

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

THESE

Pour obtenir le grade de
DOCTEUR DE L'INPG

Spécialité : « AUTOMATIQUE, PRODUCTIQUE »

Préparée au Laboratoire d'Automatique de Grenoble
Dans le cadre de l'Ecole Doctorale « ELECTRONIQUE, ELECTROTECHNIQUE,
AUTOMATIQUE, TELECOMMUNICATION, SIGNAL »

présentée et soutenue publiquement

par

André ROSSI

Le 14 octobre 2003

Titre :

ORDONNANCEMENT EN MILIEU INCERTAIN, MISE EN ŒUVRE D'UNE DEMARCHE ROBUSTE

Directeur de thèse :

Mireille JACOMINO

JURY

P.	LADET	<i>Président</i>
J.	ERSCHLER	<i>Rapporteur</i>
M-C.	PORTMANN	<i>Rapporteur</i>
E.	SANLAVILLE	<i>Examineur</i>
P.	VIALLETELLE	<i>Examineur</i>
M.	JACOMINO	<i>Directeur de thèse</i>

TABLE DES MATIERES

PARTIE I

Avant-propos.....	6
Chapitre I : Gestion des flux physiques et des flux d'informations en contexte incertain	8
1.1 Introduction.....	8
1.2 Optimisation en contexte incertain	9
1.3 Exemple : problème d'affectation de machines dans des groupes	10
1.4 Introduction à l'ordonnancement.....	12
1.4.1 La fonction ordonnancement	12
1.4.2 Les éléments de base de l'ordonnancement.....	13
1.4.3 Instance, solution et ordonnancement en contexte incertain	16
1.5 Schémas de synthèse prédictif et réactif de l'ordonnancement sans prise en compte explicite de l'incertain.....	17
1.5.1 Ordonnancement prédictif	17
1.5.2 Ordonnancement réactif.....	18
1.6 Anomalie de Graham	20
1.7 Conclusion	22
Chapitre II : Prise en compte explicite du contexte incertain	23
2.1 Introduction.....	23
2.2 Modèles de prise en compte de l'incertain	23
2.2.1 Les modèles de prise en compte de l'incertain	23
2.2.2 Approche par scénarios et analyse de risques.....	25
2.2.3 Définitions de la performance dans le contexte de l'approche par scénarios	25
2.3 Ordonnancement prédictif : robustesse.....	28
2.3.1 Robustesse d'une solution.....	28
2.3.2 Représentation ensembliste de la problématique de la robustesse.....	29
2.3.3 Exemple de mise en œuvre	31
2.3.4 Représentation systémique des problèmes d'ordonnancement avec modélisation de l'incertain	34
2.3.4.1 Ordonnancement prédictif	34
2.3.4.2 Ordonnancement réactif.....	36
2.4 Ordonnancement réactif : flexibilité	37

2.4.1 Introduction et définitions.....	37
2.4.2 Comparaison de la flexibilité avec ORABAID et l'approche par scénarios.	38
2.4.3 Définition du coût de flexibilité.....	40
2.4.4 Application.....	41
2.4.5 Flexibilité et garantie de performance	42
2.5 Choix d'une solution.....	44
2.6 Analogie avec la robustesse des systèmes continus.....	44
2.7 Conclusion	46
PARTIE II	
Introduction.....	48
Chapitre III : Configuration d'un parc de machines parallèles multifonctions en contexte incertain.....	50
3.1 Contexte général de l'étude	50
3.1.1 Les origines du contexte incertain	50
3.1.2 Les objectifs de l'étude	51
3.2 Présentation et modélisation des contraintes de l'atelier de photolithographie.....	51
3.2.1 Modélisation du problème	51
3.2.2 Calcul d'un plan de production optimal.....	52
3.2.3 Examen de deux cas particuliers de distribution des vitesses.....	57
3.3 Caractéristiques et modèles du contexte incertain	59
3.4 Le volume de la commande réelle est incertain.....	60
3.5 Conclusion	61
Chapitre IV : Le volume ou le temps de traitement de la commande est conservé.....	63
4.1 Les machines sont uniformes.....	63
4.1.1 Etude de l'espace des commandes.....	64
4.1.2 Formule de projection sur H_s	65
4.1.3 Ensemble des commandes 1-équilibrées	65
4.1.4 Caractérisation de l'ensemble des commandes 1-équilibrées.....	66
4.1.5 Exemple	69
4.1.6 Représentation graphique de H_s et de $B_1(Q_v)$	71
4.1.7 Distances dans H_s	72
4.1.8 Rayon de stabilité.....	80
4.1.9 Ensemble des commandes centrales d'une matrice de vitesse	83

4.1.10 Exemple	86
4.1.11 Synthèse d'une configuration stable du parc de machines	89
4.1.12 Outils généraux pour l'augmentation du rayon de stabilité	90
4.1.13 Augmentation du rayon de stabilité à l'aide d'un seul 1	98
4.2 Les machines sont identiques.....	102
4.2.1 Etude de l'espace des commandes	104
4.2.2 Formule de projection sur H_m et ensemble des commandes 1-équilibrées	105
4.2.4 Caractérisation de l'ensemble des commandes 1-équilibrées.....	105
4.2.5 Comparaison de H_s et de H_m	106
4.2.6 Distance dans H_m	106
4.2.7 Le rayon de stabilité et son augmentation.....	107
4.3 Conclusion	108
Chapitre V : Le volume et la répartition des produits de la commande sont incertains	109
5.1 Etude de l'espace des commandes.....	109
5.2 Caractérisation des commandes simplement équilibrées.....	109
5.2.2 Démonstration du théorème 4.....	111
5.2.3 Démonstration de cas particuliers du théorème 4	114
5.3 Exemples.....	117
5.3.1 Les machines sont uniformes.....	117
5.3.2 Les machines sont identiques.....	119
5.4 Distances dans \mathfrak{R}_+^n	120
5.4.1 Distance séparant deux commandes de \mathfrak{R}_+^n	120
5.4.2 Distance d'une commande à une frontière dans \mathfrak{R}_+^n	121
5.5 Rayon de stabilité.....	126
5.5.1 Les machines sont uniformes.....	126
5.5.2 Les machines sont identiques.....	129
5.6 Conclusion	130
6 Conclusions et perspectives	131
6.1 Problème de la configuration <i>ex-nihilo</i>	131
6.2 Problème de la suppression des « 1 »	132
6.2.1 Exemple	132
6.3 Problème de l'impact des pannes survenant dans l'atelier	133

6.4 Problèmes de l'ordonnancement des produits sur les machines	133
6.4.1 Problème de la trajectoire de commande	134
6.4.2 Ensemble des solutions optimales de (PL 1)	135
Références.....	139

Avant-propos

Ces travaux de thèse ont été réalisés au Laboratoire d'Automatique de Grenoble, dans le cadre d'une allocation du Ministère de l'Education Nationale de la Recherche et de la Technologie. Je remercie messieurs Luc Dugard et Alain Barraud, directeurs successifs du LAG, de m'avoir accueillis et permis d'effectuer mes travaux de recherche dans d'excellentes conditions.

Je tiens à remercier madame Mireille Jacomino, professeur à l'INPG, pour la qualité de son encadrement, ses conseils et la disponibilité qu'elle m'a accordé malgré les contraintes que lui impose sa charge de directrice des études de l'ENSIEG.

J'exprime toute ma gratitude à mademoiselle Marie-Claude Portmann, professeur à l'Ecole des Mines de Nancy, et à monsieur Jacques Erschler, professeur à l'INSA de Toulouse pour l'intérêt qu'ils ont porté à mes travaux de thèse et pour avoir accepté la charge d'en être les rapporteurs.

Je remercie monsieur Eric Sanlaville, maître de conférence à l'Université Blaise Pascal et animateur du groupe Flexibilité du GOTHa (Groupe de travail en Ordonnancement Théorique et Appliqué) pour ses conseils, sa relecture attentive de mes travaux et sa participation à mon jury de thèse.

Je remercie également monsieur Philippe Vialletelle, ingénieur de la société ST Microelectronics, pour m'avoir permis de travailler avec son équipe sur le problème de la configuration des machines de photolithographie, et pour sa participation à mon jury de thèse.

Un grand merci aux membres de l'équipe CROSS du LAG (Conduite Robuste, Surveillance et Supervision) pour leur soutien, leurs conseils et leur collaboration à mes travaux. Je pense en particulier à Marie-Laure Espinouse, Eric Zamaï, Bernard Descotes-Genon, Héctor Mendez Azua, Thérèse Mencerey, Philippe Granier et Sébastien Henry.

Ces travaux doivent aussi beaucoup aux réflexions et au contenu scientifique apportés par les membres du groupe Flexibilité du Gotha, qui m'ont chaleureusement accueilli dès les premiers mois de mon DEA. Je remercie en particulier Mohamed Ali Aloulou, Jean-Charles Billaut, Amine Mahjoub, Aziz Moukrim, et Claude Yugma.

Pendant ces trois années passées au LAG, j'ai été très sensible à la gentillesse (mais aussi à l'efficacité) de l'équipe technique et de l'équipe administrative du labo. Qu'ils trouvent ici l'expression de mon amitié.

Parallèlement à mes travaux de recherche, le CIES de Grenoble m'a accordé un poste de moniteur à l'ESISAR, sans lequel mes conditions de vie n'auraient pas été aussi favorables. Je tiens à remercier messieurs Claude Gaubert et Gérard Cognet, directeurs successifs du CIES de Grenoble, qui m'ont permis de vivre cette initiation à l'enseignement comme une expérience passionnante et irremplaçable. Mes remerciements s'adressent aussi aux enseignants de l'ESISAR, et en particulier à Nathalie Fulget, directrice des études, et à Damien Koenig, mon tuteur au CIES.

PARTIE I

ORDONNANCEMENT EN CONTEXTE

INCERTAIN

Chapitre I : Gestion des flux physiques et des flux d'informations en contexte incertain

1.1 Introduction

Les systèmes de production au sens large (production de biens mais aussi de services) peuvent être vus comme des systèmes de gestion de flux. Une entreprise de production manufacturière, par exemple, est traversée par des flux physiques (des matières premières aux produits finis), des flux financiers (paiements, recettes), des flux énergétiques (transformation, stockage et répartition des différentes sources d'énergie utilisées) et des flux d'informations (commandes clients, commandes passées aux fournisseurs et sous-traitants, pilotage des outils de production). Les calculateurs parallèles, ainsi que les réseaux de télécommunications et d'informations sont soumis à la même problématique de gestion des flux.

La gestion de ces flux vise à remplir un ou plusieurs objectifs, qui eux, dépendent étroitement du contexte du système étudié. La gestion effective de ces flux nécessite de prendre de très nombreuses décisions, à des niveaux divers. Dans une entreprise, une augmentation de capital est une décision qui relève du niveau stratégique alors que l'ordre de passage d'un ensemble de pièces sur une machine est une décision qui relève du niveau opérationnel. Dans les deux cas, on dit qu'on a affaire à un problème d'optimisation. En effet, le processus de prise de décision vise à optimiser un certain nombre d'objectifs parfois contradictoires. Lorsque plusieurs séries de pièces doivent être fabriquées dans un atelier, on peut maximiser la productivité en fabriquant les pièces identiques par lots importants, au détriment du temps de cycle moyen des pièces. A l'inverse, on peut minimiser le temps de cycle moyen des pièces au détriment de la productivité en traitant d'abord les pièces qui attendent dans l'atelier depuis le plus long temps. L'optimisation consiste alors à trouver un compromis rationnel entre ces objectifs.

Les domaines où l'optimisation est un outil privilégié sont la planification, l'ordonnancement et l'affectation. La planification consiste à programmer dans le temps les approvisionnements de matières premières et la fabrication afin de répondre à un cahier des charges. D'après la définition de [Baker, 1974], « l'ordonnancement est la distribution des ressources dans le temps pour effectuer un ensemble d'activités ». L'ordonnancement sera présenté de manière plus détaillée dans la section 1.4. L'affectation consiste à assigner des tâches à des ressources, sans préciser l'ordre de passage des tâches affectées aux ressources. L'affectation est un sous problème de l'ordonnancement. La résolution d'un problème d'optimisation nécessite tout d'abord une phase de modélisation du système réel. Une fois qu'un modèle est construit, ses paramètres sont instanciés par un jeu de valeurs numériques afin de représenter au mieux les conditions dans lesquelles le modèle est utilisé. Ensuite, il

convient d'expliciter les critères d'optimisation, qui reflètent les préférences de l'utilisateur quant à l'exploitation du système. Les méthodes d'optimisation ont alors pour objet de déterminer une solution optimale compte tenu des contraintes imposées par le modèle, des données numériques et des critères de performance de l'utilisateur concernant la solution. Une solution est optimale si l'on peut garantir qu'elle satisfait au mieux les critères de l'utilisateur.

En pratique, l'obtention d'une solution optimale à un problème d'optimisation se heurte à une difficulté majeure : l'explosion combinatoire. La complexité des modèles et leur grande taille interdit généralement d'énumérer toutes les solutions dans le but de choisir ensuite la meilleure. On peut alors se contenter de calculer une heuristique, c'est-à-dire une solution non optimale, mais dans ce cas, il est important de connaître la distance à l'optimum (ou au moins une borne), afin de pouvoir juger de la qualité de l'heuristique. L'obtention d'une borne lorsqu'on utilise une heuristique n'est en général pas aisée. Enfin, même s'il existe de nombreuses méthodes exactes pour résoudre les problèmes d'optimisation, celles-ci peuvent ne pas être applicables dans certaines conditions, toujours à cause de l'explosion combinatoire. Finalement, seuls quelques problèmes précis peuvent être résolus facilement et de manière exacte. La partition des problèmes en deux classes : Polynomial (facile) et Non Polynomial (difficile) est l'objet de la théorie de la complexité. On se référera à [Garey et Johnson, 1979] pour plus de détails. La plupart des problèmes de planification, d'ordonnancement et d'affectation sont équivalents à des problèmes d'optimisation. Le problème de leur résolution est donc commun à plusieurs disciplines. Dans le cadre de cette thèse, on s'intéressera à l'ordonnancement, et en particulier aux problèmes d'affectation. Mais les résultats présentés pourront être étendus à l'optimisation d'autres problèmes sans perte de généralité.

1.2 Optimisation en contexte incertain

Les problèmes d'optimisation auxquels on s'intéresse particulièrement ont pour origine des problèmes industriels. Or ces problèmes réels sont traités à travers des modèles dont le comportement ne représente la réalité de manière satisfaisante que sous certaines conditions. La phase de modélisation est une première rupture avec le problème réel que l'on souhaite traiter, d'autant plus que pour limiter la complexité du modèle, on aura tendance à ne pas lui inclure certaines contraintes difficiles à intégrer ou à exprimer.

D'autre part, le modèle est assorti de valeurs numériques censées représenter l'état du système réel. Il s'agit par exemple de la durée d'exécution d'une tâche, ou de la vitesse d'une machine. Or ces données numériques peuvent être mal connues ou être susceptibles de changer, ce qui est une autre source d'incompatibilité possible du modèle avec la réalité. On distingue généralement deux classes de phénomènes s'opposant à une instanciation parfaite des données numériques :

Les incertitudes : il s'agit des imprécisions qui affectent les mesures physiques (durées, dimensions), mais aussi des erreurs qui découlent de la différence entre l'estimation prévisionnelle d'une grandeur et sa valeur effective (variabilité).

Les aléas : il s'agit d'événements de type discret, qui entraînent des modifications dans le modèle lui-même (la panne d'une machine ou la rupture d'un approvisionnement par exemple).

Lorsqu'on applique une méthode d'optimisation, il est nécessaire de fournir un modèle (de préférence pas trop complexe), un critère à optimiser, ainsi qu'un jeu de valeurs numériques. La solution retournée, exacte ou approchée, dépend directement de ces données. Or les incertitudes sur les valeurs numériques ainsi que les aléas qui affectent l'atelier réel sont de nature à introduire des différences plus ou moins importantes entre le modèle qui a servi à calculer une solution exacte ou approchée, et le système réel. Il en résulte que l'optimalité de la solution (dans le cas du calcul d'une solution exacte) ou la valeur garantie de sa distance à l'optimum (dans le cas du calcul d'une solution approchée) peuvent être remises en question par la différence qui existe entre le modèle et les données utilisées pour l'optimisation d'une part, et le système réel auquel on souhaite appliquer cette solution d'autre part. Il n'est pas impossible non plus que les perturbations (aléas et incertitudes) soient telles que la solution calculée ne soit même plus applicable au système réel.

Ainsi, tout le bénéfice apporté par les méthodes d'optimisation peut être perdu par la mauvaise estimation de la valeur numérique d'un paramètre, ou l'occurrence d'un événement imprévu. Cette perte est d'autant plus dommageable que l'obtention d'une solution par une méthode d'optimisation se fait souvent au prix d'un temps de calcul et de besoins en mémoire relativement importants. La sensibilité des modèles aux perturbations ne constitue pas un inconvénient anecdotique : les problèmes d'optimisation issus de la planification, de l'ordonnancement ou de l'affectation se posent dans le contexte des entreprises. Or le contexte de l'activité industrielle est soumis à de fortes perturbations tant au niveau tactique (fluctuation du marché, variation des prix), qu'au niveau opérationnel (pannes de machines, retards d'approvisionnement). Hors du contexte de la production industrielle, les perturbations de nature diverses affectent profondément la gestion des réseaux de communication et de distribution d'énergie, ou l'organisation des tâches de calcul distribué utilisant Internet.

Il serait pourtant dommage de renoncer aux bénéfices de l'optimisation sous prétexte que les perturbations peuvent remettre en question l'intérêt des solutions calculées. L'objectif de cette thèse est de proposer des mesures permettant de savoir quels types de perturbations remettent en cause les performances d'une solution donnée, afin de savoir si telle solution convient ou non au contexte où l'on souhaite l'appliquer. Plus généralement, les stratégies permettant d'appliquer des solutions en contexte incertain, avec ou sans garantie de performance, seront présentées dans la partie II. Dans le paragraphe suivant, un exemple de problème d'affectation en contexte incertain illustre la nécessité de considérer explicitement l'incertain dans le calcul d'une solution.

1.3 Exemple : problème d'affectation de machines dans des groupes

On considère un atelier composé de m machines parallèles destinées à traiter une commande supposée incertaine. Une commande est un vecteur de $n=3$ composantes, où la $j^{\text{ème}}$ composante indique la quantité de produit j à fabriquer. Le but du problème est d'affecter chacune des m machines à l'un des n produits à fabriquer de manière à minimiser le temps de traitement total de la commande.

La durée de traitement du produit j dépend de la quantité à produire et du nombre de machines qui lui a été affectée. Les machines ont toutes une vitesse unitaire (une unité de

produit est fabriquée par unité de temps), et la $j^{ème}$ composante du vecteur commande indique la quantité de produit à fabriquer.

Par exemple, si la commande i_1 s'écrit :

$$i_1 = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$$

Alors 150 unités de produit 1, 150 unités de produit 2 et 150 unités de produit 3 sont à fabriquer. Pour cela, on doit affecter chacune des m machines de l'atelier à un produit et un seul, de manière à minimiser le temps maximum de traitement de tous les produits. Si l'atelier compte $m=9$ machines, la solution optimale consiste simplement à affecter 3 machines à chaque produit, ce qui conduit à un temps de traitement total T_1 de la commande i_1 égal à :

$$T_1 = \max\left(\frac{150}{3}, \frac{150}{3}, \frac{150}{3}\right) = 50$$

Comme les machines travaillent en parallèle, le temps de traitement total T_1 est égal au plus grand temps de traitement de chaque produit, et non à la somme de ces durées. La commande i_1 est traitée en 50 unités de temps. La solution optimale s associée à la commande i_1 s'écrit $s = [3 \ 3 \ 3]$, où $s(j)$ représente le nombre de machines affecté au produit j .

La solution s a été calculée pour minimiser le temps de traitement de la commande i_1 . L'algorithme qui a été utilisé est décrit dans [Rossi, 2000]. Cependant, supposons que la commande est incertaine. Cela peut provenir du fait que la commande i_1 est une commande prévisionnelle, et que le client a modifié son contenu. Considérons que la solution s calculée pour i_1 soit finalement appliquée à la commande i_2 définie par :

$$i_2 = \begin{bmatrix} 50 \\ 150 \\ 250 \end{bmatrix}$$

La solution s est toujours applicable, mais le temps de traitement total de i_2 s'écrit :

$$T_2 = \max\left(\frac{50}{3}, \frac{150}{3}, \frac{250}{3}\right) = 83.33$$

Ainsi, alors que le volume de production de i_1 et de i_2 est identique (450 unités de produits au total), le traitement de la commande i_1 nécessite 50 unités de temps lorsque les machines sont affectées aux produits suivant la solution s , et le temps nécessaire au traitement de i_2 avec la même solution est de 83.33 unités de temps, soit une augmentation de plus de 66 %.

Cet exemple illustre la nécessité de prendre en compte le caractère incertain de la commande pour calculer une solution. Ici, le calcul de s avait pour objectif de ne satisfaire qu'un seul critère : la minimisation du temps de traitement de i_1 . Si l'on souhaite assurer une certaine régularité des performances de la solution malgré les variations des données, il paraît souhaitable de tenir compte de la nature et de l'amplitude de ces variations dans les méthodes d'optimisation utilisées pour calculer la solution s . C'est ce nouvel objectif qui est appelé robustesse.

1.4 Introduction à l'ordonnancement

1.4.1 La fonction ordonnancement

Baker définit l'ordonnancement comme la distribution des ressources dans le temps pour réaliser un ensemble d'activités [Baker, 1974]. On peut trouver d'autres définitions dans [Carlier et Chrétienne, 1988] ou dans [Esquirol et Lopez, 1999]. Les ressources dont il est question sont celles d'un système de production au sens large : un atelier flexible de production manufacturière, ou un ensemble de calculateurs reliés en réseau sont des systèmes de production. Le premier manipule un flux physique, le second traite un flux de données. Dans tous les cas, un système de production a pour rôle de conférer de la valeur ajoutée au flux qui le traverse : un convoyeur ne peut pas être considéré comme un système de production si le transport des pièces convoyées n'est pas lui-même considéré comme une valeur ajoutée.

L'ordonnancement a pour objet le pilotage des systèmes de production, c'est-à-dire le processus de prise de décisions permettant d'atteindre un ou plusieurs objectifs fixés. Comme le laisse entrevoir la définition de Baker, un problème d'ordonnancement comporte deux sous problèmes : le premier consiste à affecter les tâches aux ressources, et le second a pour objet de fixer le calendrier de l'exécution des tâches, pour chaque ressource. La prise en compte d'objectifs à atteindre, exprimés sous forme de critères à optimiser, ramène souvent les problèmes d'ordonnancement à des problèmes d'optimisation, et leur résolution ne permet généralement pas de traiter les deux sous problèmes de manière indépendante.

Classiquement [Bénassy, 1987], un système de production doit être piloté dans le but d'atteindre l'un des quatre objectifs ci-dessous :

- satisfaire la demande de produits finis en termes de volume
- garantir la qualité des produits
- maîtriser les coûts liés à l'activité de production
- achever la production dans les délais impartis

D'autres objectifs, plus ou moins implicites peuvent être ajoutés à cette liste, comme garantir la sécurité des personnes, ne pas dégrader l'environnement, assurer la pérennité des

moyens de production, conserver une certaine stabilité en présence d'événements perturbateurs, etc. L'objet de l'ordonnancement est d'élaborer des décisions permettant de satisfaire au mieux les objectifs fixés.

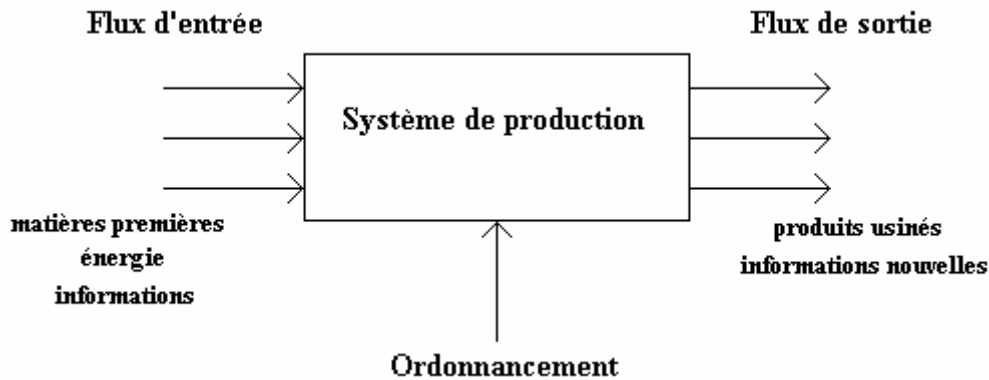


Figure 1 : représentation des flux traversant un système de production

La Figure 1 illustre la place centrale de l'ordonnancement dans la gestion des flux qui traversent le système de production.

1.4.2 Les éléments de base de l'ordonnancement

Les systèmes de production étant souvent hiérarchisés, le processus de prise de décision l'est lui aussi, et il en est de même pour l'ordonnancement. Bien que les objectifs et les enjeux des décisions ne soient pas les mêmes au niveau stratégique et au niveau opérationnel, la prise de ces décisions relève de l'ordonnancement. Un problème d'ordonnancement regroupe la description des ressources disponibles, l'ensemble des travaux à réaliser à l'aide des ressources, les contraintes inhérentes à l'utilisation des ressources et aux conditions de réalisation des travaux, et enfin, une liste de critères reflétant les préférences de l'utilisateur relativement à la solution attendue.

Un ordonnancement désigne une solution à un problème d'ordonnancement. D'après Esquirol et Lopez, un ordonnancement « décrit l'exécution des tâches et l'allocation des ressources au cours du temps et vise à satisfaire un ou plusieurs objectifs. De manière plus précise, on parle d'ordonnancement lorsqu'on fixe les dates de début ou de fin de chacune des tâches » [Esquirol et Lopez, 1999].

Le but de ce paragraphe est d'éclairer et d'illustrer les éléments de cette définition d'un problème d'ordonnancement.

1.4.2.1 Travaux et tâches

Une tâche (ou opération) est une activité élémentaire à réaliser par le système de production considéré. Un travail est un ensemble de tâches dont l'exécution est nécessaire pour atteindre un objectif. Pour une entreprise de construction, la fabrication d'une toiture est

une tâche, alors que pour un couvreur la fabrication d'une toiture est un travail. On note p_i la durée de la tâche i , r_i la date à partir de laquelle elle est disponible, et c_i sa date de fin. c_i est fixée par l'ordonnancement, contrairement à p_i et r_i qui sont des données. On notera $C_{\max} = \max_i(c_i)$ la date de fin de la dernière tâche exécutée dans l'ordonnancement.

1.4.2.2 Ressources

Le terme ressource désigne indifféremment des moyens humains, matériels, financiers ou énergétiques pouvant être utilisés pour réaliser des tâches. Une ressource est renouvelable si elle est toujours disponible après avoir été utilisée pour exécuter une tâche : un autobus est une ressource renouvelable. Elle est non renouvelable sinon : une somme d'argent est une ressource non renouvelable car elle n'est plus disponible après avoir été utilisée. On distingue également les ressources disjonctives et cumulatives. Une ressource disjonctive ne peut exécuter qu'une tâche à la fois. Une ressource cumulative peut exécuter plusieurs tâches simultanément.

1.4.2.3 Contraintes

Une contrainte traduit une limitation ou une impossibilité dans l'exploitation du système de production. Les contraintes que l'on rencontre le plus fréquemment sont liées à l'utilisation des ressources, à l'exécution des tâches ou aux gammes de fabrication des produits. Les contraintes nécessaires à la modélisation d'un système de production peuvent être nombreuses et complexes (voir [Erschler, 1976]). Dans [Billaut, 1993], les contraintes sont rangées dans trois catégories :

1. Les contraintes potentielles : ces contraintes précisent les fenêtres de temps pendant lesquelles il est possible d'exécuter une tâche. Les contraintes de précédence entre les tâches, qui traduisent l'impossibilité d'exécuter une tâche avant une autre, sont également des contraintes potentielles.
2. Les contraintes disjonctives : ces contraintes traduisent l'impossibilité d'exécuter deux tâches simultanément.
3. Les contraintes cumulatives : ces contraintes expriment la limitation des ressources. Le fait qu'une machine ne puisse pas traiter deux tâches en même temps est une contrainte cumulative.

1.4.2.4 Fonction objectif

Etablir les contraintes qui régissent l'utilisation du système étudié, c'est répondre de manière explicite et exhaustive à la question « qu'a-t-on le droit de faire ? ». Les ordonnancements qui respectent toutes les contraintes sont acceptables et réalisables du point de vue technique, mais ils ne sont pas tous nécessairement pertinents du point de vue de l'exploitation du système de production, et des objectifs fixés par l'entreprise.

La fonction objectif exprime les objectifs poursuivis par l'utilisateur, et permet d'établir un classement rationnel des ordonnancements. Mais la pertinence d'un ordonnancement ne lui est pas intrinsèque. En effet, un même système de production peut être utilisé dans des contextes différents : la production d'un même atelier peut par exemple être organisée de

manière à minimiser le temps d'exécution des tâches, ou de manière à minimiser les encours de production. Ceci souligne l'importance du choix de la fonction objectif, qui exprime les critères de performance de l'ordonnancement privilégiés par l'utilisateur.

Les critères les plus couramment utilisés portent sur :

- *Le temps* : on peut souhaiter minimiser le temps d'exécution de l'ensemble des tâches, la somme des dates de fin de chacune des tâches, le nombre de tâches en retard (si les tâches ont des dates de livraison), le plus grand retard, etc.
- *L'utilisation des ressources* : dans certains cas, on souhaite maximiser le taux d'occupation des machines (c'est-à-dire minimiser la durée des périodes d'inactivité des équipements), ou au contraire minimiser les ressources à employer.
- *Les coûts qui découlent de l'activité de production* : les coûts de stockage, d'immobilisation pour maintenance, et de transport sont à minimiser.

On trouvera une présentation plus complète des critères de performance dans [Viswanadham et Narahari, 1992].

Lorsqu'on a affaire à au moins deux critères de performance, on dit que le problème d'ordonnancement est multicritère. L'élaboration d'une solution à un tel problème consiste alors à trouver un compromis entre ces différents critères. Classiquement, on distingue trois stratégies pour atteindre un compromis :

La non dominance : lorsque on a n critères, cette méthode consiste à recenser les ordonnancements qui ne sont pas dominés. On dit qu'un ordonnancement o_1 domine un ordonnancement o_2 au sens des n critères si la performance de o_1 est supérieure à la performance de o_2 sur les n critères simultanément et strictement sur un critère. La non dominance permet d'exclure les solutions dominées, mais n'établit pas de classement des solutions non dominées. Cette méthode est particulièrement intéressante en aide à la décision.

La hiérarchisation des critères : les critères de performance sont classés par ordre d'importance décroissante. Ainsi, le premier critère permet d'éliminer les ordonnancements trop peu performants relativement à ce critère. On reproduit cette démarche avec les critères restants.

L'agrégation des critères : on établit une combinaison linéaire des critères (où chaque critère est pondéré) afin de former un nouveau critère « général ». La performance des ordonnancements est alors évaluée à l'aide de ce seul critère général. L'inconvénient majeur de cette méthode consiste à déterminer une pondération des critères acceptables pour l'utilisateur. D'autre part, les critères étant souvent non homogènes (il peut y avoir des temps, de unités monétaires, des quantités de ressources), former une combinaison linéaire « homogénéisée » par des coefficients de pondération revient à comparer des grandeurs non comparables d'un point de vue physique. Au-delà du choix des coefficients de pondération, le sens du critère général est alors contestable.

Pour plus de précisions sur l'aide à la décision multicritère, on pourra consulter l'ouvrage de [Roy, 1985].

1.4.3 Instance, solution et ordonnancement en contexte incertain

La prise en compte du contexte incertain, qui fait l'objet du chapitre II, est beaucoup moins classique que les éléments présentés dans les paragraphes précédents. On ne présentera ici que les notions utilisées dans les schémas de synthèse des méthodes prédictives et des méthodes réactives lorsque celles-ci sont appliquées en contexte incertain.

1.4.3.1 Scénario

Dans cette thèse, les termes « jeu de données », « scénario » et « instance » sont synonymes. Un scénario d'un problème d'ordonnancement est un ensemble de valeurs numériques. Il s'agit d'une instanciation complète du modèle du problème. Dans un problème d'ordonnancement à deux machines, un scénario comporte la durée opératoire de toutes les tâches à exécuter, avec toutes les contraintes de précédence, ainsi que la disponibilité des machines. En pratique, on ne considère pas de scénario décrivant le problème de manière totalement exhaustive. Un scénario ne contient que les valeurs numériques associées à des paramètres que l'on suppose susceptibles de changer. Dans le problème à deux machines, si l'on fait l'hypothèse que l'incertain n'affecte que la durée des tâches, alors il ne sera pas fait mention de la disponibilité des machines dans les scénarios. Ceci explique aussi pourquoi on n'utilise pas les scénarios en contexte certain : toutes les données numériques sont supposées connues et invariables. Dans la suite, un scénario sera noté i , et I_A désignera l'ensemble des scénarios qui a été retenu pour modéliser l'incertain.

1.4.3.2 Solution

Une solution à un problème d'ordonnancement est un ensemble de décisions portant sur la date d'exécution des tâches, et l'affectation de ces tâches aux ressources, indépendamment de l'ensemble I_A des scénarios considérés, et de toute valeur numérique relative aux éléments de cet ensemble. On notera s une solution. Une solution est par exemple un ordre de passage d'un ensemble de tâches sur une machine. Cet ordre est indépendant des durées effectives des tâches.

1.4.3.3 Ordonnancement

Un ordonnancement est un calendrier figé du déroulement des tâches sur les ressources. Si la définition donnée dans [Esquirol et Lopez, 1999] n'est pas remise en question, l'approche par scénarios permet de définir formellement un ordonnancement comme l'application d'une solution à un scénario particulier. Autrement dit, un ordonnancement est le résultat de décisions génériques sur une instance précise du problème. Un ordonnancement sera noté o , et sa définition formelle permet d'écrire $o = s(i)$. La solution s est vue comme une fonction dont la variable est un scénario i .

