming.* McGraw-Hill, New York, NY, USA, January 1996.

[81] Niedermayer, H., and Rose, O. A simulation-based analysis of the cycle time of cluster tools in semiconductor manufacturing. In *15th European Simulation Symposium* (Delft, The Netherlands, October 2003).

[82] Oechsner, S., and Rose, O. A filtered beam search approach to scheduling cluster tools in semiconductor manufacturing. In *Proceedings of IERC'05* (Atlanta, GA, USA, May 2005).

[83] Oechsner, S., and Rose, O. Scheduling cluster tools using filtered beam search and recipe comparison. In *Winter Simulation Conference* (Orlando, FL, USA, December 2005), pp. 2203–2209.

[84] Ono, M., Asakawa, Y., and Sato, T. Inspection recipe management based on captured defect distribution. In *IEEE International Symposium on Semiconductor Manufacturing* (San Jose, CA, USA, September/October 2003), pp. 145–148.

[85] Pierce, N., and Yurtsever, T. Dynamic dispatch and graphical monitoring system. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (Boston, MA, USA, September 1999), pp. 464–468.

[86] Potti, K., and Whitaker, M. Cycle time reduction at a major texas instruments wafer fab. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (Munich, Germany, March–April 2003), pp. 106–110.

[87] Qi, C., Tang, T., and Sivakumar, A. Simulation based cause and effect analysis of cycle time and wip in semiconductor wafer fabrication. In *Winter Simulation Conference* (2002), pp. 1423–1430.

[88] Robinson, J., Fowler, J., and Neacy, E. Capacity loss factors in semiconductor manufacturing. Tech. rep., Fabtime, March 2003.

[89] Rosen, J. The gradient projection method for nonlinear programming part I : linear constraints. *SIAM Journal* (1960), 180–217.

[90] Rossi, A. *Ordonnancement en milieu incertain, mise en oeuvre d'une démarche robuste.* PhD thesis, INPG, Grenoble, France, October 2003.

[91] Rossi, A., Espinouse, M.-L., and Jacomino, M. *Robustesse de la configuration d'un parc de machines partiellement multifunctions.* Hermés, Paris, France, 2005, ch. 3, pp. 51–69.

[92] Rossi, A., Jacomino, M., and Espinouse, M.-L. Etude de robustesse : Configuration d'un parc de machines partiellement multifonctions. *Journal Européen des Systèmes Automatisés* (2004), 373–395.

[93] Råde, L., and Westergren, B. *Mathematics Handbook for Science and Engineering*, 4 ed. Studentlitteratur, 2001.

[94] Rulkens, H., van Campen, E., van Herk, J., and Rooda, J. Batch size optimization of a furnace and pre-clean area by using dynamic simulations. In *Advanced Semiconductor Manufacturing Conference* (Boston, MA, USA, 1998), pp. 439–444.

[95] SEMI. Standard for definition and measurement of equipment productivity. Tech. Rep. E79-0200, 2000.

[96] Sethi, A., and Sethi, S. Flexibility in manufacturing : A survey. *The International Journal of Flexible Manufacturing Systems* (1990), 289–328.

[97] Shor, N. *Minimization Methods for Non-differentiable Functions.* Springer-Verlag.

[98] Slater, R. *Portraits in silicon.* MIT Press, Cambridge, MA, USA, 1987.

[99] Sloan, T. Shop-floor scheduling of semiconductor wafer fabs : Exploring the influence of technology, market and performance objectives. *The International Journal of Flexible Manufacturing Systems* (1990), 289–328.

[100] Sorensen, C. *My fourty years with Ford.* W.W. Norton, New York, NY, USA, 1956.

[101] Srinivasan, R. Modeling and performance analysis of cluster tools using Petri nets. *IEEE Transactions on Semiconductor Manufacturing 11* (1998), 394–403.

[102] Stalling, W. *Operating Systems Internals and Design Principles*, 5 ed. Prentice Hall, Upper Saddle River, NJ, USA, 2004.

[103] Van Campen, E. *Design of multi-process multi-product wafer fab.* PhD thesis, TU Eindhoven, Eindhoven, The Netherlands, April 2001.

[104] Veeger, C., Etman, L., van Herk, J., and Rooda, J. Generqting cycle time-throughput curves using effective process time based aggregate modeling. In *Advanced Semiconductor Manufacturing Conference* (Cambridge, MA, USA, May 2008), pp. 127–133.

[105] Venkatesh, S., Davenport, R., Foxhoven, P., and Nulman, J. A steady-state throughput analysis of cluster tools : dual-blade versus single-blade robots. *IEEE Transactions on Semiconductor Manufacturing 10* (1997), 418–424.