1.5 Schémas de synthèse prédictif et réactif de l'ordonnancement sans prise en compte explicite de l'incertain

Classiquement, on distingue deux classes d'approches en contexte incertain : les méthodes déterministes et les méthodes stochastiques. Dans ce mémoire, on ne s'intéresse qu'aux méthodes déterministes. Parallèlement à cette première classification, on peut aussi distinguer les méthodes prédictives des méthodes réactives.

- Les méthodes prédictives font l'hypothèse qu'il n'est pas possible de modifier la solution proposée dès lors que son exécution a commencé. C'est le cas du problème d'affectation décrit dans ce paragraphe. Cette approche consiste à proposer une solution « robuste », c'est-à-dire dont les performances sont peu sensibles aux variations des données. La notion de robustesse sera étudiée plus en détail dans la partie II de ce mémoire, dans le cadre de l'ordonnancement. De plus, l'exemple d'affectation développé dans ce paragraphe sera repris pour illustrer la notion de robustesse.
- Les méthodes réactives, à l'opposé des méthodes prédictives, supposent qu'il est possible de modifier ou de changer la solution courante. Cette possibilité supplémentaire doit cependant être appliquée en tenant compte des contraintes temps réel, qui imposent des temps de réponse très courts. L'avantage principal des méthodes réactives est qu'il est possible de réagir à des perturbations d'amplitude importantes ou à l'occurrence d'événements rares.

On emploie parfois les termes « statique / dynamique » au lieu de « prédictive / réactive ». Cette terminologie ne sera pas utilisée afin d'éviter certaines confusions : le mot statique est également employé dans la littérature pour désigner des situations où les données sont entièrement connues et fixe, alors que dans ce document, un problème d'optimisation « prédictif » se situe clairement en contexte incertain : si la solution d'un tel problème est « statique », le contexte de son application ne l'est pas.

1.5.1 Ordonnancement prédictif

Dans ce paragraphe, on propose une représentation systémique des problèmes d'ordonnancement en l'absence de modélisation explicite du contexte incertain. On dit qu'un problème d'ordonnancement est prédictif si, une fois la solution calculée hors ligne, aucune modification de cette solution n'est autorisée en cours d'exécution.

- Fonctionnement hors ligne

On considère un scénario prévisionnel i_A , pour lequel on calcule une solution s . Cette solution doit optimiser la valeur de la fonction objectif σ lorsque s est appliquée à i_A . On obtient alors un ordonnancement prévisionnel $o_A = s(i_A)$, pour lequel on peut calculer la performance à l'aide de la fonction objectif. La solution s étant élaborée uniquement pour i_A , on peut s'attendre à obtenir une bonne performance prévisionnelle.

- Fonctionnement en ligne

La solution s , qui a été calculée hors ligne pour l'instance i_A est appliquée à l'instance réelle notée i_r , ce qui conduit à un ordonnancement réel noté o_r dont on mesure la performance locale avec la fonction objectif. Il faut noter que i_r peut être tel que

l'ordonnancement o_r peut ne pas être admissible. Dans ce cas, la solution s n'est pas applicable. On fait l'hypothèse que s reste applicable à i_r .

L'impossibilité d'adapter la solution au scénario réel noté i_r se traduit par une représentation systémique de type « boucle ouverte » (voir Figure 2). La partie du schéma située à l'intérieur de la zone en traits pointillés n'est activée que hors ligne, c'est-à-dire avant l'exécution de la solution s . *A contrario*, les autres éléments de la Figure 2 ne sont actifs qu'en ligne, lorsque le scénario réel courant i_r est appliqué.

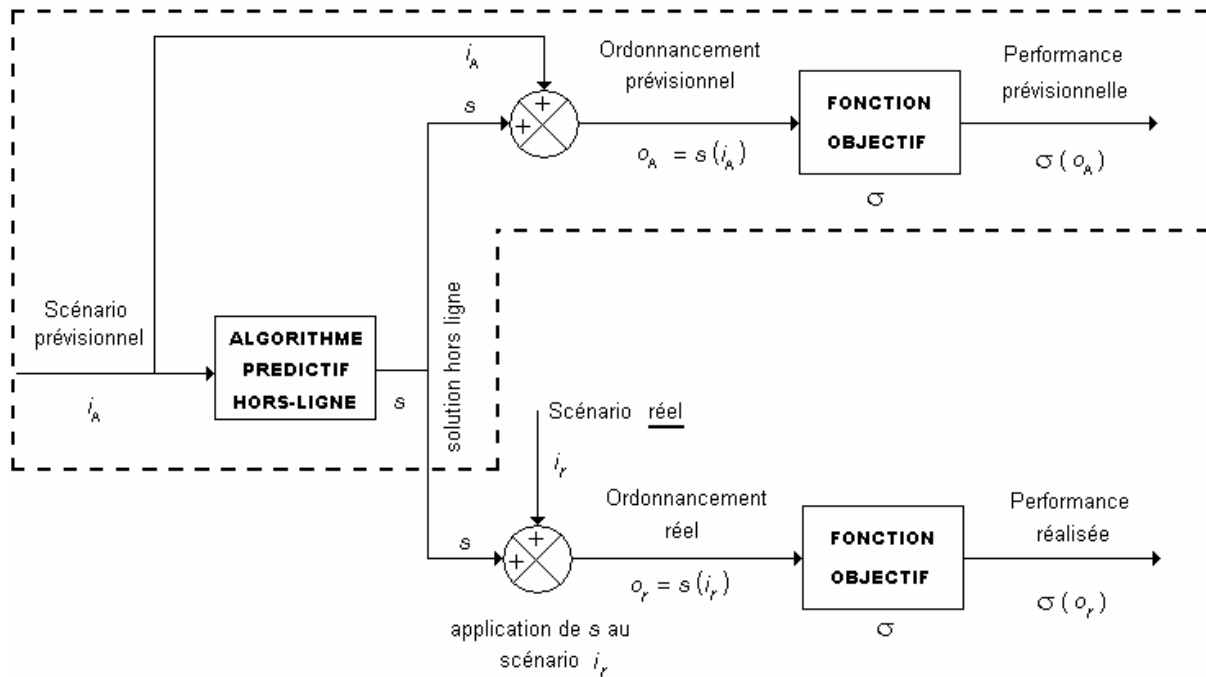


Figure 2 : Représentation systémique d'un problème d'ordonnancement prédictif, sans prise en compte explicite du contexte incertain

La Figure 2 est une représentation typique de ce qui se produit lors qu'on applique une solution calculée pour une instance précise dans un environnement perturbé : sans possibilité de modifier la solution s en ligne, l'utilisateur ne peut que « constater l'ampleur des dégâts » en comparant les performances réalisée et prévisionnelle. C'est le cas de l'exemple présenté au paragraphe 1.3.

1.5.2 Ordonnancement réactif

Un problème d'ordonnancement est dit réactif lorsqu'on autorise la modification de la solution courante en cours d'exécution. On qualifie souvent ces problèmes de « temps-réel » (voir [Manacher, 1967]). La Figure 3 est la représentation systémique d'un tel problème, mais sans modélisation explicite de l'incertain dans l'élaboration de la solution hors ligne.

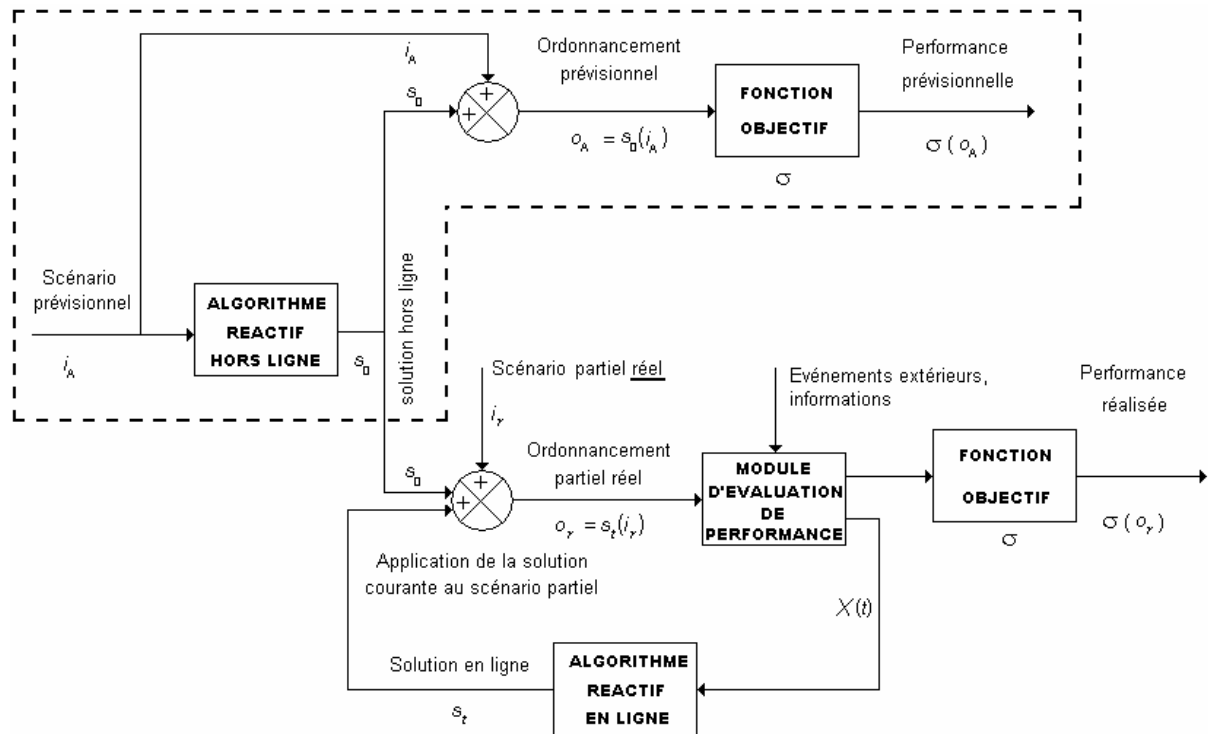


Figure 3 : Représentation systémique d'un problème d'ordonnancement réactif, sans prise en compte explicite du contexte incertain

Comme dans le cas d'un problème d'ordonnancement prédictif, seul le scénario prévisionnel i_A est pris en considération pour élaborer une solution hors ligne s_0 . Un algorithme réactif hors ligne est chargé de calculer cette solution. Le terme réactif signifie que la solution s_0 doit non seulement être performante pour l'instance i_A , mais également pouvoir être modifiée facilement par la suite, puisque cette possibilité existe. Le scénario réel noté i_r est pris en compte au fil du temps. i_r est donc un scénario réel partiel dépendant du temps car il n'est pas entièrement connu avant la fin de l'exécution de toutes les tâches, à $t = C_{max}$.

Par exemple, si la tâche T_1 précède la tâche T_2 , alors à $t = 0$, on ne connaît pas les durées p_1 et p_2 des deux tâches. Il existe une date $t_1 \geq p_1$ à laquelle p_1 est connue, mais p_2 reste inconnue. Ce n'est que lorsque les deux tâches T_1 et T_2 ont été exécutées que p_1 et p_2 sont connus.

Dans ces conditions, l'ordonnancement construit à partir de s_0 et de i_r est également partiel, et construit au fil de l'eau. Cet ordonnancement est noté $o_r = s_t(i_r)$, où s_t est initialement égale à la solution s_0 . Etant donné qu'il est possible de changer la solution courante, l'ordonnancement partiel o_r est fourni au « Module d'Evaluation de Performance » (MEP). Ce module est chargé de prendre la décision de conserver ou non la solution courante. La prise de cette décision s'appuie sur les performances de l'ordonnancement partiel courant, ainsi que sur des informations extérieures comme des prévisions éventuellement disponibles concernant le scénario, la stratégie choisie (on peut souhaiter changer très souvent de solution pour atteindre une performance donnée, ou au contraire chercher à minimiser les changements de solution), l'estimation de la durée d'une panne ou d'un retard de livraison. La façon dont fonctionne ce module n'est pas traitée ici, on se contentera de considérer que le MEP prend la

décision de conserver ou non la solution courante en fonction de l'ordonnancement partiel courant.

Si le MEP juge utile de changer la solution s_0 (qui a été calculée hors ligne), alors la boucle de rétroaction de la Figure 3 est utilisée. $X(t)$ représente cette décision, ainsi qu'un certain nombre d'informations utiles à un algorithme réactif en ligne chargé d'élaborer une nouvelle solution notée s_t . La solution calculée ne porte que sur l'ordonnancement des tâches qui n'ont pas encore été exécutées. s_t n'est donc appliquée qu'aux éléments encore inconnus du scénario $i(t)$. Notons que contrairement à l'élaboration de la solution hors ligne s_0 , le temps de calcul de la solution en ligne s_t doit satisfaire des contraintes temps réel. Ceci peut se traduire par la production de solutions de qualité inférieure.

Lorsque toutes les tâches à réaliser ont été exécutées, l'ordonnancement résultant est évalué à l'aide de la fonction objectif notée σ . Il est possible de comparer la performance réalisée et la performance prévisionnelle, mais cette comparaison aura peu de sens si le contexte de l'atelier a beaucoup changé au cours de l'exécution des tâches. En revanche, comparer la performance de l'ordonnancement obtenu par application de la solution s_0 au scénario réel $i(t)$ (à $t = C_{max}$) avec la performance de l'ordonnancement complet $o(t)$ permet de mesurer le gain de performance apporté par les changements successifs de solutions en ligne, à condition que s_0 reste admissible pour le scénario réel.

1.6 Anomalie de Graham

Considérons le problème du *bin packing*, qui est un problème bien connu en optimisation. Il s'agit de ranger une collection de n objets dans des boîtes, en minimisant le nombre de boîtes utilisées. Tout objet i est caractérisé par un scalaire x_i représentant, par exemple, sa masse. Les boîtes, quant à elles ont toutes la même capacité, c'est-à-dire que chaque boîte ne peut recevoir un sous-ensemble d'objets $\{i_1, \dots, i_k\}$ que si la masse totale des objets de ce sous-ensemble est inférieure à la capacité C de la boîte. On ne s'intéressera qu'au cas où $C \geq \max_i(x_i)$, car il n'existe pas de solution dans le cas contraire. Ce problème est connu pour être NP-difficile : il n'existe pas d'algorithme polynomial retournant une solution exacte. On va traiter une instance de ce problème à l'aide d'une heuristique appelée FFD (First Fit Decreasing), ou en français « algorithme du plus grand candidat ». Le principe de cette heuristique est le suivant : On traite les objets par poids décroissant. On range l'objet courant dans la première boîte pouvant le contenir sans dépasser sa capacité C . Si les boîtes utilisées pour ranger les objets précédents ne suffisent pas à recevoir l'objet courant, on utilise une nouvelle boîte. On continue jusqu'à ce que tous les objets aient été rangés dans une boîte.

On va considérer deux instances de ce problème, représentées par une liste d'objets à ranger et une capacité commune à toutes les boîtes :

Instance i_1 :

Liste = {44, 24, 24, 22, 21, 17, 8, 8, 6, 6}

$C = 60$

Instance i_2 :

Liste = {44, 24, 24, 22, 21, 17, 8, 8, 6, 6}

$C = 61$

La Figure 4 représente l'ordonnancement obtenu par application de la solution FFD sur l'instance i_1 . On constate que trois boîtes ont été nécessaires. De plus, toutes les boîtes sont utilisées à pleine capacité, l'ordonnancement obtenu est donc optimal, bien que la solution FFD soit une heuristique pour ce problème en général.

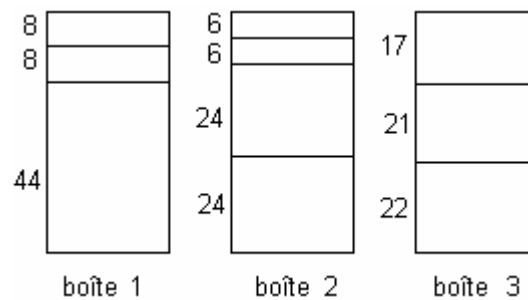


Figure 4 : Solution FFD appliquée à l'instance i_1

La Figure 5 représente l'affectation obtenue par application de la solution FFD sur l'instance i_2 . Les zones noires correspondent à la capacité non utilisée pour chaque boîte. On constate ici un phénomène de non régularité : l'instance i_2 est identique à l'instance i_1 , mais la contrainte sur la capacité des ressources (en l'occurrence il s'agit des boîtes) a été relâchée puisqu'on est passé d'une capacité de 60 à 61. Or on constate que les performances se sont dégradées. Le problème moins contraint nécessite plus de ressource (4 boîtes) que le problème contraint (3 boîtes). On voit pourtant qu'on aurait pu construire un ordonnancement n'utilisant que 3 boîtes pour l'instance i_2 en conservant le même rangement des produits que pour l'ordonnancement obtenu avec i_1 . Ce phénomène d'irrégularité est connu sous le nom d'anomalie de Graham, il constitue un des inconvénients majeurs des algorithmes de liste.

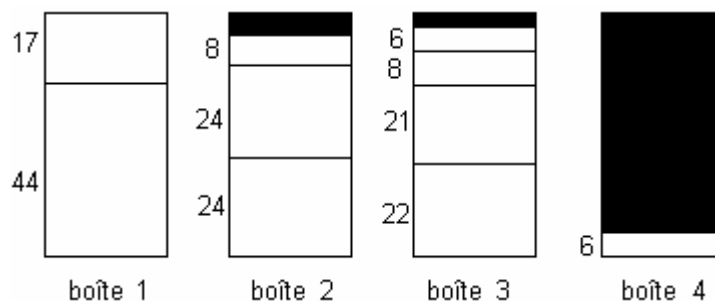


Figure 5 : Solution FFD appliqué à l'instance i_2

Au-delà du cas particulier des algorithmes de liste, cet exemple illustre la grande sensibilité des performances de la solution proposée aux variations de l'instance, ainsi que la nécessité de disposer de solutions moins sensibles aux perturbations des données.

1.7 Conclusion

Dans ce chapitre, la problématique générale de cette thèse a été posée : on s'intéresse à la résolution des problèmes d'optimisation en général, et aux problèmes d'ordonnancement en particulier, en contexte incertain. Notons qu'on ne s'intéresse pas ici aux problèmes où le critère de performance lui-même est incertain. On consultera par exemple l'étude de [Mahadev *et al.*, 1998] sur ce sujet. Le dernier exemple étudié illustre clairement la nécessité d'élaborer des solutions « robustes », c'est-à-dire dont les performances sont aussi peu sensibles aux perturbations que possible. Le chapitre suivant, qui complète la première partie de ce mémoire va permettre de préciser la notion de robustesse dans le cas de l'ordonnancement prédictif, ainsi que la notion de flexibilité qui intervient dans le cas réactif.

Chapitre II : Prise en compte explicite du contexte incertain

2.1 Introduction

L'objectif de ce chapitre est de présenter les outils développés dans la littérature pour résoudre les problèmes d'ordonnancement en contexte incertain, et plus particulièrement l'approche par scénarios qui est à la base de l'étude du cas industriel présenté dans la seconde partie de ce mémoire. C'est sur cette approche que l'on s'appuiera pour définir la robustesse d'une solution, c'est-à-dire son aptitude à garantir des performances en présence de perturbations. Comme dans le chapitre précédent, une approche systémique sera proposée pour synthétiser la démarche robuste. La notion de flexibilité proposée dans cette thèse sera présentée dans le même formalisme que la robustesse, afin d'être également une mesure de garantie de performance, mais exclusivement réservée au cas réactif. Enfin, l'analogie qui existe avec l'automatique en terme de garantie de performance sera soulignée.

2.2 Modèles de prise en compte de l'incertain

2.2.1 Les modèles de prise en compte de l'incertain

En ordonnancement classique toutes les données du problème (nombre, durée et disponibilité des tâches, validité des contraintes, disponibilité des ressources) sont supposées fixées et connues avec exactitude. Or le contexte où se posent les problèmes d'ordonnancement ne vérifie pas ces hypothèses de travail. Les entreprises sont sujettes à des aléas internes (pannes de machines, rebuts, accidents) et externes (retards d'approvisionnement, variation des prix, modification de plus en plus tardives et fréquentes des commandes des clients). En plus des aléas, certaines données ne sont connues qu'à travers des estimations (durée opératoire d'une tâche de diagnostic, volume de la production). Par conséquent, le caractère optimal d'un ordonnancement calculé sur des données qui ne décrivent pas exactement la réalité peut être remis en question. L'évolution du contexte peut être telle qu'un ordonnancement calculé à l'avance et soi-disant optimal peut n'être tout simplement plus applicable à l'atelier. Dans ces conditions, la prise en compte du contexte incertain est indispensable au traitement des problèmes d'ordonnancement réels. Cela n'interdit pas de calculer un ordonnancement prévisionnel, sachant que celui-ci devra éventuellement être adapté au cours de son déroulement. Dans [Herroelen et Leus, 2002], les approches de l'ordonnancement en contexte incertain sont rangées dans six classes décrites ci-dessous.

1. *L'ordonnancement purement réactif* : aucun ordonnancement prévisionnel n'est établi, et l'occurrence de perturbations se traduit par un ré-ordonnancement complet en ligne.
2. *L'ordonnancement stochastique* : aucun ordonnancement prévisionnel n'est établi, et les décisions sont prises en ligne, à des dates fixées à l'aide de données statistiques disponibles concernant les durées d'exécution. A chaque étape, une stratégie d'ordonnancement est appliquée. Le problème consiste à établir une stratégie qui minimise la durée de l'ordonnancement.
3. *Les réseaux de projets stochastiques* : dans cette approche, une tâche est représentée par un arc (i,j) pondéré par un vecteur (p_{ij}, F_{ij}) , où p_{ij} désigne la probabilité que la tâche (i,j) soit exécutée, sachant que l'événement i s'est produit. F_{ij} est la densité de probabilité conditionnelle de la durée (positive) d_{ij} de la tâche (i,j) , sachant que cette tâche est exécutée.
4. *L'ordonnancement flou* : cette approche est utilisée lorsque l'aspect répétitif de l'exécution des tâches n'est pas suffisant pour disposer d'un historique permettant de déduire des lois de probabilité fiables. La durée des tâches étant estimée, on a affaire à des données imprécises, pour lesquelles la représentation à l'aide de nombres ou d'intervalles flous est tout à fait adaptée.
5. *L'ordonnancement robuste* : un ordonnancement prévisionnel est construit hors-ligne, mais en prenant explicitement en compte le contexte incertain, de sorte que l'ordonnancement prévisionnel peut rester valable pour certains types de perturbations. On parle aussi d'*ordonnancement proactif*.
6. *L'analyse de sensibilité* constitue une classe à part entière dans [Herroelen et Leus, 2002]. Cette approche apparaîtra comme un outil dans la mise en oeuvre de l'ordonnancement robuste (lorsque la notion de robustesse aura été définie). L'analyse de sensibilité ne constitue pas une approche de résolution à part entière.

Notons toutefois que cette classification ne distingue pas nettement la modélisation de l'incertain, et la résolution ou (l'exploitation) des représentations qui en résultent. En effet, on peut distinguer trois classes de modèles de l'incertain : les modèles stochastiques, les modèles par intervalles et les modèles par scénarios, alors que dans [Herroelen et Leus, 2002], on compte six approches différentes basées sur ces modèles de m'incertain.

Comparer les différentes approches est difficile. Toutefois, on peut considérer la modélisation de l'ordonnancement en contexte incertain comme une tentative d'exploiter le plus d'information possible sur le fonctionnement du système étudié et les aléas ou les incertitudes qui en affectent le fonctionnement. Les approches purement réactives sont un cas particulier où aucune information *a priori* n'est exploitée pour piloter le système. La diversité des formes que prend cette information *a priori* se retrouve dans la diversité des méthodes de prise en compte de l'incertain. Leurs performances sont difficilement comparables dans la mesure où elles ne nécessitent pas la même quantité (ni la même qualité) d'information.

Cette thèse est consacrée à la recherche de solutions au problème d'ordonnancement en contexte incertain, avec garantie de performance. Pour cette raison, l'étude de l'atelier de photolithographie présentée dans la partie II peut être rangée dans la catégorie « ordonnancement robuste ». Notre choix de modélisation de l'incertain s'est porté sur l'approche par scénarios, dont la présentation fait l'objet du paragraphe suivant. L'étude de [Herroelen et Leus, 2002] confirme que cette modélisation est un outil privilégié dans la littérature lorsqu'on souhaite garantir des performances sous certaines hypothèses concernant

l'incertain. Son utilisation est illustrée dans [Kouvelis *et al.*, 2000] ou encore [Vincke, 1999a] et [Vincke, 1999b]. Lorsqu'on dispose d'informations concernant l'amplitude, la fréquence ou plus généralement l'impact des perturbations sur les données du problème, des méthodes tirant parti de ces informations sont disponibles. Les méthodes stochastiques sont adaptées lorsque ces informations sont disponibles sous forme de statistiques. [Zhang et Graves, 1997] résolvent un problème avec garantie de performance dans ce cadre. On peut aussi évaluer l'impact des perturbations sur la précision des données à l'aide des intervalles flous. Une telle approche est illustrée dans [Bourgade, 1997].

2.2.2 Approche par scénarios et analyse de risques

L'approche par scénarios est une méthode de modélisation de l'incertain consistant à considérer non pas un seul jeu de données, mais un ensemble de jeux de données pour un problème d'ordonnancement, avec comme objectif la détermination d'une ou plusieurs solutions qui garantissent des performances sur l'ensemble des jeux de données. Un scénario a déjà été défini dans le chapitre I de cette thèse. On rappelle qu'un scénario contient un jeu de données numériques associées à des variables du modèle d'optimisation que l'on suppose susceptible de changer. Le contenu des scénarios est donc le reflet d'une modélisation et d'hypothèses concernant l'incertain.

En plus de la définition du contenu des scénarios, la modélisation de l'incertain dans cette approche nécessite la construction d'un ensemble de scénarios, noté I_A , qui reflète l'étendue des données sur lesquelles une garantie de performance de la solution est souhaitée par l'utilisateur. I_A peut être un ensemble continu ou discret, dénombrable ou non. Cet ensemble regroupe tous les scénarios pour lesquels on souhaite trouver une solution garantissant des performances. Ainsi, I_A représente les risques que l'on souhaite couvrir. Les hypothèses faites concernant l'incertain s'expriment, elles, dans le contenu des scénarios, puisque les valeurs numériques portées par les scénarios ne sont relatives qu'à des données supposées incertaines.

2.2.3 Définitions de la performance dans le contexte de l'approche par scénarios

Dès lors que l'on considère plusieurs scénarios pour calculer une solution, ce que suppose l'approche par scénarios, il faut s'attendre à ce que cette solution conduise à des ordonnancements différents, dont les performances seront elles aussi a priori différentes. Un exemple d'une telle situation est donné dans ce chapitre. Comme on a distingué clairement solution et ordonnancement, il faut également distinguer la performance d'une solution de la performance d'un ordonnancement.

2.2.3.1 Performance locale

La fonction objectif évalue la performance d'un ordonnancement. Etant donné qu'une même solution permet d'obtenir différents ordonnancements sur un ensemble d'instances, on peut dire que la fonction objectif ne suffit pas à mesurer la performance d'une solution. On parlera de performance locale pour désigner la fonction objectif. Par exemple, il est clair que la fonction objectif C_{max} dépend de la durée des tâches, c'est-à-dire qu'elle est liée à des valeurs numériques qui proviennent d'un scénario i sur lequel est appliquée la solution s .

Comme un ordonnancement est l'application d'une solution à une instance, la valeur de la fonction objectif dépend aussi de la solution.

La Figure 6 donne une représentation de l'approche par scénarios. Une solution s est appliquée à chacun des scénarios de I_A , ce qui produit un ensemble d'ordonnements. La performance locale de chacun de ces ordonnancements est mesurée par la fonction objectif, notée σ . On obtient alors un ensemble de performances locales, ce qui montre clairement que la performance locale ne suffit pas à évaluer la qualité d'une solution s dès lors que I_A compte plus d'un scénario, c'est-à-dire lorsqu'on est en contexte incertain.

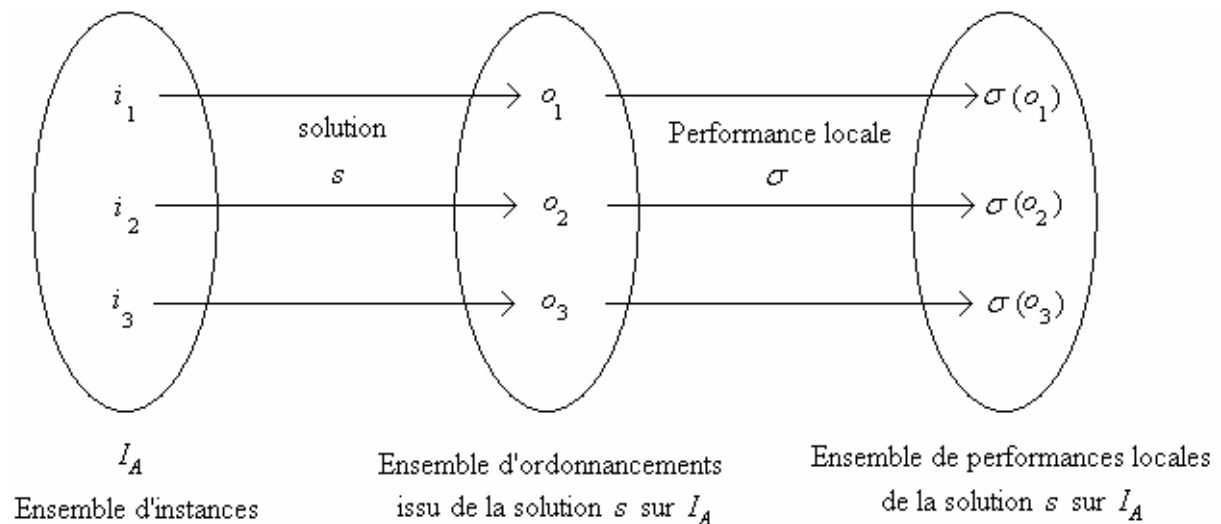


Figure 6 : Illustration de l'approche par scénarios

2.2.3.2 Performance globale

Un ordonnancement est un calendrier figé du déroulement des tâches à exécuter, associé à une affectation précise de chaque tâche à une ou plusieurs ressources. La fonction objectif est adaptée à l'évaluation de la performance d'un ordonnancement, mais si l'on s'intéresse à l'évaluation d'une solution sur un ensemble d'instances, d'autres outils doivent être développés. Ces outils sont les critères de performance globale d'une solution, par opposition à la fonction objectif qui est l'évaluation « locale » d'une solution sur une instance particulière, c'est-à-dire d'un ordonnancement. Soulignons que les critères qui vont être présentés n'ont d'intérêt qu'en contexte incertain. Les critères de performance globale qui apparaissent le plus souvent dans la littérature ([Kouvelis et Yu, 1997], [Majhoub, 2000], [Rossi et Jacomino, 2001], [Groupe flexibilité du Gotha, 2002]) sont présentés ci-après. On les appelle parfois aussi *métriques*. La performance globale mesure la qualité d'une solution s à un problème d'ordonnement, sur un ensemble d'instances I_A . Les indicateurs de performance globale définis ci-après sont basés sur la performance locale (c'est-à-dire la fonction objectif), notée σ dans les formules qui suivent :

- Le *Pire Cas* noté $W(s, \sigma, I_A)$: ce critère correspond à la plus mauvaise performance locale mesurée par la fonction objectif σ obtenue en appliquant la solution s à toutes les instances de l'ensemble I_A .

$$W(s, \sigma, I_A) = \max_{i \in I_A} (\sigma(s(i)))$$

L'expression du pire cas W donnée ci-dessus correspond au cas où le critère de performance locale doit être minimisé (c'est vrai par exemple pour $\sigma = C_{\max}$). Lorsque le critère de performance doit être maximisé (lorsque σ mesure un gain par exemple), il faut remplacer la fonction *max* par la fonction *min* dans l'expression de W .

- La *Variabilité* notée $V(s, \sigma, I_A)$: ce critère correspond à la différence entre la plus grande et la plus petite valeur de σ sur I_A .

$$V(s, \sigma, I_A) = \max_{i \in I_A} (\sigma(s(i))) - \min_{i \in I_A} (\sigma(s(i)))$$

On peut ainsi connaître le plus grand écart de performance attendu, ce qui est particulièrement important en ordonnancement cyclique.

- La *Sensibilité* notée $\Sigma(s, \sigma, I_A)$: ce critère mesure le plus grand écart observé, pour toute instance i de I_A entre la performance de la solution sur l'instance i , et la meilleure performance que l'on aurait pu atteindre avec une solution adaptée à cette instance.

$$\Sigma(s, \sigma, I_A) = \max_{i \in I_A} \left(\frac{\sigma(s(i))}{\sigma(s^*(i))} \right)$$

Où $s^*(i)$ représente la valeur optimale (au sens de la performance locale mesurée par la fonction objectif σ) associée au scénario i . La sensibilité de la solution s sur I_A mesure la valeur maximale du rapport de la performance de l'ordonnancement produit par application de s sur i , sur la meilleure performance atteignable sur i . Autrement dit, la sensibilité mesure l'écart maximal de performance par rapport à la solution optimale. L'évaluation de ce critère nécessite de disposer de la performance de l'ordonnancement optimal associée à chaque instance de I_A . Or l'obtention de cette information peut être un problème NP-difficile, même pour une seule instance. La sensibilité n'est donc pas toujours un critère facile à utiliser.

Notons enfin que ces critères de performance globale doivent être considérés comme les critères de performance usuels en ordonnancement : leur pertinence dépend du problème considéré et des critères de qualité de l'ordonnancement définis par la stratégie de pilotage du système de production considéré.

Le Tableau 1 présente les différences fondamentales entre performance locale et performance globale. La performance locale mesure les qualités de l'ordonnancement issu de la solution sur une instance, la performance globale mesure les qualités de la solution sur un ensemble d'instances. Ces deux notions de performance ne sont pas indépendantes, puisque la définition de la performance globale est basée sur un critère de performance locale.

	Objet de la mesure	Domaine d'application
Performance locale	Ordonnancement o	Instance i
Performance globale	Solution s	Ensemble d'instance I_A

Tableau 1 : Performance locale et performance globale

2.3 Ordonnancement prédictif : robustesse

2.3.1 Robustesse d'une solution

Les critères de performance globale permettent de juger de la qualité d'une solution sur un ensemble d'instances. La définition de la robustesse d'une solution traduit la garantie de performance offerte par cette solution sur un ensemble d'instances I_A , en s'appuyant sur ces critères globaux.

Dans la définition qui suit, λ_k désigne un critère de performance globale (le pire cas, la variabilité ou la sensibilité). Le réel c_k est un niveau de performance globale désiré associé au critère λ_k . On dira qu'une solution s est robuste sur I_A si elle garantit une performance globale au moins égale à c_k . On pourra par exemple décider qu'une solution est robuste si elle garantit que moins de 20 % des tâches sont en retard sur un ensemble de scénarios donné.

Définition 1: Etant donné un critère de performance locale σ , un critère de performance global λ_k et un niveau de performance globale à garantir noté c_k , on dit que la solution s est *robuste* sur l'ensemble d'instances I_A si et seulement si :

$$\lambda_k(s, \sigma, I_A) \leq c_k \quad (1)$$

Le signe inférieur ou égal est utilisé dans la définition car les trois critères de performance globale sont à minimiser. La définition de la robustesse est donc bien équivalente à une garantie de performance de la solution s (et non d'un ordonnancement particulier) sur un ensemble d'instances I_A représentant l'incertitude sur les données.

Notons que la notion de robustesse n'est pas une propriété intrinsèque à une solution. La formule (1) montre clairement que la robustesse d'une solution s donnée dépend du critère de performance globale choisi λ_k , du niveau de performance globale c_k exigé par l'utilisateur, de l'ensemble d'instances I_A considéré (c'est-à-dire du risque que l'on veut couvrir), et enfin du critère de performance locale σ qui permet d'évaluer la qualité des ordonnancements produits par la solution s sur I_A .

Le Tableau 2 regroupe les quatre paramètres qui caractérisent la notion de robustesse d'une solution s donnée.

Paramètre	Action sur la notion de robustesse
I_A	Hypothèses et risques à couvrir
λ_k	Mesure de la performance globale de la solution s
c_k	Performance globale à garantir
σ	Mesure de la performance locale des ordonnancements produits par s

Tableau 2 : Action des paramètres de la définition de la robustesse

Ainsi, il peut ne pas exister de solution robuste au sens de cette définition si le niveau de performance globale attendu est trop élevé ou si l'ensemble d'instances I_A est trop vaste, comme il peut exister plusieurs solutions robustes, au point de rendre inutile la définition de la robustesse si le niveau de performance attendu est trop modeste, ou si l'ensemble d'instances est trop réduit, de sorte que toute solution admissible est robuste.

Les définitions que l'on trouve dans [Kouvelis et Yu, 1997] et [Mahjoub, 2000] sont des approches min-max : une solution n'est robuste au sens de ces définitions que si elle maximise le critère de performance globale choisi. On préfère considérer ici que c'est à l'utilisateur de spécifier le niveau de performance à garantir qu'il juge pertinent pour qualifier une solution de robuste. Enfin, cette définition permet de conserver plusieurs solutions robustes différentes, ce qui peut être intéressant dans le cas de l'ordonnement réactif.

2.3.2 Représentation ensembliste de la problématique de la robustesse

Dans ce paragraphe, on propose une représentation ensembliste du problème d'ordonnement afin de synthétiser les différences entre l'ordonnement en contexte certain et l'ordonnement avec prise en compte explicite du contexte incertain. Cette représentation permettra de définir les trois problèmes de robustesse qui découlent de la définition 1.

2.3.2.1 Ordonnement en contexte certain

Ce problème consiste à déterminer une solution optimale s (au sens de la fonction objectif σ choisie) connaissant l'instance i du problème. La Figure 7 montre que l'ensemble d'instances est réduit à une instance unique. L'ordonnement produit par s n'a pas été représenté, seule son évaluation par la fonction objectif apparaît.

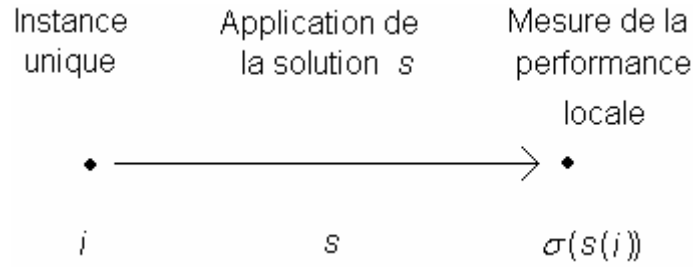


Figure 7 : ordonnancement en contexte certain

2.3.2.2 Ordonnancement avec prise en compte explicite du contexte incertain

Sur la Figure 8, la prise en compte du contexte incertain se traduit par la considération d'un ensemble d'instances I_A . La solution s appliquée à cet ensemble produit des ordonnancements dont la performance mesurée par la fonction objectif σ est représentée par un segment. L'ensemble des valeurs $\sigma(s(i))$ pour tout i dans I_A est noté $\sigma(s(I_A))$, mais ce n'est pas nécessairement un segment (en particulier si I_A est un ensemble fini dénombrable). Cette représentation illustre également la mise en oeuvre des critères de performance globale V (la variabilité) et W (le pire cas). La sensibilité Σ se prête mal à une représentation graphique.

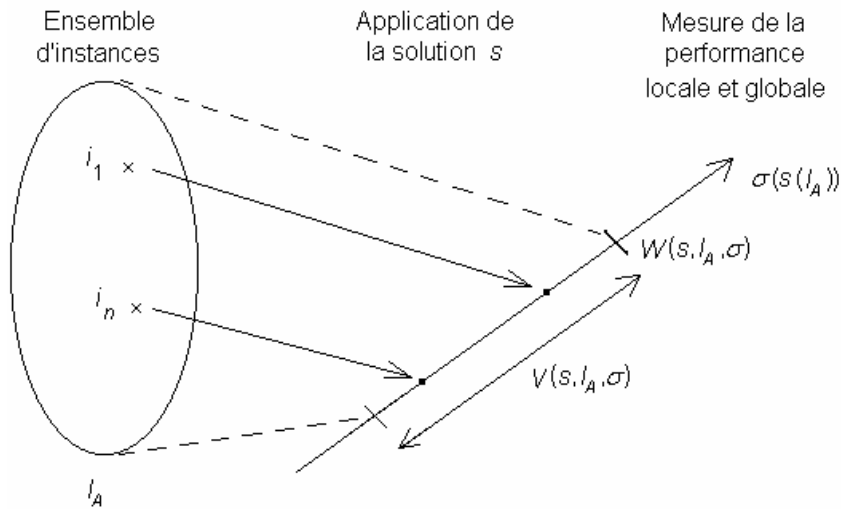


Figure 8 : ordonnancement en contexte incertain

La Figure 9 est une représentation plus synthétique de la Figure 8, où apparaissent seulement l'ensemble I_A des instances considéré, la solution s et la performance globale de s sur I_A , notée $\lambda_k(s, I_A, \sigma)$. Il s'agit de la performance globale de s sur I_A , mesurée à l'aide d'un critère de performance globale noté λ_k dans la définition 1.

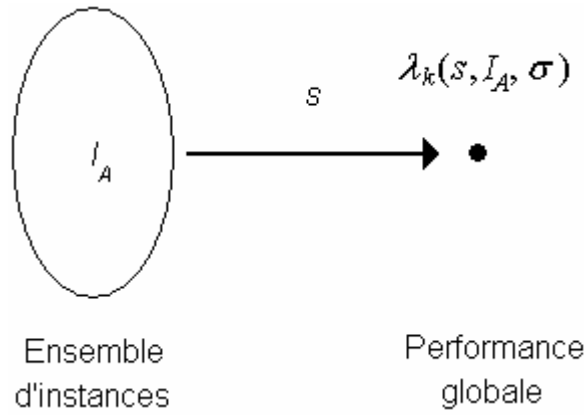


Figure 9 : Représentation simplifiée de la Figure 8

2.3.2.3 Les trois problèmes de robustesse

Connaissant I_A et $\lambda_k(s, I_A, \sigma)$, le premier problème de robustesse consiste à déterminer s , une solution robuste sur I_A , assurant un niveau de performance suffisant (au sens de λ_k). Par exemple, pour toute commande dont le temps d'exécution est à moins de 20 % d'une commande de référence attendue, on souhaite garantir que la performance de la solution ne soit pas dégradée de plus de 5%.

Dans le second problème, I_A et s sont connus et on cherche à déterminer $\lambda_k(s, I_A, \sigma)$, c'est-à-dire la garantie de performance offerte par s sur I_A . C'est l'objet par exemple de l'étude de sensibilité conduite par [Penz *et al.*, 2001], dans laquelle une relation entre l'étendue de l'ensemble d'instances et la sensibilité d'une solution particulière est établie.

Enfin, le troisième problème de robustesse suppose connus s et $\lambda_k(s, I_A, \sigma)$, et vise à déterminer I_A , c'est-à-dire l'ensemble des instances pour lesquelles la solution s est robuste (au sens de λ_k). On recherche par exemple le plus petit retard de livraison d'un composant qui se traduit par une dégradation de performance de s de 5 % ou plus. Un tel problème est abordé dans la partie II de ce mémoire.

2.3.3 Exemple de mise en œuvre

L'exemple suivant est proposé pour illustrer l'utilisation des outils présentés dans cette partie. Il s'agit du problème d'affectation de $m = 9$ machines à $n = 3$ produits à fabriquer. La description de ce problème a été donnée au paragraphe 1.3. Ici, on se propose de compléter cet exemple de manière à illustrer l'approche par scénarios et la définition de la robustesse.

La commande i passée à l'atelier n'est pas entièrement connue au moment où les machines doivent être affectées aux produits. On considère un ensemble I_A de commandes, qui constituent les instances (ou scénarios) du problème, et l'on cherche à évaluer la robustesse de quelques solutions pour cet ensemble de commandes défini par :

$$I_A = \left\{ i_1 = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}, i_2 = \begin{bmatrix} 50 \\ 150 \\ 250 \end{bmatrix}, i_3 = \begin{bmatrix} 350 \\ 50 \\ 50 \end{bmatrix}, i_4 = \begin{bmatrix} 200 \\ 50 \\ 200 \end{bmatrix} \right\}$$

2.3.3.1 Calcul d'une solution optimale sans prise en compte de l'incertain

On note x_j le nombre de machines affectées à la classe j et $i(j)$ la quantité de travail imposée aux machines de la classe j par la commande i . Une solution optimale à ce problème est un triplet d'entiers $\{x_1, x_2, x_3\}$ tel que C_{max} est minimum. On formalise ce problème de la façon suivante :

$$\begin{cases} \sum_{j=1}^n x_j = m \\ x_j \geq 1 \quad (\forall j \in [1, n]) \\ \min \left(\max_{j \in [1, n]} \left(\frac{i(j)}{x_j} \right) \right) \end{cases}$$

L'algorithme permettant d'obtenir la solution optimale de ce problème est donnée dans [Rossi, 2000]. Ces solutions s'obtiennent en temps polynomial. Les solutions optimales associées aux instances i_1, i_2, i_3 et i_4 sont notées s_1, s_2, s_3 et s_4 respectivement :

$$s_1 = \{3, 3, 3\} \quad s_2 = \{1, 3, 5\} \quad s_3 = \{7, 1, 1\} \quad s_4 = \{4, 1, 4\}$$

On examinera également la solution s_5 définie par :

$$s_5 = \{4, 2, 3\}$$

Cette solution n'est optimale pour aucune des quatre instances considérées. s_5 sera comparée aux quatre solutions optimales s_1, s_2, s_3 et s_4 . Le Tableau 3 regroupe les performances des solutions s_1 à s_5 lors de leur application à chacun des scénarios de l'ensemble I_A .

2.3.3.2 Evaluation des performances locales des solutions

C_{max}	$i_1 = \begin{bmatrix} 150 \\ 150 \\ 150 \end{bmatrix}$	$i_2 = \begin{bmatrix} 50 \\ 150 \\ 250 \end{bmatrix}$	$i_3 = \begin{bmatrix} 50 \\ 50 \\ 350 \end{bmatrix}$	$i_4 = \begin{bmatrix} 200 \\ 50 \\ 200 \end{bmatrix}$
$s_1 = \{3,3,3\}$	50	83.3	116.6	66.6
$s_2 = \{1,3,5\}$	150	50	350	200
$s_3 = \{7,1,1\}$	150	250	50	200
$s_4 = \{4,1,4\}$	150	150	87.5	50
$s_5 = \{4,2,3\}$	75	83.3	87.5	66.6

Tableau 3 : Performance des ordonnancements produits par application des solutions aux instances

Les résultats présentés au Tableau 3 confirment que la solution s_5 n'est optimale pour aucune instance appartenant à I_A : la valeur du critère C_{max} est toujours supérieure à 50 unités de temps pour les ordonnancements produits par s_5 , alors que pour toute commande, il existe un ordonnancement tel que $C_{max} = 50$.

2.3.3.3 Evaluation des performances globales des solutions

Les trois critères de performance globale peuvent être évalués sur I_A , pour chaque solution. Les résultats apparaissent dans le Tableau 4.

	Pire cas sur I_A (W)	Variabilité sur I_A (V)	Sensibilité sur I_A (Σ)
$s_1 = \{3,3,3\}$	116.6	66.6	2.3
$s_2 = \{1,3,5\}$	350	300	7
$s_3 = \{7,1,1\}$	250	200	5
$s_4 = \{4,1,4\}$	150	100	3
$s_5 = \{4,2,3\}$	87.5	20.8	1.6

Tableau 4 : Critères de performance globale sur I_A pour chaque solution

A partir de ces résultats, on peut mesurer la robustesse de chaque solution sur I_A , à condition de choisir un critère de performance global λ_k ainsi qu'un niveau de performance globale attendu c_k .

2.3.3.4 Détermination des solutions robustes

Si le critère de performance globale sur I_A est W (le pire cas), et que l'on souhaite garantir une valeur inférieure à 50 pour ce critère, alors aucune des solutions proposées n'est robuste. Si ce niveau est porté à 120, alors les solutions s_1 et s_5 sont robustes. Notons qu'il ne s'agit pas de « choisir la solution la plus robuste » mais de déterminer celles qui assurent une garantie de performance suffisante pour l'utilisateur : si l'on envisage de changer la solution en ligne, il est important de considérer aussi sa flexibilité. On peut cependant remarquer que sur cet exemple, la performance globale de la solution s_5 est toujours supérieure à celle des autres, quel que soit le critère de performance globale choisi, alors que s_5 n'est optimale pour aucune instance de I_A . Ceci montre bien qu'une solution performante sur un ensemble d'instances (au sens de la performance globale) n'est pas nécessairement à rechercher parmi les solutions optimales sur une instance de cet ensemble (c'est-à-dire au sens de la performance locale). Ceci souligne que la détermination d'une solution robuste nécessite une approche différente (ou au moins une reformulation du critère de performance) de la version « en contexte certain » du problème d'ordonnancement initial. Cependant, aucune méthode générale n'est donnée dans cette partie pour obtenir de solution robuste.

2.3.4 Représentation systémique des problèmes d'ordonnancement avec modélisation de l'incertain

2.3.4.1 Ordonnancement prédictif

Dans ce paragraphe, on propose une représentation systémique des problèmes d'ordonnancement prédictif avec modélisation du contexte incertain (voir Figure 10). Cette démarche se caractérise par la prise en compte d'un ensemble de scénarios prévisionnels noté I_A dans le calcul d'une solution hors ligne s . Il ne s'agit plus de calculer une solution dont la performance locale est bonne sur une instance particulière : il s'agit de calculer une solution dont la performance globale est bonne sur toute instance de I_A . Si une telle solution peut être trouvée, elle garantit des performances globales à condition que le scénario réel i_r soit dans I_A . L'ordonnancement réel o_r est créé en ligne, lors de l'application de la solution s à l'instance réelle i_r , où l'on fait l'hypothèse que i_r ne remet pas en cause l'admissibilité de s .

Une fois que la solution s a été calculée, elle n'est pas remise en cause, quelle que soit l'instance effective i_r . Deux cas sont possibles :

1. L'instance i_r appartient à I_A . Dans ce cas, la performance locale réalisée sera conforme aux prévisions établies hors ligne, et exprimées en terme de performance globale. Autrement dit, si la performance globale de la solution s a été jugée suffisante, alors il en sera de même de la performance locale réalisée.
2. L'instance i_r n'appartient pas à I_A . Dans ce cas il est possible que la performance locale réalisée soit mauvaise, car la solution s n'apporte aucune garantie sur i_r . Etant donné qu'il n'est pas possible de modifier la solution s en ligne, l'utilisateur ne peut que constater l'écart de performance provoqué par l'occurrence de i_r .

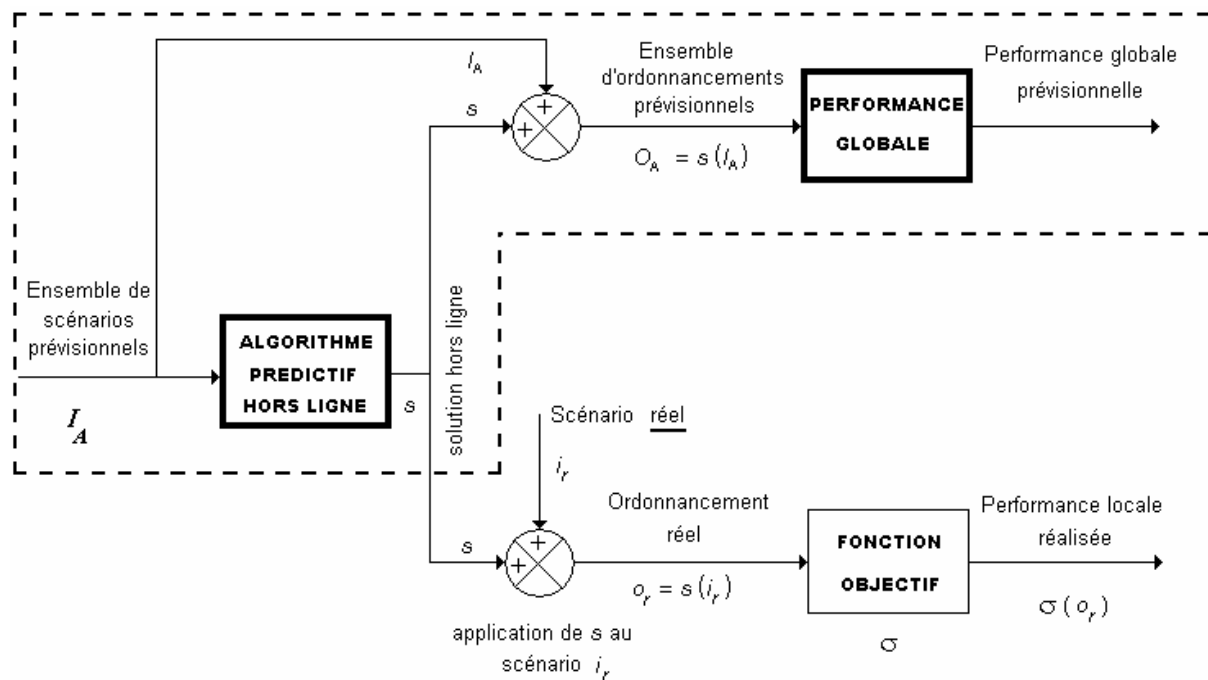


Figure 10 : Représentation systémique d'un problème d'ordonnancement prédictif, avec modélisation de l'incertain par l'approche par scénarios

Ceci illustre le fait que la performance globale d'une solution sur un ensemble d'instances I_A mesure une garantie de performance. Les modalités de cette garantie sont décrites dans la définition de la robustesse.

Le fonctionnement décrit par la Figure 10 permet de représenter :

- Les travaux de [Rossi et al., 2001] sur la configuration d'un parc de machines de photolithographie. Ce problème consiste à déterminer une solution s (une configuration) telle que son application à toute instance de I_A conduit à une performance locale garantie. L'ensemble des instances représente l'incertitude qui pèse sur la commande effectivement passée à l'atelier. Ce problème fait l'objet de la partie II de ce mémoire.
- L'étude de sensibilité menée par [Penz et al., 2001]. Les auteurs déterminent un voisinage centré sur une instance nominale dans lequel la performance locale de tout ordonnancement issu de la solution s reste borné par la racine carré de l'amplitude des perturbations. Le voisinage en question est un ensemble d'autant plus vaste que les perturbations de l'instance nominale sont de grande amplitude. La connaissance de cette borne permet de garantir les performances de la solution s si l'amplitude des perturbations reste inférieure à une valeur donnée.

Notons que dans ces deux cas, l'ensemble des scénarios considérés est continu (donc de cardinalité infinie), et non dénombrable. L'approche par scénarios n'est donc pas réservée aux problèmes où l'incertain se réduit à un ensemble de scénarios discrets en nombre fini.

2.3.4.2 Ordonnancement réactif

Dans le cas d'un problème d'ordonnancement réactif avec modélisation de l'incertain par l'approche par scénarios (voir Figure 11), l'algorithme réactif hors ligne reçoit un ensemble d'instances attendues I_A et non plus une instance unique. La solution hors ligne s_0 doit être performante si elle est appliquée à une instance appartenant à I_A , elle doit également pouvoir être changée si ce n'est pas le cas, et que les performances de s sont jugées insuffisantes. La prise en compte du contexte incertain peut également se traduire par le calcul hors ligne d'un ensemble de solutions potentielles adaptées à différents sous-ensembles de scénarios. Ces solutions sont alors disponibles lors du passage dans la boucle de rétroaction, ce qui permet de limiter le rôle de l'algorithme réactif en ligne au choix de la solution pré calculée la mieux adaptée au scénario partiel courant $i(t)$. Cette modélisation illustre l'approche de [Dean et Greenwald, 1998], qui appliquent cette démarche à la conception d'un ordonnancement de tâches en temps réel dans le contexte de l'avionique.

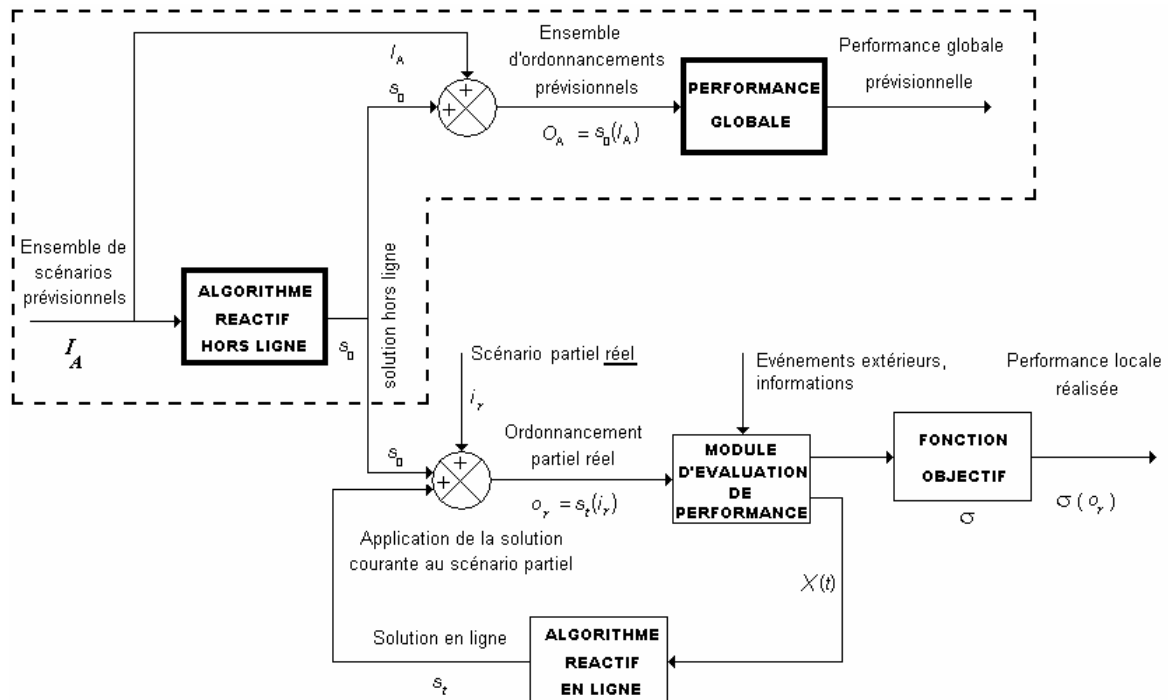


Figure 11 : Représentation systémique d'un problème d'ordonnancement réactif, avec modélisation de l'incertain par l'approche par scénarios

La considération d'un ensemble de scénarios I_A lors de l'élaboration de la solution hors ligne permet de réserver l'utilisation de l'algorithme réactif en ligne aux scénarios n'appartenant pas à I_A . En effet, la solution s_0 est censée conduire à des ordonnancements suffisamment performants pour tout scénario appartenant à I_A , il n'est donc pas nécessaire de modifier cette solution dans ce cas. Un changement de solution n'est rendu nécessaire que par l'occurrence d'un scénario hors de I_A ; il s'agit donc d'un événement plutôt « rare » si I_A et s ont été bien choisis.

2.4 Ordonnancement réactif : flexibilité

2.4.1 Introduction et définitions

Le mot flexibilité est souvent utilisé dans des contextes différents. On parle de la flexibilité d'un équipement pour souligner ses capacités multifonctions. Un atelier flexible désigne un atelier capable de fabriquer différents types de produits. On trouve dans [Erol, 1997] et [Erol, 1999] une étude très générale de la flexibilité, où cette notion est définie comme « l'aptitude d'un système à répondre aux modifications de son environnement afin d'assurer le respect de ses objectifs. ». Plus précisément, les modifications de l'environnement dont il est question comprennent aussi bien l'incertitude (ce mot désigne les aléas internes et externes dans cette définition) que la variabilité, c'est-à-dire la variation de la gamme de produits mais aussi la variation des quantités à produire. Cette dernière définition est très générale, et elle ne distingue pas les cas prédictifs et réactifs, ce qui est important dans le contexte de l'ordonnancement. De plus, l'objet qui offre cette flexibilité n'est pas précisé. Or le sens du mot flexibilité n'est pas le même quand il est question de la flexibilité d'une machine ou de la flexibilité d'une stratégie de pilotage d'atelier.

Si l'on se restreint à la flexibilité manufacturière, la classification de [Browne *et al.*, 1984] distingue huit classes de flexibilité, qui recouvrent encore des aspects très divers et hétérogènes.

1. Flexibilité machine : capacité à opérer facilement les réglages nécessaires à un changement de pièces à produire.
2. Flexibilité produit : aptitude du système de production à être adapté à de nouvelles gammes de produits.
3. Flexibilité process : capacité à produire un ensemble de types de pièces sans effectuer de changement important.
4. Flexibilité sur les ordres : capacité à interchanger l'ordre de plusieurs opérations pour chaque type de pièces.
5. Flexibilité de la gamme opératoire : capacité à réorienter les pièces devant passer une machine en panne vers les autres machines.
6. Flexibilité du volume de production : capacité à maintenir un coût de revient des produits indépendant du volume produit.
7. Flexibilité d'expansion : capacité à adapter facilement la capacité de production.
8. Flexibilité de production : capacité à produire des produits différents dans ajouter d'équipement supplémentaire.

Dans [Groupe Flexibilité du Gotha, 2002] la flexibilité est exprimée comme l'existence de modifications possibles d'un ordonnancement calculé hors ligne entraînant une perte de performance acceptable. Les modifications dont il est question sont destinées à répondre aux aléas. Les différents niveaux de flexibilité, énumérés ci-dessous, peuvent être vus comme des sous-classes de la classification de Browne : dans [Groupe Flexibilité du Gotha, 2002], seule la flexibilité dans le cadre de l'ordonnancement est étudiée. En contrepartie, les différents niveaux de flexibilité sont plus homogènes, et sont clairement hiérarchisés.

1. Flexibilité sur les temps : seules les dates effectives de début et de fin des tâches peuvent varier.
2. Flexibilité sur les ordres : les ordres relatifs d'exécution (séquences) d'un ensemble de tâches peuvent être modifiées durant l'exécution, ce qui implique la flexibilité sur les temps. Cet ensemble de tâches doit utiliser une même ressource qui oblige à les exécuter séquentiellement. Les ressources utilisées par chaque tâche restent donc inchangées.
3. Flexibilité sur les ressources : il est possible de changer l'affectation des ressources aux tâches. Il est clair que la flexibilité sur les ressources implique les deux premières.
4. Flexibilité sur les modes d'exécution : suivant le contexte à d'exécution, on peut décider, au prix peut être d'une dégradation de performances, de changer le mode d'exécution d'une ou de plusieurs tâches. En présence d'aléas, on permettra ainsi la préemption, la duplication, le recouvrement des exécutions par les communications, le changement de gamme, voire la délocalisation.

La méthode ORABAID a été développée au LAAS successivement par [Demmou, 1977], [Thomas, 1980], [Le Gall, 1989], [Billaut, 1993] et [Artigues, 1997]. Elle consiste à proposer différents ordonnancements flexibles pour le pilotage d'atelier, la flexibilité ainsi offerte pouvant être exploitée pour faire face à des aléas. La méthode ORABAID se décompose en deux étapes.

1. Au cours de l'étape hors ligne, appelée *phase d'analyse*, les solutions admissibles sont caractérisées et stockées.
2. Au cours de l'étape en ligne, appelée *phase de pilotage temps réel*, l'ensemble des solutions admissibles calculé hors ligne est mis à profit pour faire face aux aléas : lorsqu'un événement inattendu se produit, les actions proposées au décideur sont fondées sur l'étude menée hors ligne.

La méthode ORABAID est tout à fait adaptée à la définition de la flexibilité émanant du [Groupe Flexibilité du Gotha, 2002].

2.4.2 Comparaison de la flexibilité avec ORABAID et l'approche par scénarios.

Une littérature abondante porte sur les points 2 et 3 de la classification du [Groupe Flexibilité du Gotha, 2002]: dans la méthode ORABAID, la flexibilité statique d'une « séquence de groupes » est directement proportionnelle au nombre de solutions qui découlent de cette séquence de groupe. Un groupe est un ensemble de tâches totalement permutable. Une séquence de groupe est associée à chaque ressource, ce qui constitue ainsi une structure de solutions à un problème d'ordonnancement. [Esswein *et al.*, 2003] font l'hypothèse que la flexibilité d'une telle séquence est inversement proportionnelle au nombre de groupes présents dans les séquences. Ainsi, le nombre de permutations est élevé dans chaque groupe, ce qui conduit à une grande flexibilité. Les auteurs cherchent alors à construire une séquence de groupes ayant un nombre minimum de groupes, tout en maîtrisant la dégradation de la performance des solutions caractérisées par cette séquence.

La flexibilité que l'on définit dans le cadre de l'approche par scénarios porte sur une solution, c'est-à-dire sur un ensemble de décisions. Dans la méthode ORABAID, on travaille sur un ordonnancement de groupes, qui caractérise un ensemble de solutions. Cela peut sembler un avantage de la méthode ORABAID, mais il faut rappeler que la solution calculée

hors ligne dans le cadre de l'approche par scénarios est dotée de robustesse, c'est-à-dire que l'on peut garantir ses performances sur un ensemble d'instances. Autrement dit, cette solution est calculée pour ne pas avoir à être remise en question en cas de perturbation « ordinaire ». Dans le cadre de la méthode ORABAID, la robustesse des solutions découlant de l'ordonnancement de groupes n'est pas connue, puisque aucune hypothèse n'est faite sur la nature et l'amplitude des perturbations. D'autre part, plus un ordonnancement de groupes est flexible, plus il est difficile de prévoir quand et sur quelle ressource une opération donnée sera exécutée, alors que ces décisions sont prises hors ligne dans le cas de l'approche par scénarios. Le tableau ci-dessous résume les avantages et les inconvénients de la méthode ORABAID et de l'approche par scénarios.

Méthode	Objet de la flexibilité	Avantages	Inconvénients
ORABAID	Séquence d'opérations permutables (ensemble de solutions)	Aucune hypothèse sur les perturbations de l'atelier n'est nécessaire	La solution appliquée n'est pas connue avant l'exécution (méthode temps réel)
Approche par scénarios	Solution	La solution initiale est entièrement connue hors-ligne. Changements de solution réservés aux perturbations « rares »	Détermination des instances « courantes », calcul d'une solution robuste sur ces instances

Tableau 5 : Comparaison de la méthode ORABAID et de l'approche par scénarios

Le Tableau 5 montre que les deux méthodes ne sont pas adaptées aux mêmes problèmes. Lorsque les incertitudes ou les aléas sont très importants, une méthode comme ORABAID convient parfaitement. Lorsqu'on dispose d'une certaine connaissance concernant les aléas, on peut réserver l'aspect temps réel de l'approche prédictive aux incertitudes « rares », qui ne permettent plus à la solution robuste de garantir des performances souhaitées. L'avantage principal de cette approche est que la solution robuste est entièrement connue avant son application. L'organisation et la planification des ressources de l'atelier s'en trouvent facilitées, bien que cette solution puisse être remise en cause, en cas de perturbation importante.

2.4.3 Définition du coût de flexibilité

Dans le cadre de l'approche par scénarios, on définit le coût de flexibilité d'une solution comme un coût de transformation portant sur les décisions qui constituent cette solution. Le coût dont il est question peut être de nature financière, énergétique, temporelle ou autre. Le coût de flexibilité mesure donc ce qu'il faut payer pour transformer une solution connue en une autre solution elle aussi connue. Ces différentes solutions peuvent être calculées hors ligne. Dans ce cas, le rôle du MEP (voir Figure 11) se limite à décider de l'opportunité de remplacer la solution courante par une solution calculée hors ligne, en tenant compte du coût de ce remplacement.