[106] Vialletelle, P., and France, G. An overview of an original WIP management framework at a high level volume/high mix facility. In *IFAC Symposium on Information Control Problems in Manufacturing* (Saint-Etienne, France, May 2006), vol. 4, pp. 89–92.

[107] Wahab, M., Wu, D., and Lee, C.-G. A generic approach to measuring the machine flexibility of a manufacturing system. *European Journal of Operational Research 1* (2008), 137–149.

[108] Walsh, M. Specialized foundries find niche. *Semiconductor International* (August 2008).

[109] Williams, T. Recipe management : a matter of efficiency. *Semiconductor Fabtech* (1999), 47–51.

[110] Wolfe, P. Convergence condtions for ascent methods. *SIAM Review* (1969), 226–235.

[111] Yi, J., Ding, S., Song, D., and Zhang, M. Scheduling analysis of cluster tools with buffer/process modules. In *IEEE International Conference on Robotics and Automation* (Rome, Italy, April 2007), pp. 985–990.

[112] Yugma, C., Dauzère-Pérès, S., Derreumaux, A., and Sibille, O. A batch optimization software for diffustion area scheduling in semiconductor manufacturing. In *Advanced Semiconductor Manufacturing Conference* (Boston, MA, USA, May 2008), pp. 327–332.

[113] Yugma, C., Riffart, R., Vialletelle, P., Dauzère-Pérès, S., and Buttin, F. A dispatcher simulator for a photolithography workshop. In *Advanced Semiconductor Manufacturing Conference* (Stresa, Italy, June 2007).

[114] Yurtsever, T., and Comerford, M. Equipment management system (ems). In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference* (Cambridge, MA, USA, 1995), pp. 248–254.

[115] Zahara, E., and Fan, S.-K. Real-coded genetic algorithm for stochastic optimization : A tool for recipe qualification of semiconductor manufacturing under noisy environments. *International Journal for Advanced Manufacturing Technology* (2003), 1–9.

Modeling and optimizing flexible capacity allocation in semiconductor manufacturing

Abstract : In this thesis, capacity allocation is modeled for a semiconductor fabrication facility (wafer fab), with measures and methods that optimize the capacity allocation of wafer fabs. The proposed approach supports effective qualification management in wafer fabs (i.e. qualifications of products on tools), such that the engineers can increase the flexibility for operators. Operators need flexibility to decide how the workload should be allocated in order to optimally use the capacity of the tools. To do that, four flexibility measures are proposed : toolset flexibility (favors qualification of tools for process with high workload), WIP flexibility (favors possibility of balancing the workload on the tools), time flexibility (favors balancing of the production times on the tools) and system flexibility (combines all the previous measures). In order to use two of the flexibility measures (WIP flexibility and time flexibility), the optimal balance of workload and production times on the tools need to be found. To do this, optimization programs need to be solved beforehand. Furthermore, the integration of dynamically changing workload, the optimization of multiple qualifications of products on tools and numerous numerical experiments are presented. From this, conclusions are drawn and perspectives for furthers studies are presented.

Keywords : capacity allocation, wafer manufacturing, flexibility, optimization

Allocation flexible des capacités pour la fabrication de semi-conducteurs : Modélisation et optimisation

Résumé : Ces travaux de recherche ont été menés au sein d'une usine de fabrication de semi-conducteurs (appelée fab). L'allocation des capacités a été modélisée à l'aide de mesures et de méthodes permettant d'optimiser la flexibilité de répartition des capacités dans les ateliers. Ces travaux permettent de gérer efficacement les qualifications des produits sur les équipements dans la fab en donnant la possibilité aux ingénieurs de rendre plus flexible le travail des opérateurs. Les opérateurs ont besoin de flexibilité pour décider de la façon dont la charge de travail devra être allouée pour utiliser la capacité des équipements de manière optimale. Pour ce faire, quatre mesures de flexibilité sont proposées : la flexibilité des équipements (favorise des qualifications pour des recettes présentant peu de capacités), la flexibilité d'en-cours (favorise la possibilité d'équilibrer la charge de travail sur les équipements), la flexibilité du temps (favorise l'équilibrage et la minimisation du temps de la production sur les équipements) et la flexibilité du système (incluant toutes les mesures précédentes). Afin d'utiliser deux des mesures de flexibilité (la flexibilité d'en-cours et la flexibilité du temps), l'équilibrage optimal de la charge de travail et du temps de production sur les équipements doit être déterminé. Pour ce faire, des méthodes optimales sont mises en œuvre. De plus, l'intégration de l'évolution dynamique des en-cours, l'étude de l'optimisation de plusieurs qualifications sur plusieurs outils ainsi que de nombreux tests numériques sont présentés. Pour finir, des conclusions sont tirées et des perspectives de cette étude sont présentées.

Mots Clés : allocation de capacité, fabrication de semi-conducteurs, flexibilité, optimisation