Une solution est considérée comme un ensemble de décisions. On peut ainsi comparer deux solutions à travers les décisions qu'elles comportent, et considérer la distance entre deux solutions comme une mesure du coût de flexibilité. La distance entre deux solutions est d'autant plus grande que ces solutions sont différentes. Pour un problème à une machine, on pourrait calculer cette distance entre deux solutions en comptant le nombre de permutations de tâches qu'il faut effectuer pour passer d'une solution à l'autre. On préférera pondérer ces permutations pour tenir compte du fait que toutes les modifications de la séquence n'ont pas le même coût : les modifications en ligne de l'affectation des tâches aux machines ne perturbent pas forcément l'organisation de l'atelier de la même manière ; les temps de réglage ou de préparation des ressources ne sont pas nécessairement équivalents.

Le coût de flexibilité, noté $f_{p,q}$ est défini par :

$$f_{p,q} = \sum_{\substack{k \notin s_p \\ k \in s_q}} a_k + \sum_{\substack{k \in s_q \\ k \notin s_p}} b_k$$

Les indices p et q font référence à une solution courante à transformer (s_p) et à une solution cible (s_q) vers laquelle on veut évoluer. Le coefficient a_k désigne un coût d'ajout en ligne, et b_k un coût d'annulation en ligne. L'indice k est relatif à la décision k qui constitue une composante d'une solution.

Le coefficient a_k est un coût (financier, temporel ou énergétique) associé à chaque décision k qui n'existe pas dans la solution courante s_1 , et qui doit être ajoutée pour transformer la solution s_1 en s_2 , en ligne. Ce coefficient peut représenter par exemple le coût d'ajout d'une machine supplémentaire en ligne.

Le coefficient b_k est un coût de même nature que a_k , qui représente le coût d'une annulation en ligne d'une décision qui a été prise hors ligne pour la solution s_p , mais qui n'existe pas dans la solution d'arrivée s_q . Ce coefficient peut représenter les pénalités consécutives à l'annulation tardive (car en ligne) de la réservation d'une machine ou d'un service.

2.4.4 Application

On reprend l'exemple des machines à affecter à différents produits dans le cas où la commande effective est inconnue. Ce problème a été présenté dans la première partie de ce mémoire, et a été complété par une étude de robustesse au paragraphe 2.3.3. Afin d'illustrer le coût de flexibilité utilisée pour évaluer la flexibilité d'une solution, on considère que les coûts d'ajout et de suppression d'une machine en ligne sont donnés dans le tableau suivant.

	Produit 1	Produit 2	Produit 3
Coût d'ajout d'une machine	200	450	60
Coût de suppression d'une machine	40	200	30

Tableau 6 : Coûts d'ajout/suppression de machine en ligne

Le coût de flexibilité nécessaire à la transformation de la solution $s_1 = \{3, 3, 3\}$ en la solution $s_2 = \{1, 3, 5\}$ en ligne s'exprime de la manière suivante :

$$f_{1,2} = (2 \times 60) + (2 \times 40) = 200$$

Le premier terme est un coût d'ajout : il faut ajouter 2 machines pour les produits de type 3 à la solution s_1 car la solution s_2 compte 5 machines pour les produits de type 3. Le deuxième terme est un coût d'annulation : la solution s_2 ne compte qu'une seule machine pour les produits de type 1 alors que s_1 en compte 3. Le coût de flexibilité pour passer en ligne de s_1 à s_2 est donc de 200. Le Tableau 7 ci-dessous regroupe l'ensemble des coûts de flexibilité que l'on peut obtenir avec les 5 solutions considérées. Les solutions « de départ » sont en ligne, et les solutions d'arrivée sont en colonne.

→	à s_1	à s_2	à s_3	à s_4	à s_5
De s_1	-	200	1260	660	400
De s_2	460	-	1720	1030	860
De s_3	1180	1380	-	300	690
De s_4	470	1080	690	-	480
De s_5	490	690	860	260	-

Tableau 7 : Coûts de flexibilité de chaque solution

Le coût de flexibilité d'une solution vers elle-même n'est pas évalué. Ce coût n'est pas nul : il n'a aucun sens. Il s'agit en effet d'un coût de changement en ligne, et un tel « changement » n'a pas de sens. On remarque également sur le Tableau 7 que le coût de flexibilité n'est pas symétrique : transformer s_1 en s_2 coûte 200, alors que transformer s_2 en s_1 coûte 460. Ceci interdit de considérer le coût de flexibilité comme une véritable distance entre solutions.

2.4.5 Flexibilité et garantie de performance

Le coût de flexibilité peut être vu comme une mesure de performance locale d'une solution. En effet, à chaque solution s d'un ensemble de S de solutions, est associée un coût de transformation vers chacune des autres solutions de S . Pour une solution donnée, on peut évaluer un critère de performance globale pour la flexibilité. Le pire cas W et la variabilité V sont définis, mais ce n'est pas le cas de la sensibilité parce que la notion d'optimalité du coût de flexibilité n'est pas définie. Sans écrire formellement la définition des critères de performance globale, le pire cas pour la solution s est ici le plus grand coût de flexibilité vers les solutions de S , et la variabilité est la différence entre le plus grand coût de flexibilité et le plus petit. On définit la flexibilité d'une solution de manière analogue à la robustesse : on dira qu'une solution s est flexible sur un ensemble S de solutions, si la performance globale est garantie sur cet ensemble.

Dans la définition qui suit, β_k désigne un critère de performance globale (le pire cas ou la variabilité, mais pas la sensibilité). Le réel g_k est un niveau de performance globale désiré associé au critère β_k . On dira qu'une solution s est flexible sur S si elle garantit une performance globale de g_k .

Définition 2: Etant donné le coût de flexibilité f associé à un jeu de coefficients a_k et b_k , le critère de performance global β_k et un niveau de performance globale à garantir noté g_k , on dit que la solution s est *flexible* sur l'ensemble de solutions S si :

$$\beta_k(s, f, S) \leq g_k \quad (2)$$

Le signe inférieur ou égal est utilisé dans la définition car les deux critères de performance globale sont à minimiser. La définition de la flexibilité est donc bien équivalente à une garantie de performance de la solution s , étant donné un ensemble de solutions S .

Notons que la notion de flexibilité, à l'instar de la robustesse, n'est pas une propriété intrinsèque à une solution. La formule (2) montre clairement que la flexibilité d'une solution s donnée dépend du critère de performance globale choisi β_k , du niveau de performance globale g_k exigé par l'utilisateur, de l'ensemble de solution S considéré et enfin du critère de performance locale f .

2.4.5.1 Application

Pour chacune des cinq solutions s_1 , s_2 , s_3 , s_4 , et s_5 dont l'expression est rappelée ci-après, on évalue le pire cas et la variabilité du coût de flexibilité. Les résultats sont regroupés dans le Tableau 8 :

Solution	Pire cas W	Variabilité V
$s_1 = \{3,3,3\}$	1260	1060
$s_2 = \{1,3,5\}$	1720	1260
$s_3 = \{7,1,1\}$	1380	1080
$s_4 = \{4,1,4\}$	1080	610
$s_5 = \{4,2,3\}$	860	600

Tableau 8 : Evaluation des critères de performances globales pour la flexibilité

A partir de ces critères et d'un niveau de performance désiré associé, il est possible de déterminer quelles sont les éventuelles solutions flexibles. Par exemple, si l'on choisit le pire cas et un niveau garanti de 1000, seule la solution s_5 est flexible. Il faut cependant insister sur le fait que, comme pour la robustesse, la définition de la flexibilité n'a pas pour objet de désigner la solution « la plus flexible » : un ensemble de solutions peut être flexible. Cet aspect est encore plus important pour la flexibilité que pour la robustesse, dans la mesure où c'est bien un ensemble de solutions qui doit être établi (et non une solution unique) dans le cas de l'ordonnancement réactif pour faire face aux aléas.

2.4.5.2 Les limites de l'analogie avec la robustesse

Dans cette thèse, on a défini la flexibilité par analogie avec la robustesse. Il s'agit d'une extension de la notion de « garantie de performance » au cas prédictif. Le mot performance fait référence à la fonction objectif dans le cas de la robustesse, et au coût de flexibilité dans le cas de la flexibilité. Cependant, le coût de flexibilité n'est pas une fonction objectif dans la mesure où $f_{p,q}$ ne permet pas d'évaluer la performance d'un ordonnancement. La robustesse apporte des garanties relativement à un ensemble d'instances I_A alors que la flexibilité porte sur un ensemble de solutions S . Les solutions que l'on choisit d'intégrer à cet ensemble sont celles qui seront utilisées en ligne pour être adaptées au contexte de l'atelier et aux exigences de performance. Dans le cas de l'exemple, toutes les solutions optimales associées aux scénarios de I_A sont incluses dans S . Cela signifie que l'on peut par exemple choisir d'appliquer la solution optimale à toute instance de I_A . Rappelons encore une fois que comme pour la robustesse, aucune méthode de synthèse de solution flexible n'est proposée dans le cas général. Les mesures proposées ici permettent d'évaluer la flexibilité de solutions préexistantes.

2.5 Choix d'une solution

Dans le cas d'un problème d'ordonnancement prédictif (aucun changement de solution n'est possible), le choix d'une solution ne tient pas compte des aspects de flexibilité. On peut proposer n'importe quelle solution robuste car les garanties de performance sont remplies. Le choix effectif d'une solution peut être fait en fonction d'autres critères (la facilité de mise en œuvre par exemple).

Dans le cas d'un problème d'ordonnancement réactif, il peut être intéressant de prendre en compte la robustesse et la flexibilité dans le choix d'une solution (ou de plusieurs solutions). Idéalement, la solution choisie doit être robuste et flexible. Ainsi, elle assure des performances acceptables pour les instances de I_A , et elle peut être changée « facilement » en cas de perturbation importante. Si aucune solution à la fois robuste et flexible ne peut être trouvée, le choix relève de la décision multicritère. Un compromis entre robustesse et flexibilité devra être trouvé.

2.6 Analogie avec la robustesse des systèmes continus

Le terme de robustesse n'est pas nouveau dans le domaine de l'automatique. Dans ce paragraphe, on se propose de mettre en évidence les nombreuses analogies formelles qui existent entre la définition de la robustesse pour une solution au problème d'optimisation en contexte incertain, et la notion de robustesse issue de l'automatique des systèmes continus.

Le premier point commun aux deux domaines est l'indispensable prise en compte du contexte incertain. Le paragraphe précédent illustre les conséquences de l'application d'une solution mal adaptée au changement de contexte. En automatique continue, une solution peut être une loi de commande, matérialisée par un contrôleur (ou régulateur). La loi de commande doit permettre le pilotage d'un procédé réel conformément à un cahier des charges. Dans ce cas, l'incertain se manifeste par une connaissance partielle du procédé à commander, par des aspects non linéaires qu'il n'est pas toujours possible de modéliser, ou encore par l'évolution des caractéristiques du procédé à commander (vieillesse).

L'admissibilité de l'ordonnancement produit par une solution peut être vue comme l'équivalent de la stabilité asymptotique du système en automatique continue. Le terme « système » désigne ici le procédé à commander et son contrôleur en boucle fermée. En ordonnancement, si l'admissibilité de la solution est une condition *sine qua non* à l'instar de la stabilité en automatique continue, il existe d'autres critères de performance exprimés par la fonction objectif (par exemple la minimisation du C_{max} , du retard maximum). Il en va de même en automatique continue : une loi de commande doit assurer la stabilité et satisfaire le cahier des charges. Parmi les critères de performance exprimés dans ce cahier des charges, on peut trouver par exemple la rapidité du système, sa précision, le dépassement maximal de la sortie en réponse à une entrée particulière.

En ordonnancement, on a présenté trois critères de performance globale, qui sont des mesures possibles de la robustesse d'une solution. En automatique, il existe également plusieurs critères de robustesse dont la signification est brièvement rappelée ci-après :

- *la marge de gain* représente la variation maximale que peut subir le gain de la boucle ouverte suite à des perturbations, tout en garantissant la stabilité du système en boucle fermée.
- *la marge de phase* est l'amplitude maximale des perturbations en terme de phase qui ne compromettent pas la garantie de stabilité du système en boucle fermée.
- *la marge de module* est la plus petite distance du point critique au tracé du transfert de boucle dans le plan de Nyquist.

Enfin, la définition de la robustesse qui a été proposée dans ce mémoire relie directement l'appartenance d'une solution à la classe des solutions robustes au niveau de la performance globale attendu par l'utilisateur. On observe la même démarche en automatique des systèmes continus : une loi de commande est dite robuste si la marge de phase du système est supérieure à 45 degrés, ou si sa marge de phase est supérieure à 6 dB. 45 degrés et 6 dB sont les niveaux de robustesse généralement attendus par les automaticiens, mais ils peuvent changer en fonction du procédé à commander, du niveau de sécurité exigé etc. Le Tableau 9 rassemble les principaux éléments de l'analogie entre automatique et ordonnancement pour la robustesse.

Vocabulaire de l'ordonnancement	Vocabulaire de l'automatique continue
solution	loi de commande
instance	état du procédé
ordonnancement	sortie du système
admissibilité de la solution	stabilité du système
fonction objectif	Critère de performance (précision, rapidité)
critères de performance globale	marges de robustesse
$\lambda_k(s, I, \sigma) \leq c_k$	marge de gain > 6dB

Tableau 9 : Analogie conceptuelle automatique / ordonnancement

La robustesse est donc synonyme de garantie de performance en automatique continue comme en ordonnancement, et cette notion n'est pas une caractéristique intrinsèque de la solution (ou de la loi de commande en automatique). La robustesse dépend des critères que l'utilisateur juge pertinents. Cette constatation peut être étendue à la performance des ordonnancements mesurée par la fonction objectif : il n'y a pas d'ordonnancement performant dans l'absolu, c'est la fonction objectif choisie par l'utilisateur qui permet d'en juger.

2.7 Conclusion

Dans ce chapitre, les notions de robustesse et de flexibilité ont été définies formellement. Lorsque l'incertain est modélisé avec l'approche par scénarios, la robustesse et la flexibilité sont synonymes de garantie de performance, où le mot performance est pris au sens large : il ne s'agit pas seulement de la performance locale. Les garanties de performance ainsi établies permettent au décideur de savoir exactement ce qu'il est en droit d'attendre d'une solution donnée, d'un point de vue « statique » avec la robustesse ainsi que d'un point de vue « dynamique » avec la flexibilité lorsque le changement de solution est autorisé. Aucun outil de génération de solution robuste ou flexible n'a été présenté dans ce cadre très général. La partie suivante est consacrée à un problème industriel pour lequel on cherche à augmenter la robustesse de solutions préexistantes.

PARTIE II

CONFIGURATION D'UN PARC DE

MACHINES PARALLELES

MULTIFONCTIONS EN CONTEXTE

INCERTAIN

Introduction

Depuis quelques années, l'étude des problèmes d'ordonnancement en contexte incertain a pris beaucoup d'importance. De nombreuses études dans des domaines très divers ont enrichi la littérature. [Chetouane *et al.*, 1997] traitent de la robustesse dans le cadre des ateliers flexibles de productions. Dans [Guinand *et al.*, 1999], les auteurs s'intéressent aux perturbations qui surviennent dans le cadre du calcul distribué.

Cette partie est consacrée à la configuration d'un parc de machines parallèles multifonctions en contexte incertain. Le mot configuration désigne une solution destinée à garantir un niveau de performance en présence de perturbations sur la commande que traite cet atelier. L'ensemble des commandes possibles consécutif aux perturbations constitue un ensemble de scénarios. Signalons également que cette étude permet de démontrer que l'approche par scénarios est tout à fait adaptée à des ensembles continus de scénarios non dénombrables.

Les machines dont il est question sont des machines de photolithographie utiles à la fabrication de circuits intégrés. Les origines et les manifestations du contexte incertain de la production seront détaillées dans le chapitre III. Dans cette étude, on utilisera des modèles relativement simples où les contraintes technologiques propres à la photolithographie ne sont pas toujours explicitement prises en compte. On trouvera dans [Mencerey, 2002] une étude où les contraintes spécifiques à cet atelier sont prises en compte pour élaborer une configuration admissible en minimisant un critère de coût, mais sans considérer la robustesse. Une modification de la configuration reste une décision importante et coûteuse qui immobilise la machine concernée. C'est la raison pour laquelle les industriels ne souhaitent pas configurer leur parc de machines *ex nihilo* : une « re-configuration », c'est à dire un ensemble de modifications à apporter à la configuration courante, est jugé préférable. La méthode proposée dans cette thèse consiste à identifier les modifications susceptibles d'augmenter la robustesse de la configuration courante. Le problème purement académique consistant à déterminer la « meilleure » modification à apporter pour maximiser la robustesse a bien sûr été abordé, mais les contraintes complexes de l'atelier peuvent interdire l'application de cette « meilleure » modification. Cependant, l'approche choisie permet de caractériser les modifications pertinentes, de sorte que la méthode se prête tout à fait à une utilisation de type aide à la décision. Ainsi, si la solution conduisant à la plus forte augmentation de la robustesse n'est pas applicable, il est possible de proposer d'autres modifications pertinentes. Enfin, la troisième et dernière utilisation que permet cette méthode est la suivante : un utilisateur expérimenté peut proposer une modification et évaluer immédiatement son impact sur la robustesse du parc. L'évaluation rationnelle de la robustesse de la configuration courante, ainsi que ce qu'on gagne à procéder à une modification donnée est une information précieuse apportée aux responsables de l'atelier de photolithographie. D'autre part, la souplesse que permet cette méthode est un avantage supplémentaire pour son application dans le monde industriel.

Enfin, rappelons que de par sa généralité, cette méthode n'est pas réservée aux ateliers de photolithographie, mais peut aisément être utilisée dans le cadre plus général des machines parallèles multifonctions.

Les trois chapitres de cette partie ne sont pas indépendants. Le chapitre III pose les bases de l'étude : la modélisation choisie, le critère de performance retenu et les trois hypothèses formulées concernant les perturbations qui agissent sur la commande. Le problème de la configuration du parc sous la première de ces trois hypothèses est très simple, c'est la raison pour laquelle cette hypothèse est traitée dans le chapitre III, à la suite des définitions et du modèle.

Le chapitre IV est consacré à la seconde hypothèse. La notion de rayon de stabilité introduite par Sotskov se révèle très utile pour résoudre le problème. L'étude de deux distributions de vitesses sur le parc machines est conduite de front dans ce chapitre, ce qui permet d'illustrer les points communs et les différences qui résultent de ces distributions de vitesses.

Dans le chapitre V, le théorème qui est à la base de la méthode proposée pour mesurer la robustesse d'une configuration est démontré dans le cas le plus général des distributions de vitesses sur le parc de machines. L'étude de la troisième hypothèse concernant l'incertain (la plus générale) bénéficie des outils et des résultats introduits dans les deux chapitres précédents. Le chapitre IV est également l'occasion de comparer les outils et les résultats obtenus dans le cadre de chacune des trois hypothèses.

Chapitre III : Configuration d'un parc de machines parallèles multifonctions en contexte incertain

3.1 Contexte général de l'étude

Ce chapitre est consacré à l'application des outils développés pour l'ordonnancement en contexte incertain à un problème industriel. Il s'agit de déterminer la configuration d'un atelier de photolithographie en contexte incertain, afin de garantir un taux d'occupation des machines égal à 100 % (c'est le critère de performance retenu dans ce problème). Cet atelier occupe une place centrale dans une unité de production de circuits intégrés. Le renouvellement très fréquent des technologies rend rapidement obsolètes les machines qui composent cet atelier, et d'autre part, le coût d'achat de ces machines est extrêmement élevé. Il en résulte que l'atelier de photolithographie a souvent une capacité de production inférieure à celle des autres ateliers de l'unité de fabrication. Il constitue fréquemment un goulet d'étranglement dans le parcours des produits, ce qui est à l'origine des vagues de WIP (Work In Progress) qui désorganisent la production en se propageant à l'ensemble du système de production. En raison du coût d'achat très élevé des machines, il convient de configurer cet atelier de manière à assurer que toutes les machines qui le composent auront un taux d'occupation maximal. De plus, il est indispensable de conférer des propriétés de robustesse à cette configuration, afin que les incertitudes qui pèsent sur la quantité et la répartition des produits présents à l'entrée de l'atelier ne provoquent pas une baisse du taux d'occupation des machines.

3.1.1 Les origines du contexte incertain

Le volume et la répartition des produits à traiter dans cet atelier sont incertains pour deux raisons principales. Premièrement, si le contexte de toute activité de production industrielle est incertain, celui des circuits intégrés l'est particulièrement. Le lancement de nouveaux produits est très fréquent car l'évolution des technologies et l'innovation sont particulièrement rapides dans ce secteur. Or la mise en production d'un produit nécessitant une nouvelle technologie engendre presque toujours un taux de rebut particulièrement important au début. Ceci explique les prix très élevés des nouveaux produits basés sur les semi conducteurs au début de leur cycle de vie (microprocesseurs, modules de mémoire), ainsi que leurs baisses parfois spectaculaires lorsque leurs technologies de fabrication sont bien maîtrisées. Si les rebuts coûtent cher à l'entreprise, ils désorganisent également la production en perturbant les prévisions établies pour chaque atelier. D'autres ateliers, comme l'atelier d'implantation, sont constitués de machines dont les performances dépendent directement des réglages. La capacité de production de ces ateliers est donc difficile à prévoir, car elle dépend de l'aptitude

des opérateurs à effectuer de bons réglages sur les machines dont ils ont la charge. La deuxième source d'incertitude provient du fait que l'atelier de photolithographie étudié se trouve sur un site industriel dédié à la production, mais aussi aux activités de recherche et de développement. Ceci se traduit par une grande variété de produits traversant les ateliers, et par une diversité accrue des routages de ces mêmes produits. Le parc de machines sur ce site est également caractérisé par une inhomogénéité résultant de ces activités de R et D.

3.1.2 Les objectifs de l'étude

Le premier objectif de cette partie est d'établir des outils permettant de mesurer la robustesse d'une configuration de l'atelier de photolithographie. La robustesse d'une configuration est son aptitude à garantir la propriété d'équilibre de la charge des machines en dépit des perturbations qui affectent le volume et/ou la répartition des produits de la commande. Le second objectif est de déterminer les modifications à apporter à une configuration pour maximiser sa robustesse, tout en minimisant un critère de coût qui sera présenté en détails. L'étude sera conduite successivement dans trois cas, correspondant à trois hypothèses sur la nature des incertitudes qui affectent la commande de l'atelier de photolithographie.

3.2 Présentation et modélisation des contraintes de l'atelier de photolithographie

3.2.1 Modélisation du problème

3.2.1.1 Vitesse et configuration du parc de machines

Un atelier de photolithographie est composé de m machines parallèles qui ne sont pas entièrement multifonctions : certaines machines ne peuvent pas traiter certains types produits pour des raisons de limitation de ressources. En effet, chaque type de produits nécessite la présence d'une ressource (une résine en l'occurrence), et le nombre de ressources que l'on peut installer sur une machine est limité. Cette limitation est à l'origine des contraintes de qualification des machines : une machine est qualifiée pour un type de produit si elle dispose de la ressource nécessaire à son traitement, elle n'est pas qualifiée sinon. Les produits d'un même type sont identiques, et on note n le nombre de types de produits. L'ensemble des qualifications constitue la configuration du parc de machines, et cette configuration est le résultat d'un choix visant à assurer un taux d'occupation des machines de 100 %. Cependant, en plus des limitations de ressources, ce choix doit assurer le respect des contraintes technologiques : une machine trop ancienne ne peut traiter un produit récent, ou certaines ressources incompatibles ne peuvent être installées sur la même machine. La modélisation de ce problème relève de ce que la littérature anglo-saxonne appelle le Multi-Purpose Machines. Un état de l'art de la complexité des problèmes de Multi-Purpose Machine est proposé dans [Brucker *et al.*, 1997].

Dans le cas général, la vitesse d'une machine j pour un type de produit i est quelconque, et notée $v(i,j)$. On parle dans la littérature de machines non reliées. Le coefficient $v(i,j)$

représente la quantité de produits de type i que la machine j peut traiter par unité de temps si elle est qualifiée pour ce type de produit. On a donc nécessairement $v(i,j) \geq 0$. Si $v(i,j) = 0$, cela signifie que la machine j ne peut en aucun cas être utilisée pour traiter un produit de type i . Il s'agit alors d'une contrainte technologique, ce qui est différent d'une non qualification parce que la qualification d'une machine pour un produit relève d'un choix : lorsque $v(i,j) = 0$, la question de la qualification de la machine j pour le type de produit i ne se pose pas. Il ne sera donc possible de qualifier une machine j pour un produit de type i que si $v(i,j) > 0$.

Les m machines de l'atelier de photolithographie sont destinées à traiter des produits répartis dans n types de produits. On notera J l'ensemble des entiers de $[1,m]$. J désigne aussi l'ensemble des machines du parc. De même, on notera I l'ensemble des entiers de $[1,n]$. I désigne aussi l'ensemble des types de produits à traiter par le parc de machines. La matrice de qualification, notée Q est formée de n lignes et de m colonnes, et représente l'ensemble des qualifications, c'est-à-dire la configuration du parc de machines. Le coefficient $Q(i,j)$ vaut un si la machine j est qualifiée pour les produits de type i , il vaut zéro sinon. Une colonne de la matrice de qualification représente une machine, et une ligne représente un type de produits. La matrice de vitesse notée v , dont les coefficients $v(i,j)$ ont été définis plus haut a les mêmes dimensions que Q .

On note Q_v la matrice définie par : $Q_v(i,j) = Q(i,j) \times v(i,j) \quad (\forall i \in I)(\forall j \in J)$. Cette matrice représente la vitesse des machines seulement sur les produits pour lesquelles elles sont qualifiées.

Une matrice de qualification est admissible si et seulement si elle n'admet aucune ligne pleine de zéros, ni aucune colonne pleine de zéros. Si la ligne i est pleine de zéros alors aucune machine du parc n'est qualifiée pour les produits de type i . Il est impossible de traiter les produits de ce type. Si la colonne j est pleine de zéros, alors la machine j n'est qualifiée pour aucun produit. Il est alors inutile de la considérer. Dans cette thèse, on ne considérera que des matrices de qualification admissibles. On fera également l'hypothèse que les contraintes technologiques permettent toujours d'obtenir de telles matrices de qualification.

3.2.1.2 Commande

Les produits destinés à recevoir le même traitement lors de leur passage dans l'atelier de photolithographie constituent un type de produits. La commande de l'atelier est un vecteur colonne de n éléments noté N . Le coefficient $N(i)$ représente la quantité de produits de type i présente à l'entrée de l'atelier. Il s'agira également de la part des produits de type i dans la commande N (par exemple $N(i) = 33.3 \%$). Le coefficient $N(i)$ sera donc un nombre réel. Une commande est admissible si et seulement si aucun élément de N n'est strictement négatif. On ne considérera que les commandes admissibles.

3.2.2 Calcul d'un plan de production optimal

Le critère de performance retenu est la maximisation du taux d'occupation des machines. Ce critère est évalué en deux temps. On résout d'abord le problème de la minimisation du C_{max} , étudié par Lawler et Labetoulle, puis on détermine une solution à usure minimale des machines, car cette qualité n'est pas garantie pour la solution issue du programme linéaire de

Lawler et Labetoulle. Les deux paragraphes suivants présentent les programmes linéaires permettant de résoudre ces problèmes.

3.2.2.1 Calcul d'un plan de production minimisant la durée d'exécution

Etant donnés une matrice de vitesse v , une matrice de qualification Q et une commande N , on cherche à répartir les produits de la commande N sur le parc des m machines, de manière à minimiser la plus grande date de fin sur l'ensemble des machines. Cette répartition doit respecter les contraintes de qualification, et tenir compte des vitesses des machines. La préemption et le splitting sont autorisés, ce qui signifie que deux unités d'un même type de produit i peuvent être traités sur deux machines différentes (qualifiées pour i), et que ces traitements peuvent être simultanés. Une solution optimale à ce problème est obtenue en temps polynomial à l'aide d'un programme linéaire en nombre réels, dérivé du programme linéaire de [Lawler *et al.*, 1978]. Un plan de production peut s'écrire sous la forme d'une matrice de répartition temporelle R_T ayant les mêmes dimensions que Q , telle que $R_T(i,j)$ représente le temps passé par la machine j à traiter les produits de type i qui lui ont été affectés. Tout coefficient $R_T(i,j)$ doit être positif ou nul. Le respect des contraintes de qualification implique $R_T(i,j) = 0$ lorsque $Q(i,j) = 0$. On note (PL 1) le programme linéaire utilisé pour déterminer un plan de production optimal au sens du critère C_{max} :

$$\begin{cases} \sum_{j=1}^m R_T(i,j) \times v(i,j) \times Q(i,j) = N(i) & (\forall i \in I) \\ \sum_{i=1}^n R_T(i,j) \times Q(i,j) \leq C_{max} & (\forall j \in J) \quad (\text{PL 1}) \\ R_T(i,j) \geq 0 & (\forall i \in I)(\forall j \in J) \\ \min(C_{max}) \end{cases}$$

a) Remarques

La première ligne du programme linéaire indique que la demande exprimée par N doit être satisfaite. Remarquons que l'on pourrait condenser l'écriture en remplaçant $v(i,j) \times Q(i,j)$ par $Q_v(i,j)$. La seconde contrainte est une majoration du temps de travail de toutes les machines, et c'est ce temps (C_{max}) que l'on veut minimiser. La troisième ligne exprime les contraintes de positivité sur les coefficients de R_T . Une solution optimale est obtenue à l'aide de l'algorithme du Simplexe en moins d'une seconde sur un PC Pentium IV à 2 GHz doté de 512 Mo de RAM pour des instances de taille industrielle (36 types de produits, 13 machines).

b) Exemple

Soit la matrice de vitesse v , la matrice de qualification Q et la commande N définies ci-après.

$$v = \begin{bmatrix} 2 & 0 & 4 \\ 1 & 2 & 2 \\ 4 & 1 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad N = \begin{bmatrix} 8 \\ 3 \\ 5 \end{bmatrix} \quad Q_v = \begin{bmatrix} 0 & 0 & 4 \\ 1 & 2 & 2 \\ 4 & 1 & 1 \end{bmatrix}$$

Ici, la machine 2 ne peut en aucun cas traiter de produit de type 1. Il s'agit d'une contrainte technologique car $v(1,2) = 0$. En revanche, $v(1,1) = 2$ et $Q(1,1) = 0$ indiquent que la machine 1 est capable de traiter des produits de type 1, mais que la configuration courante n'exploite pas cette possibilité (par exemple parce qu'on ne peut équiper qu'une seule machine pour traiter ces produits). La matrice Q_v peut être déduite de Q et de v .

L'application de (PL 1) à ces données permet par exemple d'obtenir la solution suivante (qui n'est pas la seule solution optimale) :

$$R_T = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \\ C = \begin{bmatrix} 2 & 2 & 2 \end{bmatrix}$$

Sur cette répartition temporelle, on constate que chaque machine a une charge de 2 unités de temps. La charge temporelle de la machine j s'obtient en faisant la somme des éléments de la colonne j . La matrice de répartition matérielle R_M associée à R_T , s'obtient par la formule suivante :

$$R_M(i, j) = R_T(i, j) \times v(i, j) \quad (\forall i \in I)(\forall j \in J) \text{ d'où } R_M = \begin{bmatrix} 0 & 0 & 8 \\ 1 & 2 & 0 \\ 4 & 1 & 0 \end{bmatrix}$$

Connaissant R_M , on peut alors vérifier facilement que la demande exprimée par N est satisfaite. La somme de la ligne i est bien égale à $N(i)$.

c) Equilibre simple

Définition 3: On dit qu'un couple (Q_v, N) est simplement équilibré s'il existe une solution optimale à (PL 1) telle que toutes les machines ont la même charge temporelle, c'est-à-dire :

$$C_{\max} = \sum_{i=1}^n R_T(i, j) \quad (\forall j \in J)$$

Dans le cas de l'exemple, le couple (Q_v, N) est simplement équilibré parce que R_T , solution optimale de (PL 1) est telle que toutes les machines ont une charge égale à 2 unités de temps.

3.2.2.2 Calcul d'un plan de production minimisant l'usure des machines

Le programme linéaire (PL 1) garantit l'optimalité du temps d'exécution C_{max} de la commande N par le parc de machines. Cependant, rien n'indique que l'usure des machines est minimale. Ce critère est important car si l'usure des machines n'est pas minimale, c'est que la charge temporelle d'une ou de plusieurs machines peut diminuer sans augmenter la valeur de C_{max} . Le programme linéaire (PL 2) présenté ci-dessous permet de calculer une répartition temporelle R_T' connaissant la valeur numérique de C_{max} déterminée par (PL 1) :

$$\left\{ \begin{array}{ll} \sum_{j=1}^m R_T'(i, j) \times v(i, j) \times Q(i, j) = N(i) & (\forall i \in I) \\ \sum_{i=1}^n R_T'(i, j) \times Q(i, j) = C_{max} - z(j) & (\forall j \in J) \\ R_T'(i, j) \geq 0 & (\forall i \in I)(\forall j \in J) \\ z(j) \geq 0 & (\forall j \in J) \\ \max \left(\sum_{j=1}^m z(j) \right) & \end{array} \right. \quad (PL\ 2)$$

a) Remarques

La première ligne de (PL 2) est identique celle de (PL 1). Elle exprime la satisfaction de la demande exprimée par N . La seconde ligne est l'expression de la charge temporelle de la machine j . Cette charge vaut $C_{max} - z(j)$, où $z(j)$ représente l'écart à C_{max} . La quatrième ligne interdit à la charge d'une machine de dépasser la valeur numérique C_{max} calculé dans (PL 1) en imposant la positivité de $z(j)$. Enfin, le critère de maximisation porte sur la somme des $z(j)$, c'est-à-dire le temps total pendant lequel les machines dont la charge est strictement inférieure à C_{max} ne sont pas utilisées. Ainsi on est certain que dans la répartition temporelle R_T' obtenue par (PL 2), les machines qui peuvent ne pas travailler jusqu'à C_{max} sont utilisées le moins de temps possible, sans remettre en cause la valeur de C_{max} . La minimisation de l'usure des machines est un critère pertinent dans le contexte de la photolithographie, où l'on souhaite exploiter au mieux les machines. Comme (PL 1), ce programme linéaire est résolu en moins d'une seconde sur un PC Pentium IV avec l'algorithme du Simplexe.

b) Exemple

Reprenons l'exemple introduit dans le paragraphe précédent. Après l'exécution de (PL 1), on avait $C_{max} = 2$ unités de temps. La solution optimale à (PL 1), R_T , est rappelée pour comparaison avec R_T' solution optimale à (PL 2).

$$R_T = \begin{bmatrix} 0 & 0 & 2 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \quad R_T' = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 1.5 & 0 \\ 1.25 & 0 & 0 \end{bmatrix}$$

La valeur de C_{max} reste égale à 2 unités de temps pour les deux répartitions. Cependant, toutes les machines travaillent 2 unités de temps dans la solution R_T , alors que seule la machine 3 a cette charge dans R_T' . On peut en déduire que R_T n'est pas à usure minimum, puisqu'elle impose une charge de 2 unités de temps aux machines 1 et 2, alors que cette charge est ramenée à 1.25 et 1.5 unités de temps respectivement pour la répartition R_T' .

Plus important encore, la solution R_T laisse penser que le taux d'occupation des machines est de 100 % alors qu'il existe d'autres répartitions (dont R_T') assurant le même C_{max} et qui sollicitent moins les machines 1 et 2. Ceci est particulièrement important dans la mesure où le critère de performance qui a été retenu pour juger une configuration (une matrice Q) est son aptitude à assurer un taux d'occupation des machines de 100 %. Dans le cas de R_T , le taux d'utilisation est de 100 % parce que les machines 1 et 2 « attendent » que la machine 3 termine à $t = C_{max}$ en traitant des produits pour lesquels leur vitesse est faible. Il est clair que ce type de répartition n'est pas pertinent du point de vue industriel. C'est la raison pour laquelle on introduit la notion d'équilibre strict.

c) Equilibre strict

Définition 4: On dit qu'un couple (Q_v, N) est strictement équilibré s'il existe une solution optimale à (PL 2) telle que toutes les machines ont la même charge temporelle, c'est-à-dire :

$$C_{max} = \sum_{i=1}^n R_T'(i, j) \quad (\forall j \in J)$$

Dans le cas de l'exemple, le couple (Q_v, N) est simplement équilibré mais il n'est pas strictement équilibré, parce que R_T' , solution optimale de (PL 2) est telle que toutes les machines n'ont pas une charge égale à 2 unités de temps.

Par construction, l'équilibre strict implique l'équilibre simple, mais la réciproque est fausse en général.

d) Terminologie

Dans la suite de ce mémoire, on dira que la matrice de répartition R_T est une solution simplement équilibrée si R_T est une solution optimale de (PL 1), pour une matrice Q_v et une commande N données. Il s'agira d'une solution strictement équilibrée si R_T est aussi une solution optimale de (PL 2).

3.2.3 Examen de deux cas particuliers de distribution des vitesses

L'objet de ce paragraphe est de montrer que lorsque les machines sont uniformes ou identiques, l'équilibre simple et l'équilibre strict sont équivalents. Ceci permettra de se limiter à l'étude de l'équilibre simple dans ces deux cas particuliers sachant que l'équilibre strict est assuré.

3.2.3.1 Machines uniformes

Définition 5: Soient j une machine et i un type de produit. Si la matrice v est telle que pour tout coefficient $v(i,j)$ non nul on a $v(i,j) = V(j)$, alors on dit que les machines sont uniformes. Dans ce cas, quelle que soit la configuration de ce parc de machines, la matrice Q_v représente un parc de machines uniformes par définition de Q et de Q_v .

Propriété : Lorsque les machines sont uniformes, toute commande simplement équilibrée pour Q_v est strictement équilibrée.

Démonstration : La capacité de production d'une machine j est égale à $V(j)$ indépendamment des types de produits qu'elle traite. On définit le réel s strictement positif par :

$$s = \sum_{j \in J} V(j)$$

Le réel s représente la capacité de production du parc de machines. Celle-ci est indépendante des types de produits. On note N une commande simplement équilibrée pour Q_v . On note $C(j)$ la charge temporelle de la machine j . Pour une matrice de répartition quelconque R_T , on a :

$$C(j) = \sum_{i=1}^n R_T(i, j)$$

Comme N est simplement équilibrée pour Q_v et que la capacité de production de chaque machine est égale à $V(j)$, on peut exprimer le nombre de produits traités en fonction de la charge temporelle $C(j)$ et de la vitesse $V(j)$ de chaque machine :

$$\sum_{i \in I} N(i) = \sum_{j \in J} (C(j) \times V(j)) = m \times C_{\max} \times s$$

Supposons maintenant que N ne soit pas strictement équilibrée pour Q_v . Cela signifie qu'il existe au moins une machine j_0 telle que $C(j_0) < C_{\max}$. Par ailleurs, $C(j)$ doit être inférieur ou égal à C_{\max} pour tout j , d'où :

$$\left(\sum_{j \in J} C(j) \times V(j) \right) < \sum_{j \in J} (C_{\max} \times V(j)) = \sum_{i \in I} N(i)$$

Cela signifie que si la commande N n'est pas strictement équilibrée pour Q_v , la demande n'est pas exactement satisfaite, ce qui est absurde. On en déduit que l'équilibre simple est équivalent à l'équilibre strict dans le cas d'un parc de machines uniformes.

3.2.3.2 Machines identiques

Définition 6: Soient j une machine et i un type de produit. Si la matrice v est telle que pour tout coefficient $v(i,j)$ non nul on a $v(i,j) = V(i)$, alors on dit que les machines sont identiques. Dans ce cas, quelle que soit la configuration de ce parc de machines, la matrice Q_v représente un parc de machines identiques par définition de Q et de Q_v .

Propriété : lorsque les machines sont identiques, toute commande simplement équilibrée pour Q_v est strictement équilibrée.

Démonstration : On définit $T(i)$ le temps de traitement de tous les produits d'un type i donné par :

$$T(i) = \frac{N(i)}{V(i)}$$

L'expression de $T(i)$ est indépendante de la répartition des produits sur les machines. Supposons que la commande N soit simplement équilibrée pour Q_v , et écrivons la valeur de C_{\max} :

$$C_{\max} = \frac{\sum_{i \in I} \frac{N(i)}{V(i)}}{m}$$

Supposons maintenant que N ne soit pas strictement équilibrée pour Q_v . Cela signifie qu'il existe au moins un type de produits i_0 tel que $T(i_0) < \frac{N(i_0)}{V(i_0)}$. Or cette inégalité est absurde car elle est en contradiction avec la relation établie plus haut. On en déduit que l'équilibre simple est équivalent à l'équilibre strict dans le cas d'un parc de machines identiques.

3.3 Caractéristiques et modèles du contexte incertain

Les aléas qui surviennent dans les ateliers situés en amont de l'atelier de photolithographie peuvent avoir des impacts différents sur la commande de ce dernier. Si la panne d'une machine en amont peut avoir pour effet de diminuer le volume total de la commande N , il se peut que la répartition des produits dans les différents types de produits de N en soit également affectée. Dans les paragraphes suivants, on notera N_{ref} la commande prévisionnelle établie au début de chaque mois pour l'atelier de photolithographie, et N la commande réelle se trouvant à l'entrée de l'atelier. On s'intéressera à la conservation de la propriété d'équilibre (simple ou strict) du couple (Q_v, N) avec $N = N_{ref} + dN$, où le vecteur dN représente l'écart entre la commande réelle N et la commande prévisionnelle N_{ref} , dans les trois cas suivants :

1. le volume de la commande réelle est incertain, mais la répartition des produits est connue.
2. la répartition des produits de la commande réelle est incertaine, mais le temps de production (dans le cas des machines uniforme) ou le volume des commandes (dans le cas des machines identiques) est connu.
3. la répartition et le volume de la commande réelle sont incertains (cas général).

Chacun de ces cas correspond à des hypothèses différentes sur les perturbations (représentées par le vecteur dN) qui affectent la commande prévisionnelle N_{ref} . Notons que le vecteur dN peut avoir des composantes négatives : $dN(i) < 0$ indique que la quantité de produits de type i est inférieure à la quantité attendue exprimée par $N_{ref}(i)$. Cependant, dN ne peut pas être tel que $N(i) = N_{ref}(i) + dN(i) < 0$, car aucun élément de la commande réelle N ne peut être négatif, étant donné que N désigne la commande physiquement présente à l'entrée de l'atelier. Il en résulte que N est nécessairement une commande admissible. La Figure 12 illustre les 3 hypothèses concernant l'incertain, dans le cas où l'on a $n = 2$ types de produits. Les zones dessinées en gris sont les commandes N définies par $N = N_{ref} + dN$.

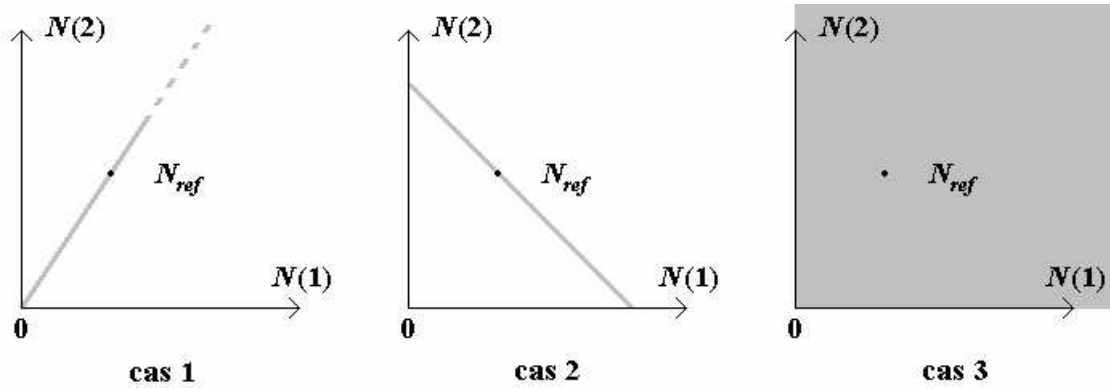


Figure 12 : Représentation des 3 hypothèses concernant les perturbations de la commande pour $n = 2$

Dans le premier cas, les commandes étudiées sont sur une demi droite passant par N_{ref} , et dont l'origine est le point $(0,0)$: ces commandes sont proportionnelles à N_{ref} car seul le volume est inconnu.

Dans le second cas, les commandes considérées constituent une portion d'hyperplan passant par N_{ref} . Comme $n = 2$ dans le cas de la Figure 12, l'ensemble des commandes est un segment de droite. Les commandes de cet ensemble ont le même volume que N_{ref} si les machines sont uniformes, ou le même temps de production que N_{ref} si les machines sont identiques.

Enfin, le troisième cas est le plus général puisque l'ensemble des commandes considérées est \mathcal{R}_+^n , c'est-à-dire l'ensemble des commandes admissibles. Il s'agit d'une portion de plan lorsque $n = 2$.

Le premier cas est étudié ci-après. Le deuxième cas fait l'objet du chapitre IV, et le chapitre V est consacré au troisième cas. Ces chapitres ne sont cependant pas indépendants : de nombreux outils sont présentés en détails lorsqu'ils apparaissent pour la première fois dans le chapitre IV et sont plus sommairement illustrés dans les chapitres suivants. Ceci s'applique en particulier à la méthode proposée pour conférer plus de robustesse à une configuration \mathcal{Q} : cette méthode étant identique dans les cas 2 et 3, n'est pas reprise dans le chapitre V.

3.4 Le volume de la commande réelle est incertain

Dans ce paragraphe, on fait l'hypothèse que la répartition des produits de la commande N dans chaque type de produits reste constante, mais que le volume total de N est inconnu. Plus formellement, cela se traduit par :

$$\frac{N(i)}{N_{ref}(i)} = a \quad (\forall i \in I)$$

Cela correspond à une perturbation de la capacité de production des ateliers amont n'affectant pas la part de la commande affectée à chaque type de produits. Autrement dit, la commande N s'écrit :

$$N = a \times N_{ref} \quad \text{où } a \text{ désigne un réel strictement positif}$$

Montrons que l'équilibre simple (strict) de la commande N_{ref} pour Q_v implique l'équilibre simple (strict) de la commande N définie par $N = a \times N_{ref}$ pour Q_v quel que soit le réel $a > 0$.

Soit N_{ref} simplement équilibrée pour Q_v . La matrice de répartition R_T , solution optimale de (PL 1) pour Q , v et N_{ref} assure la même charge temporelle notée C_{max} à toutes les machines. Soit a un réel tel que $a > 0$, alors multiplier chaque ligne de (PL 1) par a permet de déduire que la matrice de répartition R_T' définie par $R_T' = a \times R_T$ est une solution optimale de (PL 1) pour la commande N définie par $N = a \times N_{ref}$. La charge temporelle de toutes les machines avec la solution R_T' vaut alors $a \times C_{max}$. La commande N est donc simplement équilibrée pour Q_v .

En multipliant toutes les lignes de (PL 2), on montrerait de même que si N_{ref} est strictement équilibrée pour Q_v , alors c'est aussi le cas de la commande $N = a \times N_{ref}$ avec a strictement positif.

On peut également remarquer que si la commande N_{ref} n'est pas simplement (strictement) équilibrée pour Q_v , le couple (Q_v, N) avec $N = a \times N_{ref}$ n'est pas simplement (strictement) équilibré pour Q_v . En effet, s'il existait une valeur de $a > 0$ telle que N est équilibré pour Q_v , alors N_{ref} le serait aussi d'après les résultats démontrés précédemment, ce qui est absurde par hypothèse.

En résumé, on a montré que la propriété d'équilibre (simple ou strict) d'un couple (Q_v, N_{ref}) donné est conservée par multiplication de la commande.

On peut alors affirmer que si seul le volume de la commande réelle N est inconnu (la répartition des produits de N dans les différents types de produits étant identique à celle de N_{ref}), alors l'équilibre simple (respectivement strict) du couple (Q_v, N) est toujours garanti à condition que le couple (Q_v, N_{ref}) soit simplement (strictement) équilibré.

Si le couple (Q_v, N_{ref}) n'est pas équilibré (simple ou strictement), il conviendra d'appliquer la méthode d'augmentation du rayon de stabilité qui est présentée dans la partie suivante.

3.5 Conclusion

Ce chapitre a permis de situer le problème de la configuration d'un parc de machines parallèles multifonctions en contexte incertain. Le critère de performance, bien que facile à appréhender (le taux d'occupation des machines), ne s'exprime pas simplement : il nécessite

la résolution de deux programmes linéaires en nombres réels dans le cas général, pour être évalué. Le problème de la détermination des commandes strictement équilibrées pour une configuration donnée ne sera abordé que dans deux cas particuliers concernant la distribution des vitesses sur le parc de machines. Le chapitre suivant est consacré au cas où les perturbations conservent la charge de la commande (pour les machines uniforme), ou son temps de traitement brut (pour les machines identiques).

Chapitre IV : Le volume ou le temps de traitement de la commande est conservé

Dans ce chapitre, on fait l'hypothèse que les perturbations agissent sur la répartition des produits de la commande, mais que le volume de la commande reste égal au volume de la commande N_{ref} dans le cas des machines uniformes, ou que le temps de traitement de la commande reste égal au temps de traitement de N_{ref} dans le cas des machines identiques. On s'intéressera à ces deux cas particuliers concernant la matrice de vitesse v pour lesquels on a démontré que la matrice Q_v était nécessairement sans boucle, quel que soit Q admissible. On étudiera alors la propriété d'équilibre simple qui est équivalente à la propriété d'équilibre strict dans ces deux cas particuliers. Le cas général (v quelconque) n'est pas abordé pour des raisons techniques qui seront expliquées dans le paragraphe traitant du cas où la répartition et le volume de la commande sont inconnus. Les deux cas particuliers étudiés ci-après sont cependant suffisamment génériques pour représenter un parc de machines de photolithographie de manière satisfaisante.

4.1 Les machines sont uniformes

On rappelle que lorsque les machines sont uniformes, la matrice de vitesse v est telle que pour tout coefficient $v(i,j)$ non nul, on a $v(i,j) = V(j)$ pour tout i . Cela signifie que outre les contraintes technologique (qui imposent les coefficients nuls de v), une machine j a la même vitesse pour tous les types de produits qu'elle peut potentiellement traiter. Cette vitesse unique est notée $V(j)$. Par conséquent, une unité de produit de type i ne sera pas traitée à la même vitesse en fonction de la machine à laquelle elle est affectée.

La matrice de vitesse suivante entre dans le cadre de l'étude du présent paragraphe :

$$v = \begin{bmatrix} 1 & 2 & 0.5 & 1.2 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0.5 & 1.2 \end{bmatrix}$$
$$V = [1 \quad 2 \quad 0.5 \quad 1.2]$$

Le vecteur V représente la vitesse de chaque machine. On remarque deux contraintes technologiques sur v : la machine 1 ne peut jamais traiter de produit de type 3, et la machine 4

ne peut jamais traiter de produits de type 2. On remarquera également qu'aucune boucle ne peut être trouvée dans une telle matrice.

Dans ce paragraphe, on suppose que le vecteur dN est tel que le volume de la commande réelle N est toujours égal au volume de la commande prévisionnelle N_{ref} . Plus formellement, cela se traduit par l'égalité suivante :

$$\sum_{i=1}^n N(i) = \sum_{i=1}^n (N_{ref}(i) + dN(i)) = \sum_{i=1}^n N_{ref}(i)$$

On peut en déduire que $\sum_{i=1}^n dN(i) = 0$. Cela signifie bien que le volume prévisionnel n'est pas modifié. On va maintenant chercher à déterminer l'ensemble des commandes N simplement équilibrées dont la charge est égale à celle de N_{ref} , ce qui permettra de déduire la plus petite valeur de la norme de dN qui provoque la perte de la propriété d'équilibre. La valeur de cette norme constituera une mesure de robustesse de la configuration Q . Dans les paragraphes suivants, on parlera d'équilibre sans préciser s'il s'agit de l'équilibre simple ou strict car ces deux notions sont équivalentes moyennant l'absence de boucle dans v (et donc aussi dans Q_v). En pratique, on n'étudiera que l'équilibre simple.

4.1.1 Etude de l'espace des commandes

A priori, l'espace des commandes à étudier est R_+^n , c'est-à-dire l'ensemble des vecteurs de dimension n , dont aucun élément n'est strictement négatif. On a vu que la propriété d'équilibre n'était pas modifiée par multiplication de la commande. On peut donc étudier l'ensemble des commandes N dans un sous-ensemble de R_+^n défini par :

$$\sum_{i=1}^n N(i) = \text{constante}.$$

Définition 7: On note H_s l'ensemble des vecteurs N de dimension n défini par :

$$\begin{cases} N(i) \geq 0 & (\forall i \in I) \\ \|N\|_1 = \sum_{j=1}^m V(j) = s \end{cases}$$

Le réel s est la somme des vitesses des machines. Cette quantité est égale au nombre maximum de produits que le parc de machines peut traiter par unité de temps, tous types de produits confondus.

La notation $\|X\|_1$ désigne la norme 1 du vecteur X :

$$\|X\|_1 = \sum_i |X(i)|$$

Remarque : H_s est l'ensemble des commandes (admissibles) dont la norme est égale à s . On peut donc en déduire que toute commande équilibrée de H_s conduira à une valeur de C_{max} unitaire. Une commande non équilibrée de H_s se traduira par un C_{max} strictement supérieur à un.

4.1.2 Formule de projection sur H_s

Soit N une commande non nulle quelconque. Sa projection N' sur H_s est définie par :

$$N' = \frac{N}{\|N\|_1} \times s$$

Sauf mention contraire, on ne considérera que des commandes admissibles de norme s , c'est-à-dire des éléments de H_s , car toute commande non nulle peut être ramenée dans H_s par projection.

4.1.3 Ensemble des commandes 1-équilibrées

Définition 8 : Soit Q une matrice de qualification et v une matrice de vitesse. On note $B_1(Q_v)$ l'ensemble des commandes 1-équilibrées pour (Q_v, N) , c'est-à-dire telles que $C_{max} = 1$. $B_1(Q_v)$ est également l'ensemble des commandes équilibrées de H_s .

Propriété : $B_1(Q_v)$ est un ensemble convexe.

Démonstration : Soient N_1 et N_2 deux éléments de $B_1(Q_v)$. Démontrons que pour tout réel α tel que $0 \leq \alpha \leq 1$, la commande N définie par $N = \alpha \times N_1 + (1 - \alpha) \times N_2$ appartient à $B_1(Q_v)$.

Les commandes N_1 et N_2 sont équilibrées pour Q_v et telles que $C_{max} = 1$. On sait que les commandes $\alpha \times N_1$ et $(1 - \alpha) \times N_2$ sont équilibrées (mais elles ne sont plus dans H_s). Par construction, la charge temporelle d'une machine vaut respectivement α pour la commande $\alpha \times N_1$ et $(1 - \alpha)$ pour la commande $(1 - \alpha) \times N_2$. La commande N est la somme de deux

commandes équilibrées, et elle peut être exécutée en $\alpha + (1 - \alpha) = 1$ unité de temps. Donc $N \in B_1(Q_v)$.

4.1.4 Caractérisation de l'ensemble des commandes 1-équilibrées

L'existence de couples (Q_v, N) non équilibrés est liée à la présence de zéros dans la matrice Q , c'est-à-dire à des non qualifications issues de choix délibérés ou de contraintes technologiques. Le fait qu'une machine ne soit pas qualifiée pour un type de produits i rend impossible la fabrication de s produits de type i par unité de temps. Si la commande N (où N est dans H_s) requiert la fabrication de $N(i)$ produits de type i , et que la capacité de production totale du parc pour ce type de produits est inférieure à $N(i)$ à cause des non qualifications, la commande N sera nécessairement non équilibrée. De manière plus générale, une commande n'est pas équilibrée pour Q_v lorsque certaines combinaisons linéaires des éléments de la commande N dépassent certains seuils. L'objet de ce paragraphe est de déterminer ces combinaisons linéaires ainsi que les seuils qui leur sont associés afin de caractériser l'ensemble des commandes 1-équilibrées noté $B_1(Q_v)$, c'est-à-dire les commandes équilibrées de H_s . D'un point de vue méthodologique, on traite un problème de robustesse de « type 3 » défini dans le chapitre II : étant donné une solution (ici Q_v) et un niveau de performance (l'équilibre de la charge des machines), on recherche l'ensemble des instances (l'ensemble de commandes noté $B_1(Q_v)$) pour lesquelles ce niveau de performance est atteint par la solution Q_v .

La méthode proposée pour parvenir à cette fin se décompose en deux étapes. La première consiste à trouver tous les rectangles maximaux de zéros de la matrice de qualification Q . La seconde étape associe une combinaison linéaire des éléments de N ainsi qu'un seuil à chacun des rectangles trouvés au cours de la première étape. Chacune de ces inéquations constitue une frontière de $B_1(Q_v)$, qui est exactement déterminé à l'issue de la deuxième étape.

4.1.4.1 Etape 1 : recherche des rectangles maximaux de zéros de Q

Soit Q une matrice de qualification et v une matrice de vitesse ayant n lignes et m colonnes. On note I_k un sous-ensemble de I , et J_k un sous-ensemble de J .

Définition 9: Le produit cartésien $I_k \times J_k$ est un rectangle de zéros de Q si et seulement si $Q(i, j) = 0$ ($\forall (i, j) \in I_k \times J_k$).

Définition 10: Le produit cartésien $I_k \times J_k$ est un rectangle maximum de zéros de Q si et seulement si $I_k \times J_k$ est un rectangle de zéros de Q , et si les deux conditions ci-dessous sont vérifiées :

- Condition 1 : Il n'existe aucun type de produit i_0 hors de I_k tel que $(\{i_0\} \cup I_k) \times J_k$ est un rectangle de zéros de Q
- Condition 2 : Il n'existe aucune machine j_0 hors de J_k telle que $I_k \times (\{j_0\} \cup J_k)$ est un rectangle de zéros de Q

L'algorithme utilisé pour déterminer les rectangles maximaux de zéros de Q est tiré de [Nourine et Raynaud, 1999], et fait appel à des résultats provenant de l'étude des treillis de Galois (ou treillis de concepts). Les rectangles maximaux de zéros (connexes ou non) de la matrice de qualification sont des concepts au sens de la « non qualification ». L'ensemble de ces concepts constitue les sommets du treillis de Galois associé au contexte (c'est-à-dire à la matrice de qualification). Nourine et Raynaud proposent un algorithme permettant de calculer ces sommets, en dressant l'arbre lexicographique d'une famille F engendrée par une base B . Ici, la base B a m éléments : l'élément j de cette base est l'ensemble des types de produits que peut traiter la machine j . La famille F est une liste finie d'ensembles de types de produits, stable pour l'union. A chaque élément f de la famille F , on associe l'ensemble $\gamma(f)$, constitué de toutes les machines qui ne sont qualifiées que pour des produits de f . Dans la définition des rectangles maximaux de zéros de Q , F est la liste des ensembles de types de produits $\{I_1, I_2, \dots, I_h\}$, et $\gamma(F)$ est la liste associée des ensembles de machines $\{J_1, J_2, \dots, J_h\}$. La cardinalité de F et de $\gamma(F)$, notée h , correspond au nombre de rectangles maximaux de zéros que l'on peut trouver dans Q .

On notera e l'élément neutre pour l'union. L'algorithme proposé dans [Nourine et Raynaud, 1999] pour calculer l'arbre lexicographique de F , c'est-à-dire tous les couples $(f, \gamma(f)) = (I_k, J_k)$, est le suivant :

Initialisation : $F = \{e\}$ et $\gamma(F) = \{e\}$

Pour b appartenant à la base B faire

Pour f appartenant F faire

$$f' = f \cup b$$

si $f' \notin F$ alors

$$F = F \cup \{f'\}$$

$$\gamma(f') = \gamma(f) \cup \{b\}$$

Fin pour

Fin pour

Exemple : Soient v la matrice de vitesse et Q la matrice de qualification définis par :

$$v = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

On construit la base B en recherchant, pour chaque machine (donc pour chaque colonne de Q), l'ensemble des produits pour lesquels la machine est qualifiée. Ici, la base B est définie par : $B = \{ \{1,3\}, \{5\}, \{2,3,4\} \}$. En effet, la machine 1 est qualifiée pour les produits de type 1 et 3, d'où le premier élément de la base. L'algorithme présenté ci-dessus permet d'obtenir les familles F et $\gamma(F)$, dont on peut déduire les rectangles de zéros maximaux de Q exprimés sous la forme d'un produit cartésien d'un ensemble de produits et d'un ensemble de machines :

$$\begin{aligned} I_1 &= \{2,4,5\} \quad I_2 = \{1,2,3,4\} \quad I_3 = \{2,4\} \\ I_4 &= \{1,5\} \quad I_5 = \{5\} \quad I_6 = \{1\} \\ J_1 &= \{1\} \quad J_2 = \{2\} \quad J_3 = \{1,2\} \quad J_4 = \{3\} \quad J_5 = \{1,3\} \quad J_6 = \{2,3\} \end{aligned}$$

On compte 6 rectangles définis par le produit cartésien $I_k \times J_k$, avec $k \in [1,6]$. Par exemple, pour $k = 1$, le rectangle est réduit aux zéros de la première colonne de Q : $J_1 = \{1\}$ désigne la première machine, et $I_1 = \{2,4,5\}$ désigne les produits pour lesquels la machine 1 n'est pas qualifiée.

L'ensemble F contient les ensembles de produits correspondant aux rectangles maximaux de zéros de Q . Au lancement de l'algorithme, F est vide.

L'élément f s'obtient en faisant l'union des ensembles de produits d'un élément f avec l'ensemble des produits qualifiés pour l'élément b de la base. Si l'ensemble de produits ainsi obtenu n'existe pas déjà, c'est qu'un nouveau rectangle maximal de zéros a été découvert dans Q . $\gamma(f)$ est l'ensemble de machines correspondant : il faut noter qu'en faisant l'union des qualifications d'un ensemble J_k de machines, on obtient l'ensemble I_k des produits qu'il est possible de traiter avec au moins une machine de cet ensemble. Autrement dit, les produits qu'il est impossible de traiter avec l'ensemble J_k de machines est $I - I_k$.

En conséquence, lorsque l'algorithme a déterminé l'ensemble des sommets du treillis de Galois associé à la matrice de qualification, il convient de remplacer tout élément f de F par son complément dans I , afin de caractériser les rectangles maximaux de zéros de Q . A la fin de la procédure, la famille F compte h éléments notés f_k , auxquels sont associés les h ensembles de machines $\gamma(f_k)$. Nourine et Raynaud ont démontré que la construction de l'arbre lexicographique à partir de la base B nécessite $O((n+m) \times m \times h)$ opérations. D'autre part, la détermination de h , sans construire l'arbre lexicographique associé est un problème ouvert.

4.1.4.2 Etape 2 : génération des contraintes définissant $B_1(Q_v)$

On suppose que tous les rectangles maximaux de zéros de Q sont déterminés. On peut en déduire les h ensembles $I_k \times J_k$ (avec $1 \leq k \leq h$). En posant $I_k = f_k$ et $J_k = \gamma(f_k)$.

Définition 11: Soit $I_k \times J_k$ un rectangle maximal de zéros de Q . On définit la contrainte notée (C_k) par :

$$(C_k): \sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k} V(j)$$

La frontière associée à cette contrainte s'écrit simplement :

$$\sum_{i \in I_k} N(i) = s - \sum_{j \in J_k} V(j)$$

Les points situés sur la frontière associée à la contrainte (C_k) vérifient (C_k) , mais sont infiniment proches de points qui eux, ne vérifient pas (C_k) .

Théorème 1: Soit N un élément de H_s . N appartient à $B_1(Q_v)$ si et seulement si N satisfait les contraintes (C_k) , pour tout entier k dans $[1, h]$. La démonstration de ce théorème n'est pas indiquée ici car ce résultat découle d'un théorème plus général qui sera présenté dans la suite de ce mémoire, avec sa démonstration.

4.1.5 Exemple

Soient la matrice de vitesse v et la matrice de qualification Q définies par :

$$v = \begin{bmatrix} 1 & 2 & 0.5 & 1.2 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0.5 & 1.2 \\ 1 & 2 & 0.5 & 1.2 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{d'où } Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix}$$

La capacité de production, notée s est donnée par :

$$s = \sum_{j=1}^m V(j) = 4.7$$

H_s est donc l'ensemble des commandes N telles que : $\sum_{i=1}^n N(i) = s$

Calculons maintenant $B_1(Q_v)$:

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad B_1(Q_v) : \begin{cases} N(1) \leq 3 & I_1 = \{1\} & J_1 = \{3,4\} \\ N(1) + N(2) \leq 3.5 & I_2 = \{1,2\} & J_2 = \{4\} \\ N(3) \leq 3.2 & I_3 = \{3\} & J_3 = \{1,3\} \\ N(1) + N(3) \leq 4.2 & I_4 = \{1,3\} & J_4 = \{3\} \end{cases}$$

On peut vérifier expérimentalement que $B_1(Q_v)$ est bien l'ensemble des commandes 1-équilibrées. Pour cela, considérons deux commandes N_1' et N_2' définies par :

$$N_1' = \begin{bmatrix} 10 \\ 20 \\ 17 \end{bmatrix} \quad \text{et} \quad N_2' = \begin{bmatrix} 34 \\ 40 \\ 20 \end{bmatrix}$$

Si l'on norme à $s = 4.7$ les commandes ci-dessus, on obtient les commandes N_1 et N_2 données ci-dessous, qui sont des éléments de H_s . On rappelle que les propriétés d'équilibre d'une commande sont conservées par multiplication.

$$N_1 = N_1' \times \frac{\|N_1'\|_1}{s} = \begin{bmatrix} 1 \\ 2 \\ 1.7 \end{bmatrix} \quad \text{et} \quad N_2 = N_2' \times \frac{\|N_2'\|_1}{s} = \begin{bmatrix} 1.7 \\ 2 \\ 1 \end{bmatrix}$$

On peut vérifier facilement que N_1 satisfait les 4 contraintes de $B_1(Q_v)$, et que N_2 ne vérifie pas la seconde contrainte (parce que $N_2(1) + N_2(2) > 3.5$). On s'attend donc à ce que le couple (Q_v, N_1) soit équilibré et à ce que le couple (Q_v, N_2) ne le soit pas. C'est effectivement le cas : après avoir exécuté (PL 1) pour Q_v et N_1 (respectivement Q_v et N_2) on obtient la matrice de répartition R_{T1} (respectivement R_{T2}) donnée ci-après :

$$R_{T1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.75 & 1 & 0 \\ 0 & 0.25 & 0 & 1 \end{bmatrix} \quad R_{T2} = \begin{bmatrix} 0.5 & 0.6 & 0 & 0 \\ 0.557 & 0.457 & 1.057 & 0 \\ 0 & 0 & 0 & 0.8333 \end{bmatrix}$$

$$C_1 = [1 \quad 1 \quad 1 \quad 1] \quad C_2 = [1.057 \quad 1.057 \quad 1.057 \quad 0.8333]$$

Les vecteurs C_1 et C_2 représentent la charge de chaque machine. Ils permettent de confirmer que R_{T1} est équilibrée, et que R_{T2} ne l'est pas. Il est toutefois important de rappeler

que $B_1(Q_v)$ n'a pas été construit pour déterminer si une commande est équilibrée pour une matrice Q_v donnée. En effet, la résolution de (PL 1), qui est polynomiale, permet déjà d'obtenir cette information. La détermination de $B_1(Q_v)$ est beaucoup plus lourde parce qu'elle n'est pas nécessairement polynomiale (la complexité dépend de h qui est inconnu avant d'avoir calculé $B_1(Q_v)$). L'intérêt de calculer $B_1(Q_v)$ réside dans l'obtention de ses frontières (les h inégalités) qui vont permettre de mesurer la robustesse d'une configuration Q , étant donnée une commande prévisionnelle N_{ref} . Afin de présenter cette idée plus clairement, une représentation graphique de H_s et de $B_1(Q_v)$ est proposée dans le cas particulier de l'exemple.

4.1.6 Représentation graphique de H_s et de $B_1(Q_v)$

Lorsque le nombre de types de produits, n , est inférieur à 3 on peut utiliser une représentation graphique de H_s et de $B_1(Q_v)$. Dans le cadre de l'exemple, l'ensemble des commandes est un espace de dimension $n = 3$. L'équation définissant H_s est celle d'un plan. Comme les commandes admissibles n'ont aucun coefficient strictement négatif, H_s est en réalité l'intersection de l'espace des vecteurs dont les composantes sont positives ou nulles, et du plan orthogonal au vecteur dont tous les coefficients valent 1. H_s est représenté sur la Figure 13.

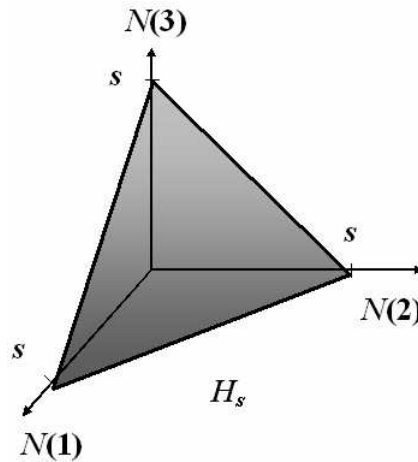


Figure 13 : Représentation de H_s dans l'espace

La Figure 14 représente le tracé des frontières de $B_1(Q_v)$ sur H_s dans le cas précis de l'exemple. La zone grisée représente $B_1(Q_v)$: c'est une portion de plan. La représentation des commandes N_1 et N_2 confirme que N_2 est hors de $B_1(Q_v)$, donc non équilibrée pour Q_v .

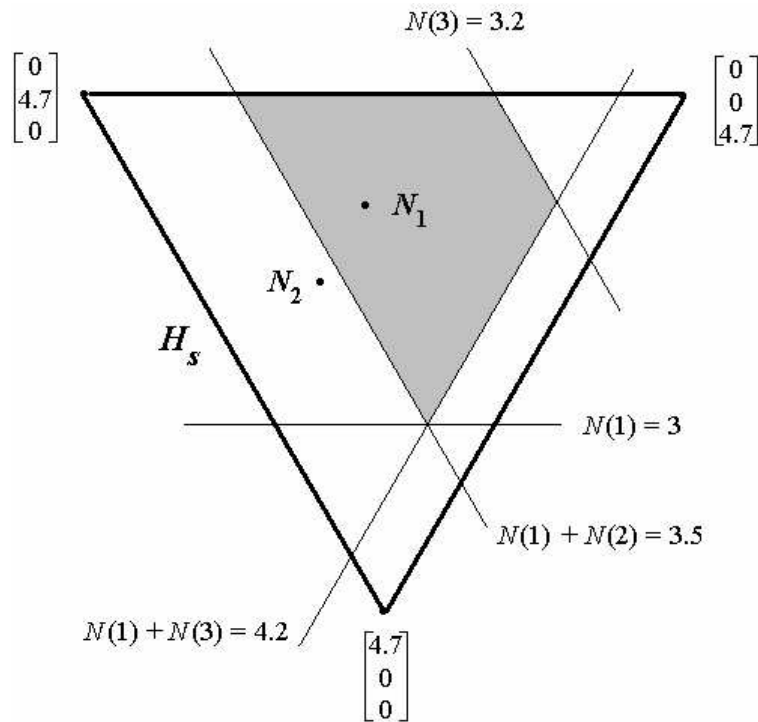


Figure 14 : Représentation de $B_1(Q_v)$ dans H_s

Pour définir la notion de robustesse d'une configuration, il est nécessaire de pouvoir évaluer des distances dans H_s . Le paragraphe suivant est consacré à la mise au point de ces mesures et à leur justification.

4.1.7 Distances dans H_s

Dans ce paragraphe, on présente des résultats utiles au calcul de la distance d'un point de H_s à la frontière d'une contrainte de type (C_k) .

4.1.7.1 Distance séparant deux commandes de H_s

Soient A et B deux éléments de H_s . On définit la distance entre A et B par :

$$d(A, B) = \|B - A\|_1$$

4.1.7.2 Distance d'une commande à la frontière d'une contrainte

Soit N une commande de H_s . On note (C_k) une contrainte de H_s définie par l'inéquation suivante :

$$(C_k): \sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k} V(j)$$

Afin de simplifier l'expression des calculs, on posera $\alpha = s - \sum_{j \in J_k} V(j)$.

On note (EP) l'ensemble des éléments P de H_s défini par :

$$(EP): \sum_{i \in I_k} P(i) = \alpha$$

Où I_k est un ensemble d'entiers inclus dans I , et α un réel vérifiant $0 \leq \alpha \leq s$.

La distance de tout point N (avec $N \in H_s$) à (EP) s'écrit :

$$d(N, EP) = 2 \times \left| \alpha - \sum_{i \in I_k} N(i) \right|$$

Remarque : (EP) est la frontière du demi-espace défini par $(C_k): \sum_{i \in I_k} N(i) \leq \alpha$

Démonstration

Il est évident que si le point N est un élément de (EP) , alors sa distance de N à (EP) est nulle. Nous supposons dans la suite que le point N n'appartient pas à (EP) . Cela conduit à considérer successivement deux cas.

- Dans le premier cas, on suppose que N est tel que $\sum_{i \in I_k} N(i) > \alpha$. Il sera alors nécessaire d'utiliser le lemme 1, dont l'énoncé et la preuve sont présentés avant la démonstration dans laquelle ce lemme intervient.
- Dans le second cas, on suppose que N est tel que $\sum_{i \in I_k} N(i) < \alpha$. Il sera alors nécessaire d'utiliser le lemme 2, dont l'énoncé et la preuve sont présentés avant la démonstration dans laquelle ce lemme intervient.

Lemme 1: Pour tout point N de H_s vérifiant $\sum_{i \in I_k} N(i) > \alpha$ et tout point Y de (EP) , on a l'égalité suivante :

$$\sum_{i \in I_k} |N(i) - Y(i)| = \left(\sum_{i \in I_k} N(i) - \alpha \right) + 2\beta \quad \text{avec } \beta \geq 0$$

Preuve du lemme 1:

On considère la partition de I_k en deux sous-ensembles notés I_k^+ et I_k^- tels que :

$$\begin{cases} Y(i) > N(i) \Rightarrow i \in I_k^+ \\ Y(i) \leq N(i) \Rightarrow i \in I_k^- \end{cases} \quad \text{On a alors :}$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \sum_{i \in I_k^+} (Y(i) - N(i)) + \sum_{i \in I_k^-} (N(i) - Y(i))$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \sum_{i \in I_k^+} (Y(i) - N(i)) + \left(\sum_{i \in I_k} N(i) - \alpha \right) - \sum_{i \in I_k^+} (N(i) - Y(i))$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \left(\sum_{i \in I_k} N(i) - \alpha \right) + 2 \sum_{i \in I_k^+} (Y(i) - N(i))$$

On pose $\beta = \sum_{i \in I_k^+} (Y(i) - N(i))$.

$\beta \geq 0$ car les éléments i de I_k^+ vérifient $Y(i) - N(i) > 0$. β peut être nul si I_k^+ est vide. Le lemme 1 est démontré.

Lemme 2: Pour tout point N de H_s vérifiant $\sum_{i \in I_k} N(i) < \alpha$ et tout point Y de (EP) , on a l'égalité suivante :

$$\sum_{i \in I_k} |N(i) - Y(i)| = \left(\alpha - \sum_{i \in I_k} N(i) \right) + 2\beta \quad \text{avec } \beta \geq 0$$

Preuve du lemme 2:

On considère la partition de I_k en deux sous-ensembles notés I_k^+ et I_k^- tels que :

$$\begin{cases} Y(i) \geq N(i) \Rightarrow i \in I_k^+ \\ Y(i) < N(i) \Rightarrow i \in I_k^- \end{cases} \quad \text{On a alors :}$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \sum_{i \in I_k^+} (Y(i) - N(i)) + \sum_{i \in I_k^-} (N(i) - Y(i))$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \alpha - \sum_{i \in I_k^-} Y(i) - \left(\sum_{i \in I_k} N(i) - \sum_{i \in I_k^-} N(i) \right) + \sum_{i \in I_k^-} N(i) - \sum_{i \in I_k^-} Y(i)$$

$$\sum_{i \in I_k} |N(i) - Y(i)| = \left(\alpha - \sum_{i \in I_k} N(i) \right) + 2 \sum_{i \in I_k^-} (N(i) - Y(i))$$

$$\text{On pose } \beta = \sum_{i \in I_k^-} (N(i) - Y(i)).$$

$\beta \geq 0$ car les éléments i de I_k^- vérifient $N(i) - Y(i) > 0$. β peut être nul si I_k^- est vide. Le lemme 2 est démontré.

Ces résultats étant acquis, la démonstration de l'expression de la distance d'un point N de H_s à (EP) est proposée pour chacun des deux cas suivants :

Premier cas : le point N est tel que : $\sum_{i \in I_k} N(i) > \alpha$

Démarche : nous allons montrer qu'il est toujours possible de construire un point X appartenant à (EP) , dans H_s , et tel que sa distance à N soit égale à $2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$. Ensuite, nous établirons qu'il n'existe aucun point Y appartenant à (EP) tel que la distance de Y à N soit strictement inférieure à $2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$.

Soit $\{\varepsilon_i\}_{i \in I}$ une suite de n réels positifs ou nuls vérifiant les trois conditions :

$$\begin{cases} 0 \leq \varepsilon_i \leq N(i) \quad (\forall i \in I_k) & (C1) \\ \sum_{i \in I_k} \varepsilon_i = \alpha & (C2) \\ \sum_{i \in I - I_k} \varepsilon_i = \sum_{i \in I_k} N(i) - \alpha > 0 & (C3) \end{cases}$$

Remarque : $\{\varepsilon_i\}_{i \in I}$ existe toujours. En effet :

La condition C1 peut toujours être atteinte.

La condition C2 peut être atteinte sans violer C1 car $\sum_{i \in I_k} N(i) > \alpha$

La condition C3 peut toujours être atteinte car c'est la seule contrainte qui pèse sur les ε_i pour i hors de I_k , hormis la contrainte de positivité.

On construit alors le point X appartenant à (EP) et à H_s à partir de N et de la suite $\{\varepsilon_i\}_{i \in I}$:

$$\begin{cases} X(i) = \varepsilon_i & (\forall i \in I_k) & (C4) \\ X(i) = N(i) + \varepsilon_i & (\forall i \in I - I_k) & (C5) \end{cases}$$

Les conditions C1 et C4 assurent que $X(i) \leq N(i)$ ($\forall i \in I_k$)

La condition C5 assure que $X(i) \geq N(i)$ ($\forall i \in I - I_k$)

La condition C2 assure que X est un point de (EP) .

On peut vérifier que X appartient bien à H_s en calculant sa norme.

Calculons la distance de N à X . Pour cela, on utilisera la formule donnant la distance séparant deux points de H_s :

$$d(A, B) = \|B - A\|_1 = \sum_{i=1}^n |B(i) - A(i)|$$

$$d(N, X) = \sum_{i \in I_k} |N(i) - X(i)| + \sum_{i \in I - I_k} |N(i) - X(i)| = \sum_{i \in I_k} (N(i) - X(i)) + \sum_{i \in I - I_k} (X(i) - N(i)) = 2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$$

Ceci démontre que l'on peut construire un point X appartenant à (EP) tel que X se trouve à la distance $2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$ du point N .

Montrons par l'absurde qu'il n'existe aucun point Y appartenant à (EP) tel que sa distance à N soit strictement inférieure à $2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$.

Supposons qu'il existe un point Y de (EP) vérifiant $d(N, Y) < 2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$.

Calculons alors la distance de N à Y :

$$d(N, Y) = \sum_{i \in I_k} |N(i) - Y(i)| + \sum_{i \in I - I_k} |N(i) - Y(i)|$$

D'après le lemme 1, on a :

$$d(N, Y) = \sum_{i \in I - I_k} |Y(i) - N(i)| + \sum_{i \in I_k} N(i) - \alpha + 2\beta$$

$$d(N, Y) \geq \sum_{i \in I - I_k} (Y(i) - N(i)) + \sum_{i \in I_k} N(i) - \alpha + 2\beta$$

$$d(N, Y) \geq (s - \alpha) - \left(s - \sum_{i \in I_k} N(i) \right) + \sum_{i \in I_k} N(i) - \alpha + 2\beta$$

$$d(N, Y) \geq 2 \times \left(\sum_{i \in I_k} N(i) - \alpha + \beta \right)$$

Or le point Y appartenant à (EP) vérifie par définition la relation :

$$d(N, Y) < 2 \times \left(\sum_{i \in I_k} N(i) - \alpha \right)$$

On en déduit que $\beta < 0$, ce qui est absurde d'après le lemme 1. La relation

$$d(N, P) = 2 \times \left| \alpha - \sum_{i \in I_k} N(i) \right| \text{ est donc démontrée dans le cas où } \sum_{i \in I_k} N(i) > \alpha$$

Deuxième cas $\sum_{i \in I_k} N(i) < \alpha$

On applique la même démarche que pour le cas précédent. Soit $\{\mathcal{E}_i\}_{i \in I}$ une suite de n réels positifs ou nuls vérifiant les trois conditions :

$$\begin{cases} 0 \leq N(i) \leq \varepsilon_i & (\forall i \in I_k) & (C1) \\ \sum_{i \in I_k} \varepsilon_i = \alpha & & (C2) \\ \sum_{i \in I - I_k} \varepsilon_i = - \sum_{i \in I_k} N(i) + \alpha > 0 & & (C3) \end{cases}$$

Remarque : $\{\varepsilon_i\}_{i \in I}$ existe toujours. En effet :

La condition C1 peut toujours être atteinte.

La condition C2 peut être atteinte sans violer C1 car $\sum_{i \in I_k} N(i) < \alpha$

La condition C3 peut toujours être atteinte car c'est la seule contrainte qui pèse sur les ε_i pour i hors de I_k , hormis la contrainte de positivité.

On construit alors le point X appartenant à (EP) et à H_s à partir de N et de la suite $\{\varepsilon_i\}_{i \in I}$:

$$\begin{cases} X(i) = \varepsilon_i & (\forall i \in I_k) & (C4) \\ X(i) = N(i) - \varepsilon_i & (\forall i \in I - I_k) & (C5) \end{cases}$$

Les conditions C1 et C4 assurent que $X(i) \geq N(i)$ $(\forall i \in I_k)$

La condition C5 assure que $X(i) \leq N(i)$ $(\forall i \in I - I_k)$

La condition C2 assure que X est un point de (EP) .

On peut vérifier que X appartient bien à H_s en calculant sa norme.

Calculons la distance de N à X . Pour cela, on utilisera la formule donnant la distance séparant deux points de H_s :

$$d(A, B) = \|B - A\|_1 = \sum_{i=1}^n |B(i) - A(i)|$$

$$d(N, X) = \sum_{i \in I_k} |N(i) - X(i)| + \sum_{i \in I - I_k} |N(i) - X(i)| = \sum_{i \in I_k} (X(i) - N(i)) + \sum_{i \in I - I_k} (N(i) - X(i)) = 2 \times \left(\alpha - \sum_{i \in I_k} N(i) \right)$$

Ceci démontre que l'on peut construire un point X appartenant à (EP) tel que X se trouve à la distance $2 \times \left(\alpha - \sum_{i \in I_k} N(i) \right)$ du point N .

Montrons par l'absurde qu'il n'existe aucun point Y appartenant à (EP) tel que sa distance à N soit strictement inférieure à $2 \times \left(\alpha - \sum_{i \in I_k} N(i) \right)$.

Supposons qu'il existe un point Y de (EP) vérifiant $d(N, Y) < 2 \times \left(\alpha - \sum_{i \in I_k} N(i) \right)$.

Calculons alors la distance de N à Y :

$$d(N, Y) = \sum_{i \in I_k} |N(i) - Y(i)| + \sum_{i \in I - I_k} |N(i) - Y(i)|$$

D'après le lemme 2, on a :

$$d(N, Y) = \alpha - \sum_{i \in I_k} N(i) + 2\beta + \sum_{i \in I - I_k} |N(i) - Y(i)|$$

$$d(N, Y) \geq \alpha - \sum_{i \in I_k} N(i) + 2\beta + \sum_{i \in I - I_k} (N(i) - Y(i))$$

$$d(N, Y) \geq \alpha - \sum_{i \in I_k} N(i) + 2\beta + \left(s - \sum_{i \in I_k} N(i) \right) - (s - \alpha)$$

$$d(N, Y) \geq 2 \times \left(\alpha - \sum_{i \in I_k} N(i) + \beta \right)$$

Or le point Y appartenant à (EP) vérifie par définition la relation :

$$d(N, Y) < 2 \times \left(\alpha - \sum_{i \in I_k} N(i) \right)$$

On en déduit que $\beta < 0$, ce qui est absurde d'après le lemme 2. La relation

$$d(N, P) = 2 \times \left| \alpha - \sum_{i \in I_k} N(i) \right| \text{ est donc démontrée quel que soit le signe de } \sum_{i \in I_k} N(i) - \alpha.$$

Cette étude permet de mesurer des distances dans H_s , et d'introduire la notion de rayon de stabilité qui s'appuie sur la distance d'un point à un hyperplan de H_s .

4.1.8 Rayon de stabilité

Les paragraphes précédents ont permis de caractériser l'ensemble des commandes équilibrées d'une matrice de qualification associée à une matrice de vitesse donnée. Au début de chaque mois, la commande prévisionnelle de l'atelier de photolithographie est connue. Cette commande prévisionnelle, notée N_{ref} , dépend des commandes de produits finis passées par les clients du système de production. Cependant, les commandes qui seront effectivement passées chaque jour à l'atelier de photolithographie peuvent s'écarter sensiblement de la commande prévisionnelle. Cet écart est dû aux pannes de machines, et plus généralement aux aléas de production qui surviennent dans les ateliers amont. Or, s'il est aisé de vérifier que la commande prévisionnelle est équilibrée, la commande effectivement passée à l'atelier peut ne pas l'être. Dans ce paragraphe, on propose une mesure de robustesse, permettant de quantifier l'écart minimum susceptible de faire perdre la propriété d'équilibre à la commande effective traitée par l'atelier. Nous utilisons pour cela la notion de rayon de stabilité introduite dans [Sotskov *et al.*, 1998] :

Définition 12 : Soient :

R^q , l'ensemble des vecteurs de dimension q n'ayant aucun coefficient négatif.

p , un élément de R^q désignant une instance du problème (un jeu de données numériques).

s , une solution du problème (un ensemble de décisions portant sur ces données).

Alors la boule fermée $O_r(p)$ de rayon r et de centre p dans l'espace de dimension q est appelée boule de stabilité de s si, pour tout vecteur p' appartenant à l'intersection de $O_r(p)$ et de R^q , la solution s reste optimale. La valeur maximale du rayon r de la boule de stabilité $O_r(p)$ de la solution s est appelée le rayon de stabilité de s .

En remplaçant R^q par R^n , p par N_{ref} , s par Q_v et en considérant que le critère de performance qui permet de juger de l'optimalité d'une solution est la propriété d'équilibre, on peut adapter la définition ci-dessus au problème de l'atelier de photolithographie.

On utilise cet outil pour caractériser un ensemble d'instances pour lequel une configuration Q garantit l'équilibre. Cet ensemble est défini par le voisinage d'une instance particulière N_{ref} pour laquelle la solution est équilibrée.

4.1.8.1 Distances d'une commande aux frontières de $B_I(Q_v)$

Le théorème 1, énoncé dans la partie précédente, permet d'affirmer que l'ensemble des commandes N qui satisfont toutes les contraintes (C_k) constitue l'ensemble des commandes 1-équilibrées de Q_v .

Chaque contrainte (C_k) définit une partition de H_s en deux demi-espaces : l'ensemble des commandes N de H_s qui satisfont (C_k) d'une part, l'espace des commandes de H_s qui ne satisfont pas (C_k) d'autre part.

On rappelle que la contrainte (C_k) s'écrit : $\sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k} V(j)$.

L'ensemble des commandes N de H_s défini par : $\sum_{i \in I_k} N(i) = s - \sum_{j \in J_k} V(j)$ constitue la frontière de (C_k) . Le calcul du rayon de stabilité de la matrice Q_v pour la commande prévisionnelle N_{ref} est fondé sur la distance de N_{ref} à chacune des frontières de (C_k) , qui constituent aussi les frontières de $B_1(Q_v)$.

Pour une commande N_{ref} de H_s , et une contrainte (C_k) définissant $B_1(Q_v)$, on définit le réel ω_k par :

$$\omega_k = 2 \times \left(s - \sum_{j \in J_k} V(j) - \sum_{i \in I_k} N(i) \right) \geq 0 \quad \text{si } N \text{ satisfait la contrainte } (C_k),$$

$$\omega_k = 2 \times \left(-s + \sum_{j \in J_k} V(j) + \sum_{i \in I_k} N(i) \right) \geq 0 \quad \text{sinon.}$$

D'après l'étude des distances dans H_s , ω_k est égal à la distance de N à la frontière de (C_k) si N satisfait la contrainte (C_k) , ω_k est égal à l'opposé de cette distance sinon.

Le vecteur $\Omega(Q_v, N_{ref})$ regroupe les h valeurs de ω_k .

4.1.8.2 Définition du rayon de stabilité

Définition 13 : On définit le rayon de stabilité d'une matrice de qualification Q d'un vecteur de vitesses v et d'une commande N_{ref} par :

$$r(Q_v, N_{ref}) = \min(\Omega(Q_v, N_{ref})) = \min_{0 \leq k \leq h} (\omega_k)$$

Si N_{ref} est équilibrée pour Q_v , alors le rayon de stabilité est positif. Il est strictement négatif si N_{ref} n'est pas équilibrée pour Q_v .

Théorème 2: Si $r(Q_v, N_{ref}) \geq 0$, alors on peut garantir que toute commande N de H_s définie par $N = N_{ref} + dN$ est équilibrée, à condition que $\|dN\|_1 \leq r(Q_v, N_{ref})$.

Démonstration : En effet, le rayon de stabilité est la plus grande distance de N_{ref} à N telle que l'on peut garantir la propriété d'équilibre pour toute commande de H_s située à une distance de N_{ref} inférieure ou égale à $r(Q_v, N_{ref})$. On a donc :

$$d(N_{ref}, N) \leq r(Q_v, N_{ref})$$

En appliquant la définition de la distance séparant deux points de H_s , on peut écrire :

$$d(N_{ref}, N) = \|N_{ref} - N\|_1$$

Or la commande N s'écrit $N = N_{ref} + dN$, d'où :

$$d(N_{ref}, N) = \|dN\|_1$$

Ce qui permet d'obtenir la relation :

$$\|dN\|_1 \leq r(Q_v, N_{ref})$$

L'ensemble des commandes N ainsi définies constitue un voisinage centré sur N_{ref} , pour lequel on peut garantir l'équilibre de N . Notons cependant qu'il peut exister des commandes N équilibrées pour lesquelles $\|dN\|_1 > r(Q_v, N_{ref})$. Il est important de noter que la réciproque du théorème 2 est fausse. Il n'y a pas équivalence entre l'équilibre d'une commande N et le respect de l'inégalité $\|dN\|_1 \leq r(Q_v, N_{ref})$.

4.1.8.3 Définition de la tolérance

Le rayon de stabilité permet de définir un voisinage stable autour de N_{ref} , dans H_s . Or par définition, le volume de production de toute commande de H_s est égal à s . La valeur du rayon de stabilité obtenue est donc relative à une charge totale valant s . On peut définir la tolérance $t(Q_v, N_{ref})$ d'un couple (Q_v, N_{ref}) où N_{ref} est une commande de charge quelconque ($N_{ref} \notin H_s$), par :

$$t(Q_v, N_{ref}) = \frac{100}{s} \times r(Q_v, N_{ref}) \quad \text{où la tolérance s'exprime en pourcents.}$$

Par exemple, si la commande réelle N_{ref} d'un atelier de photolithographie configuré suivant Q a une charge quelconque, on doit normer à s cette commande pour calculer le rayon de stabilité. Le calcul de la tolérance permet ensuite de garantir que toute commande

$N = N_{ref} + dN$ de même norme que N_{ref} sera équilibrée pour Q_v à condition que $\|dN\|_1 \leq t(Q_v, N_{ref})$, où $t(Q_v, N_{ref})$ est exprimée en pourcents de la charge de N_{ref} .

4.1.8.4 Contraintes critiques

On dit qu'une contrainte (C_k) définissant $B_1(Q_v)$ est critique si et seulement si :

$$2 \times \left(s - \sum_{j \in J_k} V(j) - \sum_{i \in I_k} N(i) \right) = r(Q_v, N_{ref})$$

En d'autres termes, une contrainte est critique si la distance de sa frontière à N_{ref} est minimale sur l'ensemble des frontières de $B_1(Q_v)$.

Remarque : L'ensemble des contraintes critiques n'est vide que si $B_1(Q_v) = H_s$. Ce cas se produit lorsque aucun coefficient de Q n'est nul. Le rayon de stabilité est alors infini, quelle que soit la commande prévisionnelle N_{ref} .

Remarque : Il peut exister plusieurs contraintes critiques. On notera h_r leur nombre ($1 \leq h_r \leq h$).

4.1.8.5 Conclusion partielle

Les outils permettant de mesurer la robustesse d'une configuration ont été présentés. En définissant $B_1(Q_v)$, on a répondu à la question suivante « quel est l'ensemble des instances pour lesquelles Q_v assure un traitement équilibré ? ». Avec le rayon de stabilité, on répond à la question plus locale mais non moins pertinente pour l'exploitation de l'atelier : « Quelle est l'amplitude maximale des perturbations sur la commande qui ne remet pas en cause l'équilibre du traitement des commandes au voisinage d'une commande N_{ref} ? ». Dans les deux cas, c'est un problème de robustesse de « type 3 » qui a été traité. Dans la partie suivante, on étudie un problème légèrement différent. On recherche pour une configuration donnée l'ensemble des commandes pour lesquelles la robustesse est maximale. On peut alors savoir quelle robustesse maximale peut offrir la configuration, et ainsi juger de son adéquation à la commande N_{ref} . On répond donc à la question : « la commande N_{ref} fait-elle partie des commandes pour lesquelles la configuration courante garantit la plus grande robustesse ? ».

4.1.9 Ensemble des commandes centrales d'une matrice de vitesse

4.1.9.1 Définition d'une commande centrale

Définition 14 : soient Q une matrice de qualification et v une matrice de vitesse donnée. On appelle commande centrale tout élément N_C appartenant à H_s vérifiant :

$$r(Q_v, N_C) = \max_{\forall N \in H_s} r(Q_v, N) = r^*(Q_v)$$

N_C est une commande assurant un rayon de stabilité maximal pour une matrice Q_v donnée. La valeur du rayon de stabilité atteinte pour N_C est une mesure de robustesse intrinsèque de Q_v . La comparaison de la valeur du rayon de stabilité obtenue pour N_{ref} et pour N_C permet de juger de l'adéquation de Q_v à N_{ref} . On notera $r^*(Q_v)$ la plus grande valeur du rayon de stabilité que l'on peut obtenir avec Q_v .

Remarque : Pour une matrice Q_v n'ayant aucun coefficient nul, toute commande de H_s est centrale, quel que soit v .

4.1.9.2 Caractérisation de l'ensemble des commandes centrales

Propriété : Le rayon de stabilité de toute commande centrale N_C vérifie :

$$r(Q_v, N_C) \leq 2 \times \left(s - \sum_{j \in J_k} V(j) \right) \quad (\forall k \in [1, h])$$

Les contraintes dont le rectangle maximum de zéros correspond aux groupes de machines ayant la capacité de production la plus élevée bornent la valeur du rayon de stabilité que l'on peut atteindre avec la matrice de qualification Q et le vecteur de vitesses v associé aux machines.

On suppose que l'on connaît la matrice Q_v et l'ensemble de ses commandes équilibrées $B_1(Q_v)$. On recherche une commande notée N_C telle que :

N_C appartient à H_s

La plus petite distance de N_C à une contrainte est notée z . On veut que z soit maximum.

$r^*(Q_v)$ est inférieur à $2 \times \left(s - \sum_{j \in J_k} V(j) \right)$ pour tout k , donc en particulier

$$r^*(Q_v) \leq 2 \times \left(s - \max_{k \in [1, h]} \left(\sum_{j \in J_k} V(j) \right) \right)$$

Ces conditions peuvent s'exprimer à l'aide du programme linéaire (PL 3) ci-dessous :

$$(PL\ 3) \left\{ \begin{array}{l} \sum_{i=1}^n N_C(i) = s \\ 2 \times \left(s - \sum_{j \in J_k} V(j) - \sum_{i \in I_k} N_C(i) \right) \geq r^*(Q_v) \\ r^*(Q_v) \leq 2 \times \left(s - \max_{k \in [1, h]} \left(\sum_{j \in J_k} V(j) \right) \right) \\ \max(r^*(Q_v)) \end{array} \right.$$

Remarques : La dernière propriété est à l'origine de la troisième contrainte de (PL 3). Celle-ci interdit de converger systématiquement vers des solutions du type $r^*(Q_v) \rightarrow +\infty$, qui sont sans intérêt. Ce programme linéaire en nombres réels admet toujours une solution optimale, que l'on peut obtenir en temps polynomial. S'il n'existait pas de solution, cela signifierait qu'il n'existe aucune commande équilibrée. Cela implique que Q n'est pas admissible, ce qui est contraire aux hypothèses formulées. La solution est un vecteur N_C (n éléments positifs ou nuls), et une valeur de $r^*(Q_v)$.

L'ensemble des commandes de stabilité maximum d'une matrice Q_v est l'ensemble des commandes centrales, c'est-à-dire l'ensemble des commandes pour lesquelles la plus petite distance aux contraintes définissant $B_1(Q_v)$ vaut $r^*(Q_v)$. Cet ensemble peut être caractérisé à partir des équations de $B_1(Q_v)$, si l'on connaît la valeur de $r^*(Q_v)$.

4.1.9.3 Coefficient d'adéquation de Q_v à N_{ref}

Connaissant la valeur maximale du rayon de stabilité atteignable pour la matrice Q_v , on peut savoir dans quelle mesure la commande nominale N_{ref} exploite la capacité de la configuration à conserver la propriété d'équilibre. On mesure l'adéquation d'une commande N_{ref} à une configuration à l'aide d'un rapport noté $A(Q_v, N_{ref})$, et défini par :

$$A(Q_v, N_{ref}) = \frac{r(Q_v, N_{ref})}{r(Q_v, N_C)} = \frac{t(Q_v, N_{ref})}{t(Q_v, N_C)}$$

Ce rapport est inférieur ou égal à 1. S'il est négatif, c'est que N_{ref} n'est pas dans $B_1(Q_v)$, et n'est donc pas équilibrée pour la matrice Q_v .

4.1.10 Exemple

4.1.10.1 Rayon de stabilité

On reprend ici la matrice de qualification Q , la matrice de vitesse v et la commande N_1 définies au paragraphe 4.1.5, ainsi que les contraintes de $B_1(Q_v)$. Pour calculer le rayon de stabilité $r(Q_v, N_1)$, on détermine d'abord le vecteur $\Omega(Q_v, N_1)$ à partir de $B_1(Q_v)$:

$$B_1(Q_v) : \begin{cases} N(1) \leq 3 \\ N(1) + N(2) \leq 3.5 \\ N(3) \leq 3.2 \\ N(1) + N(3) \leq 4.2 \end{cases} \quad \Omega(Q_v, N_1) = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 3 \end{bmatrix} \quad r(Q_v, N_1) = \min(\Omega(Q_v, N_1)) = 1$$

La figure ci-dessous donne une représentation graphique du rayon de stabilité.

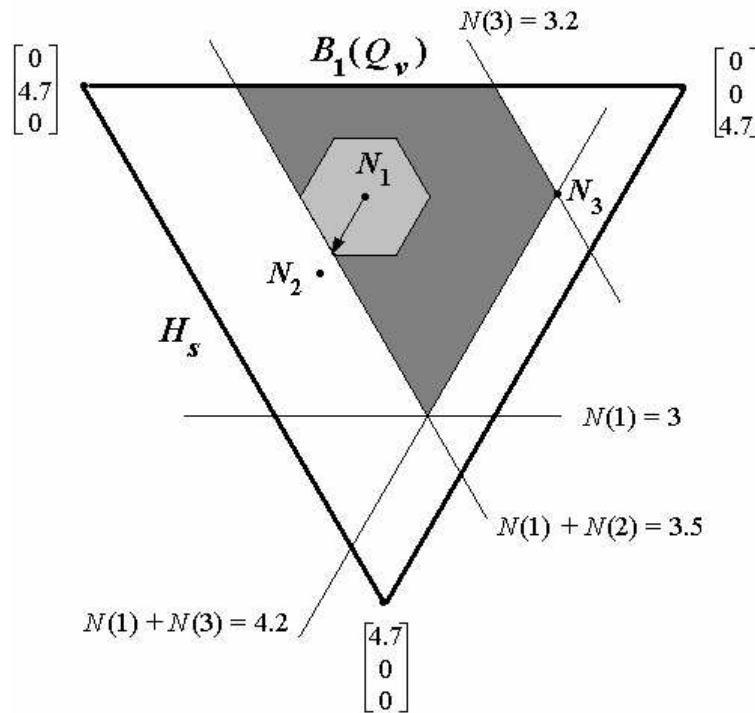


Figure 15 : Représentation du rayon de stabilité dans H_s

L'ensemble des commandes $B_1(Q_v)$ est représenté en gris foncé. L'hexagone régulier dont l'intérieur est gris clair fait aussi partie de $B_1(Q_v)$: il s'agit du voisinage centré sur N_1 défini par le rayon de stabilité. L'hexagone gris clair est l'ensemble des commandes N définies par $N = N_1 + dN$, avec $\|dN\|_1 \leq r(Q_v, N_1)$.

Le rayon de stabilité est représenté par une flèche noire partant de la commande de référence N_1 . L'intérieur de l'hexagone est la « boule » de rayon 1 centrée en N_1 . Cette boule n'est pas circulaire parce que la norme utilisée est la norme 1, et non la norme 2. Plus précisément, l'hexagone est l'intersection de la boule de rayon 1 (au sens de la norme 1) centrée sur N_1 avec H_s .

La commande N_2 est telle que $N_2 = N_1 + dN_2$ avec $dN_2 = \begin{bmatrix} 0.7 \\ 0 \\ -0.7 \end{bmatrix}$. Comme on a :

$$\|dN_2\|_1 = 1.4 > r(Q_v, N_1)$$

On ne peut pas garantir l'équilibre de N_2 . Ce résultat était prévisible, car on sait que N_2 n'est pas équilibrée pour Q_v . Cependant, la Figure 15 montre clairement qu'il existe des points de $B_1(Q_v)$ n'appartenant pas à la boule de rayon $r(Q_v, N_1)$ centrée sur N_1 , mais qui sont équilibrés : considérons la commande N_3 définie par :

$$N_3 = \begin{bmatrix} 1 \\ 0.5 \\ 3.2 \end{bmatrix} \quad N_3 = N_1 + dN_3 \quad \text{d'où } dN_3 = \begin{bmatrix} 0 \\ -1.5 \\ 1.5 \end{bmatrix}$$

Il est clair que $\|dN_3\|_1 = 3 > r(Q_v, N_1)$.

Donc l'équilibre du traitement de N_3 n'est pas garanti. Cependant, N_3 est équilibré (N_3 est à l'intersection de deux frontières de $B_1(Q_v)$).

4.1.10.2 Tolérance

Connaissant le rayon de stabilité, on peut en déduire la tolérance :

$$t(Q_v, N_1) = \frac{100}{4.7} \times 1 = 21.28\%$$

On peut maintenant revenir à la commande N_1' qui n'appartenait pas à H_s , et qui était définie par :

$$N_1' = \begin{bmatrix} 10 \\ 20 \\ 17 \end{bmatrix}$$

Si N_1' est la commande de référence, alors l'équilibre de la commande réelle N définie par $N = N_1' + dN$ (où dN conserve la charge) est garanti à condition que l'on ait : $\|dN\|_1 \leq t(Q_v, N_1)$, c'est-à-dire que la norme de dN doit être inférieure à 21.28 % de $(10 + 20 + 17)$, c'est-à-dire à 10.

4.1.10.3 Commandes centrales et coefficient d'adéquation

Si on résout le programme linéaire (PL 3) on obtient $r^*(Q_v) = 2$. On en déduit la valeur du coefficient d'adéquation de Q_v à N_1 : $A(Q_v, N_1) = 50\%$. Cela signifie que la matrice Q_v assure un rayon de stabilité deux fois plus important pour ses commandes centrales que pour N_1 . La stabilité potentielle de Q_v n'est exploitée qu'à moitié si N_1 est la commande prévisionnelle. L'ensemble des commandes centrales est représenté en trait épais sur la Figure 16 : il s'agit des commandes équilibrées les plus éloignées des frontières de $B_1(Q_v)$. Les deux flèches représentent la distance d'une commande centrale aux deux plus proches contraintes de $B_1(Q_v)$. Par définition des commandes centrales, cette distance est supérieure ou égale à $r^*(Q_v)$, qui vaut 2.

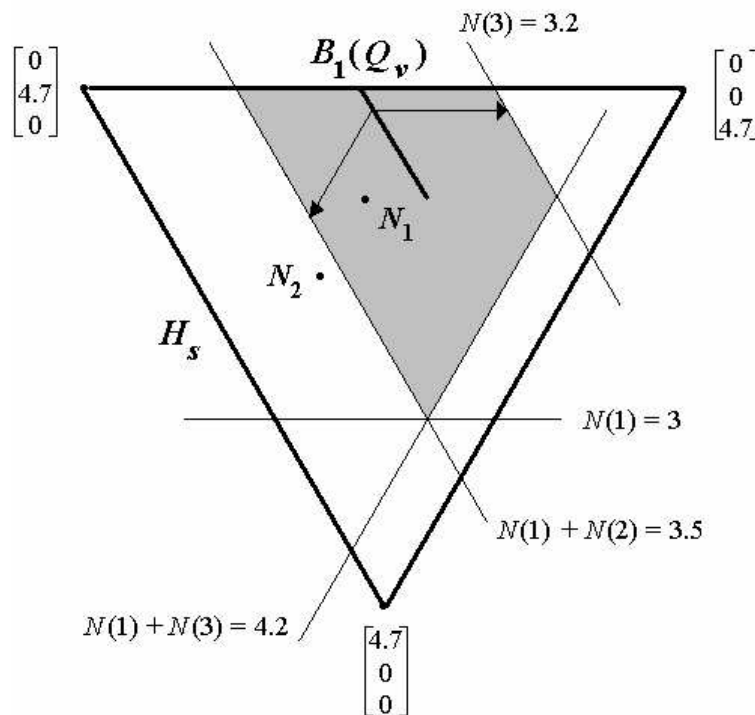


Figure 16 : Représentation des commandes centrales dans H_s

4.1.11 Synthèse d'une configuration stable du parc de machines

Dans les sections précédentes, des outils permettant de déterminer l'ensemble des commandes équilibrées pour une matrice de qualification et un vecteur de vitesse donnés ont été présentés. Ces outils ont permis d'obtenir une information locale : la valeur du rayon de stabilité, qui permet de quantifier le voisinage stable d'une commande prévisionnelle connue. Dans ce paragraphe, le problème inverse qui correspond au problème industriel, est étudié : il s'agit d'adapter la matrice de qualification Q à la commande prévisionnelle N_{ref} afin d'augmenter le rayon de stabilité. A partir d'une matrice de qualification Q donnée il s'agit de modifier les frontières de l'ensemble $B_1(Q_v)$ pour éloigner de la commande de référence des frontières qui limitent le rayon de stabilité. Cela est possible en particulier en réduisant les rectangles de zéros de Q c'est-à-dire en ajoutant des 1 à Q , lorsque cela est possible.

4.1.11.1 Extension de l'ensemble des commandes 1-équilibrées

L'ensemble $B_1(Q_v)$ des commandes 1-équilibrées est limité par des contraintes (C_k) associées aux rectangles maximaux de zéros de la matrice de qualification Q . L'objectif de ce paragraphe est d'étudier l'impact de la qualification d'une machine supplémentaire sur l'ensemble $B_1(Q_v)$, et sur la valeur du rayon de stabilité.

Soit $Q(i_0, j_0) = 0$ appartenant au rectangle maximal de zéros $I_k \times J_k$. Soit Q_+ la matrice obtenue à partir de Q en posant $Q_+(i_0, j_0) = 1$. Suite à l'ajout de ce 1, la contrainte (C_k) n'existe plus dans $B_1(Q_{v+})$, elle est remplacée par deux nouvelles contraintes, notées (C_k') et (C_k'') définies par les équations suivantes :

$$(C_k') : \sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k - \{j_0\}} V(j)$$

$$(C_k'') : \sum_{i \in I_k - \{i_0\}} N(i) \leq s - \sum_{j \in J_k} V(j)$$

Evaluons l'impact de l'ajout de ce 1 sur l'éloignement de N_{ref} aux nouvelles contraintes (C_k') et (C_k'') . Pour ce faire calculons l'écart entre N_{ref} et (C_k') et entre N_{ref} et (C_k'') :

$$\omega_k' = 2 \times \left(s - \sum_{j \in J_k - \{j_0\}} V(j) - \sum_{i \in I_k} N_{ref}(i) \right) = \omega_k + 2 \times V(j_0)$$

$$\omega_k'' = 2 \times \left(s - \sum_{j \in J_k} V(j) - \sum_{i \in I_k - \{i_0\}} N_{ref}(i) \right) = \omega_k + 2 \times N_{ref}(i_0)$$

Les équations précédentes montrent que N_{ref} est plus éloignée de chacune des deux nouvelles contraintes qu'elle ne l'était de (C_k) . Le rayon de stabilité est donc susceptible

d'augmenter. On appelle gain sur la contrainte (C_k) , noté $G(Q_v, N_{ref}, k, i_0, j_0)$ le minimum de l'accroissement de l'éloignement de N_{ref} aux deux contraintes (C_k') et (C_k'') :

$$G(Q_v, N_{ref}, k, i_0, j_0) = \min(\omega_k' - \omega_k, \omega_k'' - \omega_k)$$

Notons que l'ajout de ce 1 a pu avoir des conséquences semblables sur d'autres contraintes que (C_k) . Plus précisément, toutes les contraintes dont le rectangle maximal de zéros dans Q comprend le coefficient $Q(i_0, j_0)$ sont modifiées de la même manière que (C_k) .

Cas particulier : Quand l'un des ensembles I_k ou J_k est de cardinalité égale à 1, l'ajout d'un 1 se traduit par le remplacement d'une contrainte par une unique nouvelle contrainte, voire par aucune contrainte.

Propriété : Soit (C_k) une contrainte quelconque de $B_1(Q_v)$. Si $(i_0, j_0) \notin I_k \times J_k$, alors $G(Q_v, N_{ref}, k, i_0, j_0) = 0$. La réciproque est fausse.

Démonstration : L'implication énoncée est vraie par définition du gain. La contrainte (C_k) n'étant pas modifiée l'éloignement de N_{ref} à (C_k) est inchangé.

Montrons que la réciproque est fausse par contraposition. Soient $(i_0, j_0) \in I_k \times J_k$, et N_{ref} telle que $N_{ref}(i_0) = 0$. En portant cette valeur dans l'expression de ω_k'' , on trouve que $\omega_k'' = \omega_k$, donc le gain est nul. L'ajout d'un 1 dans le rectangle maximal de zéros associé à la contrainte (C_k) n'entraîne pas de gain parce que $N_{ref}(i_0) = 0$.

4.1.12 Outils généraux pour l'augmentation du rayon de stabilité

Pour augmenter le rayon de stabilité il faut ajouter au moins un 1 dans les rectangles maximaux de zéros associés aux contraintes qui imposent la valeur du rayon de stabilité. On appellera contraintes critiques ces contraintes. On note h_r le nombre de contraintes critiques ($1 \leq h_r \leq h$).

Propriété : L'ajout d'un 1 en position (i, j) dans la matrice de qualification entraîne une augmentation du rayon de stabilité pour cette commande si le couple (i, j) appartient à toutes les contraintes critiques et $N_{ref}(i) > 0$.

Démonstration : La démonstration de cette propriété est triviale à partir de la définition d'une contrainte critique et de l'expression de ω_k'' .

Remarque : Pour $N_{ref}(i) > 0$ s'il n'existe pas de couple $(i, j) \in I_k \times J_k$ pour tout k dans $[1, h_r]$ (il n'existe pas de 0 à l'intersection de toutes les contraintes critiques), alors il faudra ajouter plusieurs 1 dans Q pour augmenter le rayon de stabilité d'une valeur strictement positive.

4.1.12.1 Redondance et propriétés connexes

Toutes les contraintes issues de l'algorithme de détermination de $B_1(Q_v)$ sont indispensables, comme le montrera la démonstration du résultat général dont découle le théorème 1. Cependant la détermination de $B_1(Q_v)$ pouvant être un problème exponentiel, on souhaite éviter de calculer $B_1(Q_{v+})$ suite à l'ajout d'un 1 dans la matrice de qualification. L'étude menée au paragraphe précédent permet d'identifier les contraintes qui seront modifiées, c'est-à-dire remplacées par de nouvelles contraintes moins restrictives suite à l'ajout d'un 1. Mais les contraintes ainsi engendrées peuvent être superflues. Le présent paragraphe a pour objet de définir les contraintes redondantes, et de fournir un critère permettant de les mettre en évidence, afin de construire $B_1(Q_{v+})$ à partir de $B_1(Q_v)$.

Définition 15: On dit qu'une contrainte (C_k) issue du rectangle de zéros $I_k \times J_k$ de $B_1(Q_v)$ est redondante s'il existe une contrainte (C_k') issue du rectangle de zéros $I_k' \times J_k'$ telle que (C_k') est vérifiée implique que (C_k) l'est aussi.

La propriété suivante fournit un critère permettant de détecter les contraintes redondantes de $B_1(Q_v)$.

Propriété : une contrainte (C_k) issue du rectangle de zéros $I_k \times J_k$ de $B_1(Q_v)$ est redondante si et seulement si il existe une contrainte (C_k') issue du rectangle de zéros $I_k' \times J_k'$ de $B_1(Q_v)$ vérifiant :

$$\begin{cases} I_k \subset I_k' \\ J_k \subset J_k' \end{cases}$$

En d'autres termes, une contrainte issue d'un rectangle de zéros non maximal $I_k \times J_k$ est redondante à cause de la contrainte issue du rectangle maximal de zéros $I_k' \times J_k'$ qui contient $I_k \times J_k$.

Démonstration :

1) Montrons que $I_k \subset I_k'$ et $J_k \subset J_k'$ implique que (C_k) est redondante. Pour cela, on suppose que l'inégalité de la contrainte (C_k') est vraie :

$$(C_k') : \sum_{i \in I_k'} N(i) \leq s - \sum_{j \in J_k'} V(j)$$

Comme $I_k \subset I_k'$, on peut écrire :

$$\sum_{i \in I_k} N(i) + \sum_{\substack{i \in I_k' \\ i \notin I_k}} N(i) \leq s - \sum_{j \in J_k'} V(j)$$

Comme $J_k \subset J_k'$, on a $\sum_{j \in J_k'} V(j) \geq \sum_{j \in J_k} V(j)$, donc :

$$\sum_{i \in I_k} N(i) + \sum_{\substack{i \in I_k' \\ i \notin I_k}} N(i) \leq s - \sum_{j \in J_k'} V(j) \leq s - \sum_{j \in J_k} V(j)$$

$$\sum_{i \in I_k} N(i) + \sum_{\substack{i \in I_k' \\ i \notin I_k}} N(i) \leq s - \sum_{j \in J_k} V(j) \text{ puis finalement,}$$

$$\sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k} V(j)$$

La contrainte (C_k) est toujours vérifiée lorsque (C_k') est vraie, (C_k) est donc redondante.

2) Réciproquement, montrons que si la validité de la contrainte (C_k') implique la validité de la contrainte (C_k) , alors cela implique $I_k \subset I_k'$ et $J_k \subset J_k'$. Pour cela, on raisonne par contraposition, ce qui revient à démontrer la propriété suivante :

$I_k \not\subset I_k'$ ou $J_k \not\subset J_k'$ entraîne (C_k') vérifiée n'implique pas (C_k) vérifiée.

Supposons que $I_k \not\subset I_k'$, et montrons que (C_k') vraie n'implique pas (C_k) vraie. Un contre-exemple suffit à démontrer qu'une implication est fausse :

Soit les contraintes (C_1) et (C_2) définies par :

$$\begin{cases} N(i_1) \leq s - 4 & (C_1) \\ N(i_1) + N(i_2) \leq s - 5 & (C_2) \end{cases}$$

Si l'on pose $(C_k) = (C_2)$ et $(C_k') = (C_1)$, la condition $I_k \not\subset I_k'$ est vérifiée. Pour $V(j)=1$ pour tout j , $s = 10$, $N(i_1) = 6$ et $N(i_2) = 1$, la contrainte (C_k') est vraie ($6 \leq 10 - 4$), mais la contrainte (C_k) est fausse ($6 + 1 > 10 - 5$).

Supposons que $J_k \not\subset J_k'$, et montrons que (C_k') vraie n'implique pas (C_k) vraie. Un contre-exemple suffit à démontrer qu'une implication est fausse. Le contre-exemple du paragraphe précédent peut être utilisé car on a bien $J_k \not\subset J_k'$ puisque $\sum_{j \in J_k} V(j) > \sum_{j \in J_k'} V(j)$ et la validité de (C_k') n'entraîne pas celle de (C_k) .

On pourrait trouver deux autres contre-exemples pour lesquels $I_k \not\subset I_k'$ et $J_k \subset J_k'$, d'une part et $I_k \subset I_k'$ et $J_k \not\subset J_k'$ d'autre part pour lesquels (C_k') vérifiée n'implique pas (C_k)

vérifiée, ce qui terminerait la démonstration. De tels contre-exemples ne sont pas donnés pour des raisons de place.

Propriété : L'algorithme de détermination de $B_1(Q_v)$ tiré de [Nourine et Raynaud, 1999] ne retourne aucune contrainte redondante. Cette propriété vient du fait que les contraintes retournées sont toutes déduites de rectangles maximaux de zéros, par construction.

Propriété : Suite à l'ajout d'un 1 dans une matrice de qualification Q en (i_0, j_0) , toute contrainte (C_k) affectée par ce changement disparaît au profit de deux nouvelles contraintes (C_k') et (C_k'') définies plus haut, et éventuellement redondantes. Les trois étapes décrites ci-après permettent de supprimer les nouvelles contraintes redondantes.

- Etape 1 : Supprimer les contraintes telles que $\text{card}(I_k) = 0$ ou $\text{card}(J_k) = 0$
- Etape 2 : (C_k') est définie par $I_k \times (J_k - \{j_0\})$. S'il existe dans $B_1(Q_v)$ une contrainte (C_l) définie par $I_l \times (J_k - \{j_0\})$ avec $I_k \subset I_l$, alors (C_k') est redondante, et doit être supprimée
- Etape 3 : (C_k'') est définie par $(I_k - \{i_0\}) \times J_k$. S'il existe dans $B_1(Q_v)$ une contrainte (C_l) définie par $(I_k - \{i_0\}) \times J_l$ avec $J_k \subset J_l$, alors (C_k'') est redondante, et doit être supprimée

En résumé, les trois étapes de cette propriété visent à supprimer les nouvelles contraintes qui sont issues de rectangles de zéros non maximaux.

Démonstration : L'étape 1 supprime les nouvelles contraintes « inutiles » : $\text{card}(I_k) = 0$ conduit à des contraintes toujours vraies de la forme $0 \leq s - \sum_{j \in J_k} V(j)$. $\text{card}(J_k) = 0$ conduit à des contraintes toujours vraies de la forme $\sum_{i \in I_k} N(i) \leq s$ (car N appartient à H_s).

L'étape 2 consiste à vérifier qu'aucune contrainte (C_l) de $B_1(Q_v)$ ayant le même ensemble de machines que (C_k') ne peut rendre (C_k') redondante. Montrons qu'aucun autre type de contrainte ne peut rendre (C_k') redondante.

Il n'existe pas de contrainte (C_l) telle que $J_l = (J_k - \{j_0\}) \cup J_+$ avec J_+ non vide, $j_0 \notin J_+$ et $I_k \subset I_l$. En effet, s'il existait une telle contrainte, (C_k) serait redondante à cause de l'existence d'une contrainte définie par $I_l \times J_l$. C'est absurde, car (C_k) n'est pas redondante. Il en résulte que (C_k') ne peut être rendue redondante que par une contrainte (C_l) ayant le même ensemble de machines que (C_k') . Si une telle contrainte existe, elle vérifie nécessairement $I_k \subset I_l$. Si ce n'était pas le cas, la contrainte (C_l) serait redondante.

L'étape 3 consiste à vérifier qu'aucune contrainte (C_l) de $B_1(Q_v)$ ayant le même ensemble de produits que (C_k'') ne peut rendre (C_k'') redondante. Montrons qu'aucun autre type de contrainte ne peut rendre (C_k'') redondante.

Il n'existe pas de contrainte (C_l) telle que $I_l = (I_k - \{i_0\}) \cup I_+$ avec I_+ non vide, $i_0 \notin I_+$ et $J_k \subset J_l$. En effet, s'il existait une telle contrainte, (C_k) serait redondante à cause de l'existence d'une contrainte définie par $I_l \times J_l$. C'est absurde, car (C_k) n'est pas redondante. Il en résulte que (C_k'') ne peut être rendue redondante que par une contrainte (C_l) ayant le

même ensemble de produits que (C_k') . Si une telle contrainte existe, elle vérifie nécessairement $J_k \subset J_l$. Si ce n'était pas le cas, la contrainte (C_l) serait redondante.

Suite à ces étapes, aucune contrainte redondante n'est introduite dans la description de $B_1(Q_{v+})$.

Propriété : Si (C_1) et (C_2) sont deux contraintes critiques non redondantes telles que :

$$\begin{cases} I_1 \subset I_2 \\ J_2 \subset J_1 \end{cases}$$

alors il existe un couple $(i_0, j_0) \in I_1 \times J_2$ tel que poser $Q(i_0, j_0) = 1$ entraîne une augmentation strictement positive du gain de (C_1) et du gain de (C_2) .

Démonstration :

On sait que l'ajout d'un 1 à l'intersection de contraintes critiques peut entraîner un gain strictement positif pour chacune des contraintes à condition que $N_{ref}(i) > 0$. Supposons que l'on ait deux contraintes critiques non redondantes (C_1) et (C_2) telles que $I_1 \subset I_2$ et $J_2 \subset J_1$, et montrons par l'absurde que l'ensemble I_1 n'est pas tel que $N_{ref}(i) = 0 \quad (\forall i \in I_1)$.

$$(C_1) \text{ est critique, donc on a : } r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{i \in I_1} N_{ref}(i) \right).$$

On suppose que l'on a $N_{ref}(i) = 0 \quad (\forall i \in I_1)$, d'où :

$$r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_1} V(j) \right)$$

La contrainte (C_2) est également critique, d'où :

$$r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{i \in I_1} N_{ref}(i) - \sum_{\substack{i \in I_2 \\ i \notin I_1}} N_{ref}(i) \right) = 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{\substack{i \in I_2 \\ i \notin I_1}} N_{ref}(i) \right)$$

Comme $J_2 \subset J_1$, on a $\sum_{j \in J_2} V(j) < \sum_{j \in J_1} V(j)$. L'inégalité est stricte. En effet, s'il y avait égalité, la contrainte (C_1) serait redondante, car on aurait $I_1 \subset I_2$ et $J_1 \subset J_2$.

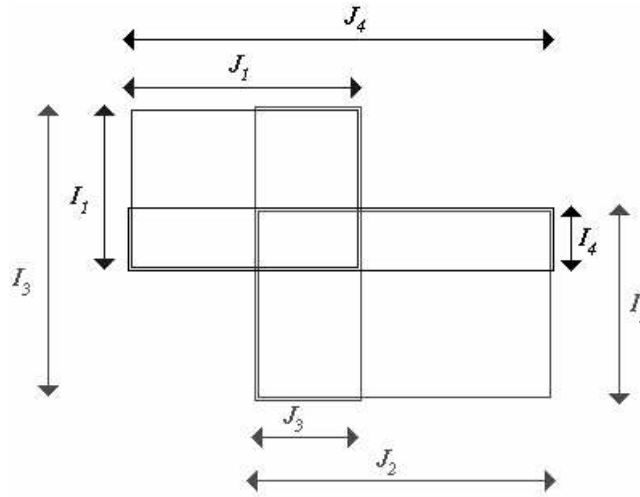
Majorons l'expression du rayon de stabilité obtenu à partir de (C_2) :

$$r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{\substack{i \in I_2 \\ i \notin I_1}} N_{ref}(i) \right) \leq 2 \times \left(s - \sum_{j \in J_2} V(j) \right) < 2 \times \left(s - \sum_{j \in J_1} V(j) \right) = r(Q_v, N_{ref})$$

L'inéquation $r(Q_v, N_{ref}) < r(Q_v, N_{ref})$ est absurde, donc la propriété est vraie.

Propriété : Si les rectangles maximaux de zéros de Q associés à deux contraintes critiques non redondantes (C_1) et (C_2) ont une intersection non vide, alors il existe un couple (i_0, j_0) dans cette intersection, tel que poser $Q(i_0, j_0) = 1$ entraîne un gain strictement positif sur chacune de ces contraintes.

Démonstration :



Pour démontrer cette propriété, il suffit de prouver que les contraintes (C_3) et (C_4) définies ci-dessous sont nécessairement critiques si (C_1) et (C_2) le sont. On pourra alors utiliser la propriété précédente pour montrer qu'il existe une ligne i_0 dans I_4 telle que $N_{ref}(i) > 0$, ce qui garantit une augmentation du rayon de stabilité lorsqu'on pose $Q(i_0, j_0) = 1$, où $j_0 \in J_1 \cap J_2$.

La contrainte (C_3) est associée au rectangle de zéros $I_3 \times J_3$ défini par $I_3 = I_1 \cup I_2$ et $J_3 = J_1 \cap J_2$.

La contrainte (C_4) est associée au rectangle de zéros $I_4 \times J_4$ défini par $I_4 = I_1 \cap I_2$ et $J_4 = J_1 \cup J_2$.

On note d_3 (respectivement d_4) la distance de (C_3) (respectivement de (C_4)) à N_{ref} .

$$d_3 = 2 \times \left(s - \sum_{j \in J_3} V(j) - \sum_{i \in I_3} N_{ref}(i) \right)$$

$$d_4 = 2 \times \left(s - \sum_{j \in J_4} V(j) - \sum_{i \in I_4} N_{ref}(i) \right)$$

Par définition du rayon de stabilité, on a : $r(Q_v, N_{ref}) \leq d_3$. Comme (C_1) est critique, on peut écrire $r(Q_v, N_{ref})$ à l'aide de l'expression de (C_1) .

$$r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{i \in I_1} N_{ref}(i) \right)$$

$$d_3 = 2 \times \left(s - \sum_{j \in J_3} V(j) - \sum_{i \in I_1} N_{ref}(i) - \sum_{i \in I_2} N_{ref}(i) + \sum_{i \in I_4} N_{ref}(i) \right) \geq 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{i \in I_1} N_{ref}(i) \right)$$

$d_3 \geq r(Q_v, N_{ref})$ revient à écrire :

$$\sum_{j \in J_1} V(j) \geq \sum_{j \in J_3} V(j) + \sum_{i \in I_2} N_{ref}(i) - \sum_{i \in I_4} N_{ref}(i), \text{ soit encore :}$$

$$\sum_{j \in J_3} V(j) \leq \sum_{j \in J_1} V(j) - \sum_{i \in I_2} N_{ref}(i) + \sum_{i \in I_4} N_{ref}(i)$$

On peut alors réécrire d_4 et la majorer à l'aide de l'expression de $\sum_{j \in J_3} V(j)$:

$$d_4 = 2 \times \left(s - \sum_{j \in J_4} V(j) - \sum_{i \in I_4} N_{ref}(i) \right) = 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{j \in J_2} V(j) + \sum_{j \in J_3} V(j) - \sum_{i \in I_4} N_{ref}(i) \right)$$

$$d_4 \leq 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{j \in J_2} V(j) + \left[\sum_{j \in J_1} V(j) - \sum_{i \in I_2} N_{ref}(i) + \sum_{i \in I_4} N_{ref}(i) \right] - \sum_{i \in I_4} N_{ref}(i) \right)$$

$$d_4 \leq 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{i \in I_2} N_{ref}(i) \right)$$

Or la contrainte (C_2) est critique, donc : $r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{i \in I_2} N_{ref}(i) \right)$.

C'est-à-dire que $d_4 \leq r(Q_v, N_{ref})$.

Par définition du rayon de stabilité, $d_4 \leq r(Q_v, N_{ref})$ est impossible, donc on a nécessairement $d_4 = r(Q_v, N_{ref})$. On a démontré que la contrainte (C_4) était critique.

Montrons que (C_3) est critique. Comme (C_4) est critique, on peut écrire :

$$d_4 = 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{j \in J_2} V(j) + \sum_{j \in J_3} V(j) - \sum_{i \in I_4} N_{ref}(i) \right) = r(Q_v, N_{ref}) = 2 \times \left(s - \sum_{j \in J_2} V(j) - \sum_{i \in I_2} N_{ref}(i) \right)$$

d'où :

$$\sum_{j \in J_3} V(j) = \sum_{j \in J_1} V(j) - \sum_{i \in I_2} N_{ref}(i) + \sum_{i \in I_4} N_{ref}(i)$$

On peut remplacer l'expression de $\sum_{j \in J_3} V(j)$ dans l'expression de d_3 donnée ci-dessous :

$$d_3 = 2 \times \left(s - \sum_{j \in J_3} V(j) - \sum_{i \in I_1} N_{ref}(i) - \sum_{i \in I_2} N_{ref}(i) + \sum_{i \in I_4} N_{ref}(i) \right)$$

$$d_3 = 2 \times \left(s - \sum_{j \in J_1} V(j) - \sum_{i \in I_1} N_{ref}(i) \right) = r(Q_v, N_{ref})$$

On retrouve l'expression de $r(Q_v, N_{ref})$ traduisant que (C_1) est critique, donc (C_3) est critique.

On a démontré que si les contraintes (C_1) et (C_2) étaient critiques, alors (C_3) et (C_4) l'étaient aussi. On peut appliquer la propriété précédente aux contraintes (C_1) et (C_4) .

On a bien $I_4 \subset I_1$ et $J_1 \subset J_4$, car $I_4 = I_1 \cap I_2$ et $J_4 = J_1 \cup J_2$. On peut alors affirmer qu'il existe une ligne $i_0 \in I_4$ telle que $N_{ref}(i_0) > 0$. Comme $I_4 = I_1 \cap I_2$, alors poser $Q(i_0, j_0) = 1$ avec $j_0 \in J_1 \cap J_2$ permet d'obtenir un gain strictement positif sur les quatre contraintes critiques, et donc en particulier sur les contraintes (C_1) et (C_2) .

La propriété est démontrée.

4.1.13 Augmentation du rayon de stabilité à l'aide d'un seul 1

Une manière triviale d'augmenter le rayon de stabilité consiste à remplir la matrice de qualification de 1. Ceci conduit à un rayon de stabilité infini, quels que soient N_{ref} , N et v , parce que toute commande est alors équilibrée. Cependant, cette solution n'est pas admissible compte tenu des contraintes techniques et économiques. On a vu que les contraintes technologiques ne permettaient pas de qualifier certaines machines pour certains types de produits. Par ailleurs, la qualification d'une machine nécessite des vérifications et des réglages périodiques pour s'assurer que cette qualification est effective. Ces opérations de maintenance entraînent un coût. On fait l'hypothèse que le coût de maintenance du parc de machines est proportionnel au nombre de qualifications, c'est-à-dire au nombre de 1 de la matrice de qualification. Dans ce paragraphe, on suppose que l'on ne peut ajouter qu'un seul 1 à Q . Une méthode de placement de ce 1 conduisant à maximiser le rayon de stabilité de Q_{v+} est présentée. La démarche sera illustrée par deux exemples. Le premier est purement académique, le second correspond à une situation industrielle réelle.

4.1.13.1 Démarche

S'il existe plus d'une contrainte critique, et que l'intersection des rectangles de zéros associés à ces contraintes est vide, alors l'ajout d'un seul 1 ne peut suffire à augmenter le rayon de stabilité. En effet, quel que soit le placement du 1 à ajouter à Q , il existe toujours au moins une contrainte critique qui n'est pas modifiée. Il en résulte que la valeur du rayon de stabilité est nécessairement inchangée.

S'il n'existe qu'une seule contrainte critique, alors il est clair que l'ajout d'un 1 peut permettre l'augmentation du rayon de stabilité, si ce 1 est ajouté dans le rectangle de zéros associé à cette contrainte. Mais cette augmentation peut varier en fonction de la position du 1 à ajouter dans le rectangle de zéros.

En effet, ajouter un 1 sur une ligne i telle que $N(i)$ est nul n'entraîne aucune augmentation du rayon de stabilité. D'autre part, une fois que la contrainte critique aura été remplacée suite à l'ajout du 1, c'est une autre contrainte qui deviendra critique pour $B_1(Q_{v+})$. Par conséquent, le placement du 1 à ajouter à Q doit :

- Se situer dans le rectangle de zéros de la contrainte critique
- Ne pas être situé sur une ligne i telle que $N(i) = 0$

- Se situer à l'intersection du plus grand nombre de rectangles de zéros associés à des contraintes dont la distance à N est proche du rayon de stabilité.

On rappelle que la proximité d'une contrainte (C_k) à N est mesurée par ω_k .

Enfin, rappelons que l'atelier de photolithographie est soumis à des contraintes complexes qui ne sont intégrées au modèle proposé ici. Il peut donc arriver que l'ajout du « 1 » maximisant l'augmentation du rayon de stabilité ne soit pas possible à cause de ces contraintes. Dans ce cas, il faudra proposer d'ajouter un « 1 » à l'intersection des contraintes critiques, sur une ligne i telle que $N(i)$ est non nul, pour permettre l'augmentation stricte du rayon de stabilité. Par conséquent, la méthode présentée peut être appliquée directement à la configuration robuste d'un parc de machines uniformes ou identiques, si l'on souhaite garantir un taux d'occupation des machines égal à 100 %.

4.1.13.2 Exemple académique

Considérons la matrice Q_v et la commande N_1 et $B_1(Q_v)$:

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad N_1 = \begin{bmatrix} 1 \\ 2 \\ 1.7 \end{bmatrix} \quad B_1(Q_v) : \begin{cases} N(1) \leq 3 \\ N(1) + N(2) \leq 3.5 \\ N(3) \leq 3.2 \\ N(1) + N(3) \leq 4.2 \end{cases} \quad \Omega(Q_v, N_1) = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 3 \end{bmatrix}$$

On peut constater qu'il n'y a qu'une seule contrainte critique car le vecteur $\Omega(Q_v, N_1)$ admet un minimum unique égal à 1. Ce minimum est atteint pour la contrainte (C_2) : $N(1) + N(2) \leq 3.5$. Il suffit donc d'ajouter un seul 1 à Q pour provoquer une augmentation du rayon de stabilité. Ce 1 doit être ajouté dans le rectangle maximal de zéros de Q associé à la contrainte critique, c'est-à-dire au rectangle $I_2 \times J_2$ avec $I_2 = \{1, 2\}$ et $J_2 = \{4\}$. Deux zéros peuvent être remplacés par un 1. On peut voir également sur le vecteur $\Omega(Q_v, N_1)$ que les contraintes qui sont immédiatement plus proches de N_1 après (C_2) sont les contraintes (C_3) et (C_4). Les rectangles de zéros associés à ces contraintes ont une intersection vide avec $I_2 \times J_2$, donc l'augmentation du rayon de stabilité sera la même quel que soit le zéro du rectangle $I_2 \times J_2$ que l'on remplace par un 1.

Si l'on choisit par exemple de poser $Q_+(1, 4) = 1$, on obtient la matrice Q_+ , l'ensemble des commandes équilibrés et le vecteur d'écart à N_1 suivants :

$$Q_+ = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad B_1(Q_{v+}) : \begin{cases} N(1) + N(3) \leq 4.2 \\ N(2) \leq 3.5 \\ N(3) \leq 3.2 \end{cases} \quad \text{d'où } \Omega(Q_{v+}, N_1) = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$$

Ceci conduit à un rayon de stabilité égal à $r(Q_{v+}, N_1) = 3$. On aurait obtenu la même valeur si on avait posé $Q_+(2,4) = 1$. En revanche, ajouter un 1 hors de $I_2 \times J_2$ ne permet pas d'augmenter le rayon de stabilité.

4.1.13.3 Exemple industriel

On dispose d'une matrice de qualification Q et la commande prévisionnelle N_{ref} associées à un atelier réel de photolithographie :

$Q =$	0	0	0	1	0	1	0	0	0	0	0	0	0	6
	0	0	0	0	0	0	0	0	0	1	1	1	1	392
	0	0	0	0	0	1	1	0	1	0	0	0	0	158
	0	0	0	0	0	1	1	0	1	0	0	0	0	328
	0	0	0	0	0	0	0	0	0	1	0	0	0	6
	0	1	0	0	1	0	1	0	0	0	0	0	0	274
	1	1	1	1	0	1	0	1	1	1	0	1	0	1333
	1	0	0	1	0	1	0	1	1	1	0	1	0	1324
	0	0	0	1	1	1	0	0	0	0	0	0	0	283
	0	0	0	0	0	0	0	0	0	1	1	1	0	109
	0	0	0	0	0	0	0	0	0	1	0	1	0	19
	0	0	0	1	1	1	0	0	0	0	0	0	0	132
	0	0	0	0	1	1	1	0	1	0	0	0	0	982
	0	0	0	0	1	1	1	0	1	0	0	0	0	133
	0	0	0	0	1	1	1	0	1	0	0	0	0	349
	1	1	1	0	0	0	0	0	0	0	0	0	0	854
	1	1	1	0	0	0	0	0	0	0	0	0	0	374
	0	0	0	1	1	0	1	0	0	0	0	0	0	69
	0	0	0	1	1	0	1	0	0	0	0	0	0	170
	0	1	1	0	0	0	0	0	0	0	0	0	0	437
	0	0	0	1	1	1	1	1	1	0	0	0	0	297
	1	0	0	0	0	0	0	0	0	1	1	1	0	92
	0	0	0	0	0	0	0	0	0	1	1	1	1	412
	0	0	0	0	0	0	0	0	0	1	1	1	1	61
	0	0	0	0	0	0	0	0	0	1	0	1	0	30
	1	0	0	0	0	0	0	0	0	1	1	1	0	68
	0	0	0	0	0	0	0	0	0	1	0	1	0	28
	0	0	0	0	0	0	0	0	0	1	0	1	0	16
	0	0	0	0	0	0	0	0	0	1	1	0	0	6
	0	0	0	0	0	0	0	0	0	0	1	1	0	39
	0	0	0	0	0	0	0	0	0	1	1	1	0	59
	0	0	0	0	0	0	0	0	1	0	0	0	1	40
0	0	0	0	0	0	0	0	1	0	0	0	1	23	
0	0	0	0	0	0	0	0	1	0	0	0	1	70	
0	0	0	0	0	0	0	0	1	0	0	0	1	57	
0	0	0	0	0	0	0	0	1	0	0	0	0	15	

Cet atelier comprend 13 machines et 36 types de produits, et la matrice Q compte 113 coefficients non nuls. Aucune matrice de vitesse n'est mentionnée car toutes les machines ont des vitesses supposées unitaires. La détermination des contraintes de $B_1(Q)$ nécessite moins de 3 secondes de calcul sur un PC Pentium IV à 2 GHz. Ceci montre que la méthode de détermination de $B_1(Q_v)$, bien qu'exponentielle, est utilisable pour traiter un problème réel. On dénombre $h = 722$ contraintes, dont une seule est critique. On note N_{ref} la projection de N_{ref} dans H_s , avec $s = m$. Le rayon de stabilité vaut 0.1048. Il est intéressant de constater que la contrainte la plus proche de N_{ref} après la contrainte critique est à une distance de 1.0326. Ceci indique que le rayon de stabilité pourra être augmenté au moins d'un facteur 10 si on ajoute un 1 dans le rectangle de zéros de la contrainte critique. Les trois contraintes les plus proches de N_{ref} (dont la contrainte critique) ont une intersection non vide, réduite aux points dont les coordonnées sont données sous la forme (ligne,colonne) : (16,13) (17,13) et (20,13). Les quatre contraintes les plus proches de N_{ref} ont une intersection vide. L'ajout d'un seul 1 permet donc de faire « reculer » les trois contraintes les plus proches de N_{ref} . Cette intersection concerne la machine 13, il reste à choisir le type de produit pour lequel cette machine doit être qualifiée, parmi les types de produit 16, 17 et 20. Pour cela, on écrit la valeur de ω_k pour les trois contraintes les plus proches de N_{ref} : $\omega_1 = 0.1048$ $\omega_2 = 1.0326$ $\omega_3 = 1.2140$ (on a trié le vecteur $\Omega(Q_v, N_{ref})$ par ordre croissant, de sorte que $\omega_1 = r(Q_v, N_{ref})$). On écrit également la valeur de la commande N_{ref} pour les produits de type 16, 17 et 20 : $N_{ref}(16) = 1.2274$, $N_{ref}(17) = 0.5375$ et $N_{ref}(20) = 0.6281$. D'après les résultats obtenus dans cette partie, le gain sur une contrainte est nécessairement inférieur au minimum de $V(j_0)$ (ici 1) et $N_{ref}(i_0)$. C'est donc en qualifiant la machine 13 pour le type de produits 16 que le gain sera maximum. La valeur du nouveau rayon de stabilité est donc fixée par la quatrième contrainte la plus proche de N_{ref} : $r(Q_{v+}, N_{ref}) = 1.5586$. Si on avait choisi le type de produits 17 ou 20, le gain aurait été limité par $N_{ref}(i)$, et on aurait obtenu un nouveau rayon de stabilité valant respectivement 1.1798 et 1.3610. Poser $Q_+(16,13) = 1$ conduit à un rayon de stabilité égal à 1.5586. L'ajout de ce 1 a permis d'augmenter la valeur du rayon de stabilité d'un facteur 14.8, ce qui est considérable.

Par définition, il en est de même pour la tolérance. Pour la commande N_{ref} fournie (qui n'est pas dans H_s), la tolérance initiale valait $t(Q, N_{ref}) = 0.806 \%$ ce qui correspond à 72.9 produits. On peut donc garantir l'équilibre de toute commande $N = N_{ref} + dN$ à condition que N reste dans H_s et que la norme 1 de dN soit inférieure ou égale à 72.9. Après l'ajout du 1 dans Q , la tolérance devient : $t(Q_+, N_{ref}) = 12 \%$. Ainsi, on peut garantir l'équilibre de N dans H_s pour une perturbation dN impliquant jusqu'à 1085 produits.

4.2 Les machines sont identiques

On rappelle que lorsque les machines sont identiques, la matrice de vitesse v est telle que pour tout coefficient $v(i,j)$ non nul, on a $v(i,j) = V(i)$ pour tout j . Cela signifie que outre les contraintes technologiques qui imposent les coefficients nuls de v , un produit de type i est traité à la même vitesse quelle que soit la machine à laquelle il est affecté (à condition bien sûr que la machine soit qualifiée pour ce type de produit). Cette vitesse unique est notée $V(i)$. Par conséquent, une même machine ne traitera pas tous les produits qui lui sont affectés à la même vitesse, puisque cette vitesse change d'un type de produits à l'autre.

La matrice de vitesse suivante entre dans le cadre de l'étude du présent paragraphe :

$$v = \begin{bmatrix} 1.3 & 1.3 & 1.3 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 1.5 & 1.5 & 1.5 & 1.5 \end{bmatrix} \quad V = \begin{bmatrix} 1.3 \\ 0.2 \\ 1.5 \end{bmatrix}$$

Le vecteur V représente la vitesse sur chaque type de produits. Rappelons qu'à l'instar des machines uniformes, on a démontré qu'aucune boucle ne peut être trouvée dans une telle matrice.

Dans ce paragraphe, on suppose que le vecteur dN est tel que :

$$\begin{cases} N = N_{ref} + dN \\ \sum_{i=1}^n \frac{N(i)}{V(i)} = \sum_{i=1}^n \frac{N_{ref}(i)}{V(i)} \end{cases}$$

La quantité $\frac{N(i)}{V(i)}$ représente le temps nécessaire à une machine pour traiter tous les produits de type i . Par conséquent, on notera T et on appellera temps de traitement brut de la commande N , la quantité suivante :

$$T = \sum_{i=1}^n \frac{N(i)}{V(i)}$$

T est le temps qui serait nécessaire au traitement de N s'il n'y avait qu'une seule machine (qualifiée pour tous les types de produits) dans l'atelier. Contrairement au cas de la partie précédente, on ne fait pas l'hypothèse que le volume de la commande N est constant : on suppose ici que le temps de traitement brut T (indépendamment des contraintes de qualification) est constant.

Comme $N(i) = N_{ref}(i) + dN(i)$ pour tout i , on peut déduire que le vecteur dN vérifie :

$$\sum_{i=1}^n \frac{dN(i)}{V(i)} = 0 .$$

On va maintenant chercher à déterminer l'ensemble des commandes N simplement équilibrées dont le temps de traitement brut T est égal à celui de N_{ref} , ce qui permettra de déduire la plus petite valeur de la norme de dN qui provoque la perte de la propriété

d'équilibre. La valeur de cette norme constituera une mesure de robustesse de la configuration Q . Dans les paragraphes suivants, on continuera à parler d'équilibre sans préciser s'il s'agit de l'équilibre simple ou strict car ces deux notions sont toujours équivalentes en l'absence de boucle dans ν (et donc aussi dans Q_ν). En pratique, on n'étudiera que l'équilibre simple.

4.2.1 Etude de l'espace des commandes

A priori, l'espace des commandes à étudier est R_+^n , c'est-à-dire l'ensemble des vecteurs de dimension n , dont aucun élément n'est strictement négatif. On a vu que la propriété d'équilibre n'était pas modifiée par multiplication de la commande. On peut donc étudier l'ensemble des commandes N dans un sous-ensemble de R_+^n défini par :

$$\sum_{i=1}^n \frac{N(i)}{V(i)} = \text{constante}.$$

Définition 16: On note H_m l'ensemble des vecteurs N de dimension n défini par :

$$\begin{cases} N(i) \geq 0 & (\forall i \in I) \\ \|N\|_1 = \sum_{j=1}^m V(j) = s \end{cases}$$

La notation $\|X\|_T$ désigne ce que l'on appellera « la norme T » du vecteur X . Cette norme est définie par :

$$\|X\|_T = \sum_i \left| \frac{X(i)}{V(i)} \right| \quad (V(i) > 0 \forall i)$$

Il s'agit bien d'une norme : on peut s'assurer très facilement que l'opérateur $\|\cdot\|_T$ vérifie les axiomes de non dégénérescence, l'inégalité triangulaire, et $\|\lambda \times X\|_T = |\lambda| \times \|X\|_T$.

Remarque : H_m est l'ensemble des commandes (admissibles) dont la norme est égale à m . On peut donc en déduire que toute commande équilibrée de H_m conduira à une valeur de C_{max} unitaire. Une commande non équilibrée de H_m se traduira par un C_{max} strictement supérieur à un.

4.2.2 Formule de projection sur H_m et ensemble des commandes 1-équilibrées

Soit N une commande non nulle quelconque. Sa projection N' sur H_m est définie par :

$$N' = \frac{N}{\|N\|_T} \times m$$

Sauf mention contraire, on ne considérera que des commandes admissibles de norme m , c'est-à-dire des éléments de H_m , car toute commande non nulle peut être ramenée dans H_m par projection.

L'ensemble des commandes 1-équilibrées, noté $B_1(Q_v)$, est convexe. La démonstration est analogue à celle qui a été faite dans le cas des machines uniformes.

4.2.4 Caractérisation de l'ensemble des commandes 1-équilibrées

La première étape de la caractérisation des commandes 1-équilibrées d'une matrice Q_v donnée est identique au cas où les machines sont uniformes : la recherche des rectangles de zéros de Q ne dépend pas de v .

En revanche, l'étape 2, où une contrainte est associée à chaque rectangle de zéros est différente. On montrera plus loin que le théorème qui est à l'origine de cette étape est le même que celui qui a été utilisé dans la partie précédente.

Définition 17: Soit $I_k \times J_k$ un rectangle maximal de zéros de Q . On définit la contrainte notée (C_k) par :

$$(C_k): \sum_{i \in I_k} \frac{N(i)}{V(i)} \leq m - \text{card}(J_k)$$

Théorème 3: Soit N un élément de H_m . N appartient à $B_1(Q_v)$ si et seulement si N satisfait les contraintes (C_k) , pour tout entier k dans $[1, h]$.

A titre d'exemple, pour la matrice Q_v définie ci-dessous, on a :

$$Q_v = \begin{bmatrix} 1.3 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 1.5 & 1.5 & 0 & 1.5 \end{bmatrix} \quad B_1(Q_v) : \begin{cases} \frac{N(1)}{1.3} \leq 1 & I_1 = \{1\} & J_1 = \{2,3,4\} \\ \frac{N(1)}{1.3} + \frac{N(2)}{0.2} \leq 3 & I_2 = \{1,2\} & J_2 = \{4\} \\ \frac{N(1)}{1.3} + \frac{N(3)}{1.5} \leq 3 & I_3 = \{1,3\} & J_3 = \{3\} \end{cases}$$

4.2.5 Comparaison de H_s et de H_m

La manière de procéder étant tout à fait similaire, aucun exemple numérique ne sera donné. Concernant la représentation graphique de H_m , on peut noter une différence avec H_s . H_s est une portion d'hyperplan orthogonal au vecteur dont tous les coefficients valent 1. Il suffit pour s'en convaincre de rappeler l'équation de H_s :

$$H_s : \sum_{i=1}^n N(i) = s$$

De même, H_m est orthogonal à vecteur W défini par $W(i) = \frac{1}{V(i)}$ pour tout i :

$$H_m : \sum_{i=1}^n \frac{N(i)}{V(i)} = m$$

Il en résulte que H_m et H_s ne sont parallèles que lorsque toutes les machines ont la même vitesse pour tous les types de produits, c'est-à-dire lorsque les machines sont à la fois identiques et uniformes.

4.2.6 Distance dans H_m

Dans ce paragraphe, on présente les distances d'un point de H_m à un autre, et d'un point à une contrainte de type (C_k) . Les démonstrations ne sont pas données parce qu'elles sont analogues à celles du paragraphe 4.1.7 concernant les machines uniformes et les distances dans H_s .

4.2.6.1 Distance séparant deux commandes de H_m

Soit A et B deux éléments de H_m . La distance de A à B est définie par :

$$d(A, B) = \|B - A\|_T$$

4.2.6.2 Distance d'une commande à une frontière

Soit N une commande de H_m . On note (C_k) une contrainte de H_m définie par l'inéquation suivante :

$$(C_k) : \sum_{i \in I_k} \frac{N(i)}{V(i)} \leq m - \text{card}(J_k)$$

Si (F_k) est la frontière de (C_k) , les éléments de (F_k) vérifient :

$$(F_k) : \sum_{i \in I_k} \frac{N(i)}{V(i)} = m - \text{card}(J_k)$$

Où I_k est un ensemble d'entiers inclus dans I . La distance de tout point N (avec $N \in H_m$) à la frontière de (C_k) s'écrit :

$$d(N, C_k) = 2 \times \left| m - \text{card}(J_k) - \sum_{i \in I_k} \frac{N(i)}{V(i)} \right|$$

4.2.7 Le rayon de stabilité et son augmentation

Le rayon de stabilité se calcule comme dans le cas des machines uniformes, mais en utilisant les distances dans H_m . Il en est de même pour la tolérance et la définition des contraintes critiques et des commandes centrales.

$$r(Q_v, N_{ref}) = \min_k (d(N_{ref}, C_k))$$

On peut garantir qu'une commande N de H_m définie par : $N = N_{ref} + dN$ est équilibrée si :

$$\|dN\|_T \leq r(Q_v, N_{ref})$$

Par exemple, pour Q_v définie au paragraphe 4.2.4, et pour $N_{ref} = \begin{bmatrix} 0.65 \\ 0.3 \\ 3 \end{bmatrix}$, le vecteur des

distances est : $\Omega(Q_v, N_{ref}) = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$

D'où le rayon de stabilité : $r(Q_v, N_{ref}) = 1$.

Le problème de l'augmentation du rayon de stabilité en ajoutant un seul 1 se traite de la même manière que pour les machines uniformes. Il s'agit de trouver les contraintes critiques, et de sélectionner le coefficient nul de Q situé à l'intersection du plus grand nombre de contraintes critiques pour maximiser l'augmentation du rayon de stabilité.

4.3 Conclusion

La notion de rayon de stabilité a permis d'établir une mesure de robustesse, qu'une méthode permet d'augmenter en modifiant la configuration courante Q en lui ajoutant des « 1 ». Paradoxalement, l'absence de prise en compte des contraintes spécifiques à la photolithographie constitue un avantage de cette méthode : les rectangles de zéros associés aux contraintes critiques regroupent les coefficients nuls de Q candidats à l'ajout d'un « 1 » susceptibles de conduire à une augmentation de la robustesse de la configuration courante. Le décideur peut utiliser cette méthode pour évaluer l'impact de ses propres choix sur la robustesse de la configuration courante, ou pour identifier les coefficients nuls susceptibles de conduire à une augmentation du rayon de stabilité s'ils sont remplacés par un « 1 ». Les degrés de liberté restant (le choix du coefficient nul à remplacer effectivement) peuvent être très utiles si une contrainte non modélisée (propre au fonctionnement et à la technologie de l'atelier) interdit de choisir le coefficient conduisant à la plus forte augmentation du rayon de stabilité. Si les contraintes spécifiques à l'atelier de photolithographie ne permettent pas d'ajouter un « 1 » dans les rectangles de zéros associés aux contraintes critiques, c'est que l'augmentation de la robustesse par ajout d'un seul « 1 » est impossible. Il convient alors d'ajouter plusieurs « 1 ».

Chapitre V : Le volume et la répartition des produits de la commande sont incertains

On considère que le cas le plus général du Multi-Purpose Machines : les vitesses des machines dépendent des types de produits et des machines. La matrice de vitesses v peut admettre des boucles. On considère également que la commande N peut être perturbée de n'importe quelle manière (en volume et/ou en répartition). La seule limitation est l'admissibilité, c'est-à-dire que pour tous les types de produits i , on a :

$$dN(i) \leq N(i) \quad (\forall i \in I)$$

Ceci garantit l'admissibilité de la commande N définie par $N = N_{ref} + dN$.

5.1 Etude de l'espace des commandes

Contrairement aux cas précédents, il n'est plus possible de ramener l'étude de l'équilibre d'une commande dans un hyperplan particulier comme H_s ou H_m . En effet, l'existence de ces hyperplans traduisait la conservation de la charge (pour les machines uniformes) ou du temps de traitement brut (pour les machines identiques). Or dans le présent chapitre, les perturbations sont telles qu'aucune quantité particulière relative à la commande N ne peut être conservée (sauf bien sûr la positivité). Ainsi, on ne cherche plus à déterminer $B_1(Q_v)$, mais $B(Q_v)$, c'est-à-dire l'ensemble des commandes équilibrées, quel que soit le C_{max} .

5.2 Caractérisation des commandes simplement équilibrées

Dans cette partie, on énonce un théorème général qui est à la base des théorèmes 1 et 3 présentés dans le chapitre précédent, mais qui n'ont pas été démontrés. Il s'agit des théorèmes qui permettent d'associer une contrainte à tout rectangle maximal de zéro de Q , pour les machines uniformes (théorème 1) et pour les machines identiques (théorème 3).

Le produit cartésien $I_k \times J_k$ désigne un rectangle maximum de zéros de Q , et les ensembles I'_k et J'_k sont les compléments de I_k dans I et de J_k dans J . Rappelons que I'_k et J'_k sont nécessairement non vides : si I_k (respectivement J_k) était vide, alors il existerait une ligne (respectivement une colonne) de zéros dans la matrice Q , et cette dernière ne serait alors pas admissible.

Soit I_k un ensemble de types de produits et J_k un ensemble de machines. On note $T(I_k, J_k)$ le plus petit temps moyen de traitement de tous les produits de type I_k par les machines de l'ensemble J_k . La quantité de produits dans chaque type est donnée par la commande N . Les vitesses des machines sur les produits sont données par la matrice Q_v .

Soient la matrice Q_v et la commande N définie ci-dessous :

$$Q_v = \begin{bmatrix} 0 & 0 & 0 & 1.7 \\ 0 & 0 & 0 & 3.2 \\ 1 & 0.2 & 3 & 2.1 \\ 2.2 & 0.9 & 1.2 & 2.9 \end{bmatrix} \quad N = \begin{bmatrix} 53 \\ 62 \\ 23 \\ 84 \end{bmatrix}$$

On a un seul rectangle maximal de zéros : $I_1 = \{1, 2\}$ et $J_1 = \{1, 2, 3\}$. Ecrivons $T(I_1, J_1)$. On obtient ce temps en résolvant (PL 1) pour la matrice $Q_v(I_1, J_1)$ et la commande $N(I_1)$. $T(I_1, J_1)$ est égal à la valeur du C_{max} de (PL 1) : $T(I_1, J_1) = 21.6744$.

Théorème 4 : Soit N un élément de \mathfrak{R}_+^n . N appartient à $B(Q_v)$ si et seulement si N satisfait toutes les contraintes (C_k) , pour tout entier k dans $[1, h]$.

La contrainte (C_k) est associée au rectangle maximal de zéros $I_k \times J_k$, et elle est définie par :

$$T(I_k, J'_k) \leq T(I'_k, J_k)$$

La Figure 17 ci-dessous représente le rectangle de zéros $I_k \times J_k$ dans la matrice Q_v . Les temps $T(I_k, J'_k)$ et $T(I'_k, J_k)$ apparaissent dans les rectangles types de produits / machines auxquels ils sont associés.

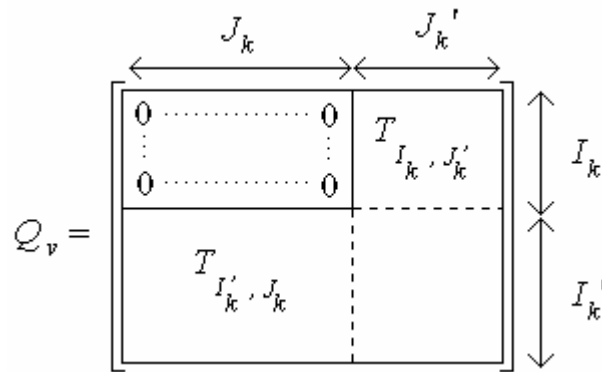


Figure 17 : découpage de la matrice Q_v

La contrainte (C_k) de $B(Q_v)$ est respectée si la commande N est telle que le traitement des produits de type I_k par les machines J_k dure au moins aussi longtemps que le traitement des produits I_k par les seules machines de J_k .

5.2.2 Démonstration du théorème 4

Démontrons l'équivalence suivante, pour tout élément N de \mathfrak{R}_+^n :

N appartient à $B(Q_v) \Leftrightarrow N$ satisfait toutes les contraintes (C_k)

5.2.2.1 Montrons que $(N \text{ appartient à } B(Q_v)) \Rightarrow (N \text{ vérifie toutes les contraintes } (C_k))$

Par contraposition, il est équivalent de démontrer l'implication suivante :

N ne satisfait pas une contrainte $(C_k) \Rightarrow N$ n'est pas équilibrée pour Q_v .

Soit (C_k) une contrainte issue d'un rectangle maximal de zéros de Q . Elle s'écrit : $T(I_k, J'_k) \leq T(I'_k, J_k)$. Supposons que la commande N ne satisfait pas (C_k) . On a alors : $T(I_k, J'_k) > T(I'_k, J_k)$. Cela signifie que la charge moyenne des machines de J'_k sur les seuls types de produits I_k est strictement supérieure à la charge moyenne des machines de J_k . Il en résulte que le couple (Q_v, N) est nécessairement non équilibré, parce que les machines de J'_k seront toujours strictement plus chargées que celles de J_k .

5.2.2.2 Montrons que $(N \text{ satisfait toutes les contraintes } (C_k)) \Rightarrow (N \text{ appartient à } B(Q_v))$

Par contraposition, il est équivalent de démontrer l'implication suivante :

N n'est pas équilibrée \Rightarrow il existe une contrainte (C_k) non vérifiée par N

Pour démontrer l'implication ci-dessus, on va utiliser un résultat exposé ci-après.

Résultat intermédiaire : Quelle que soit la commande N non équilibrée pour Q_v , il est toujours possible de construire une solution optimale à (PL 1) « à nombre minimal de machines critiques »

Soit N non équilibrée pour Q_v , et soit R_T une matrice de répartition temporelle solution optimale de (PL 1). En général, (PL 1) admet plusieurs solutions optimales différentes. On va transformer la solution optimale R_T en une solution optimale R_T' particulière, dite « à nombre minimal de machines critiques ».

Définition 18: Soit R_T une solution optimale à (PL 1) pour une commande N non équilibrée. L'ensemble des machines critiques de R_T , noté $J_C(R_T)$ est défini par :

$$J_C(R_T) = \left\{ j / \sum_{i \in I} R_T(i, j) = C_{\max} \right\}$$

Comme N n'est pas simplement équilibrée, $J_C(R_T)$ est un ensemble dont le cardinal est compris entre 1 et $m-1$.

Définition 19: On dit qu'une solution optimale R_T' de (PL 1) est « à nombre minimal de machines critiques » si le cardinal de $J_C(R_T')$ est minimum.

Propriété : toute solution optimale de (PL 1) peut être transformée en une solution optimale « à nombre minimal de machines critiques ». La transformation est décrite ci-après.

Soit $j \in J_C(R_T)$. S'il existe un type de produit i_0 tel que :

$$\begin{cases} R_T(i_0, j) > 0 \\ (\exists j_0 \notin J_C(R_T) \text{ t.q. } Q(i_0, j_0) = 1) \end{cases}$$

Alors R_T n'est pas « à nombre minimal de machines critiques », et on peut la modifier de sorte que la charge de la machine j soit strictement inférieure à C_{\max} de la façon suivante :

Soit $C(j_0)$ la charge de la machine j_0 : $C(j_0) = \sum_{i=1}^n R_T(i, j_0) < C_{\max}$. On a $C_{\max} - C(j_0) > 0$

On déplace une quantité notée $e(i_0, j)$ de produit de type i_0 initialement affectée à la machine j par la solution R_T , vers la machine j_0 . Ce déplacement rendra la machine j non critique à condition que sa charge diminue, c'est à dire $e(i_0, j) > 0$. De plus, on ne peut déplacer une quantité de produits $e(i_0, j)$ supérieure à la quantité affectée par R_T . Cela se traduit par :

$$e(i_0, j) \leq R_T(i_0, j) \times Q_v(i_0, j)$$

Enfin, la quantité déplacée $e(i_0, j)$ ne doit pas rendre critique la machine j_0 , ou pire, faire perdre l'optimalité au sens de (PL 1) en ajoutant une charge telle que $C(j_0) > C_{\max}$. D'où :

$$e(i_0, j) < (C_{\max} - C(j_0)) \times Q_v(i_0, j_0)$$

Finalement, il existe nécessairement une quantité de produit $e(i_0, j)$ vérifiant :

$$0 < e(i_0, j) < \min((C_{\max} - C(j_0)) \times Q_v(i_0, j_0), R_T(i_0, j) \times Q_v(i_0, j))$$

De sorte que déplacer $e(i_0, j)$ sur la machine j_0 provoque la sortie de j de l'ensemble $J_C(R_T)$ sans perdre l'optimalité de la solution à (PL 1), et sans que j ne devienne une machine critique.

On peut ainsi définir une nouvelle solution R_T' en posant :

$$\begin{cases} 0 < e(i_0, j) < \min((C_{\max} - C(j_0)) \times Q_v(i_0, j_0), R_T(i_0, j) \times Q_v(i_0, j)) \\ R_T'(i_0, j) = R_T(i_0, j) - e(i_0, j) \times Q_v(i_0, j) \\ R_T'(i_0, j_0) = R_T(i_0, j_0) + e(i_0, j) \times Q_v(i_0, j_0) \\ R_T'(i_1, j_1) = R_T(i_1, j_1) \quad (\forall (i_1, j_1) \neq (i_0, j)) (\forall (i_1, j_1) \neq (i_0, j_0)) \end{cases}$$

Quand, pour tout j dans $J_C(R_T')$ on ne peut trouver aucun produit i_0 tel que $\begin{cases} R_T'(i_0, j) > 0 \\ (\exists j_0 \notin J_C(R_T') \text{ t.q. } Q(i_0, j_0) = 1) \end{cases}$, alors la solution R_T' est dite « à nombre minimal de machines critiques ».

Application à la démonstration de « N satisfait toutes les contraintes $(C_k) \Rightarrow N$ appartient à $B(Q_v)$ »

On suppose que N n'est pas équilibrée pour Q_v . On va faire l'hypothèse que la solution optimale R_T au problème de répartition de N sur Q_v est « à nombre minimal de machines critiques ».

Cela signifie que si l'on considère une machine j dans $J_C(R_T)$, alors tout produit i tel que $R_T(i, j) > 0$ ne peut être traité que par des machines de l'ensemble $J_C(R_T)$. Il existe donc un ensemble non vide de types de produits noté I_k tel que $(\forall i \in I_k)(Q(i, j) = 1 \Rightarrow j \in J_C(R_T))$. La charge temporelle des machines de $J_C(R_T)$ n'est due qu'aux produits de l'ensemble I_k . Par définition de $J_C(R_T)$, la charge d'une machine critique est exactement égale à C_{\max} . On peut donc exprimer la charge moyenne des machines critiques par :

$$T(I_k, J_C(R_T)) = C_{\max}$$

Or, le fait que les produits de l'ensemble I_k ne puissent être traités que par les machines de l'ensemble $J_C(R_T)$ implique l'existence d'un rectangle de zéros (maximal par définition) aux lignes $i \in I_k$, et aux colonnes j telles que $j \notin J_C(R_T)$ dans la matrice Q . La présence de ce rectangle maximal de zéros se traduit par l'existence d'une contrainte (C_k) , définie par :

$$T(I_k, J'_k) \leq T(I'_k, J_k)$$

Avec $J'_k = J_C(R_T)$.

Or si la contrainte (C_k) était vérifiée, on aurait :

$$C_{\max} \leq T(I'_k, J_k)$$

Ce qui est absurde. En effet, R_T n'est pas une matrice de répartition équilibrée, donc on a nécessairement $T(I_k, J'_k) \neq T(I'_k, J_k)$. La vérification de la contrainte (C_k) conduit donc à une inégalité stricte remettant en cause l'optimalité de R_T pour (PL 1) :

$$C_{\max} < T(I'_k, J_k)$$

La commande non équilibrée N ne vérifie donc pas une contrainte (C_k) .

5.2.3 Démonstration de cas particuliers du théorème 4

Les temps $T(I_k, J_k)$ qui apparaissent dans la formulation du théorème 4 ne s'expriment pas aisément en fonction des vitesses et de la commande N en général. C'est pourquoi l'utilisation pratique de ce théorème n'a été réservée qu'aux cas des machines uniformes et au cas des machines identiques. Dans les paragraphes suivants, on utilise le théorème 4 pour démontrer les théorèmes 1 et 3 utilisés plus haut pour les machines uniformes et identiques, où l'expression de $T(I_k, J_k)$ en fonction de Q_v et de N est aisée.

5.2.3.1 Les machines sont uniformes et les perturbations conservent la charge

On rappelle que lorsque les machines sont uniformes, $V(j)$ désigne la vitesse de la machine j indépendamment des types de produits. On note s la capacité de production du parc de machines :

$$s = \sum_{j=1}^m V(j)$$

Le fait que les perturbations conservent la charge se traduit par la considération de l'ensemble de commandes noté H_s , dans lequel on cherche les commandes équilibrées dont le C_{\max} vaut 1 (c'est $B_1(Q_v)$).

Une commande admissible N appartient à H_s si et seulement si :

$$\sum_{i=1}^n N(i) = s$$

Considérons un rectangle maximal de zéros $I_k \times J_k$ et exprimons $T(I_k, J'_k)$ et $T(I'_k, J_k)$ en fonction de N et de V :

$$T(I_k, J'_k) = \frac{\sum_{i \in I_k} N(i)}{\sum_{j \in J'_k} V(j)} \quad T(I'_k, J_k) = \frac{\sum_{i \in I'_k} N(i)}{\sum_{j \in J_k} V(j)}$$

A l'aide des définitions de s et de H_s , on peut écrire les égalités suivantes :

$$\sum_{i \in I'_k} N(i) = s - \sum_{i \in I_k} N(i) \quad \sum_{j \in J'_k} V(j) = s - \sum_{j \in J_k} V(j)$$

Dans le cas général, la contrainte (C_k) s'écrit : $T(I_k, J'_k) \leq T(I'_k, J_k)$. Ici, on a :

$$\frac{\sum_{i \in I_k} N(i)}{s - \sum_{j \in J_k} V(j)} \leq \frac{s - \sum_{i \in I_k} N(i)}{\sum_{j \in J_k} V(j)} \Leftrightarrow \sum_{i \in I_k} N(i) \times \sum_{j \in J_k} V(j) \leq \left(s - \sum_{i \in I_k} N(i) \right) \times \left(s - \sum_{j \in J_k} V(j) \right)$$

$$0 \leq s^2 - s \left(\sum_{i \in I_k} N(i) + \sum_{j \in J_k} V(j) \right) \Leftrightarrow \sum_{i \in I_k} N(i) \leq s - \sum_{j \in J_k} V(j)$$

On retrouve bien l'expression de (C_k) qui a été donnée au paragraphe 4.1

5.2.3.2 Les machines sont identiques et les perturbations conservent la temps de traitement brut

On rappelle que lorsque les vitesses sont identiques, $V(i)$ désigne la vitesse de traitement d'une unité de produit de type i indépendamment de la machine chargée de ce traitement. On rappelle que m désigne le nombre de machines du parc.

Le fait que les perturbations conservent le temps de traitement brut se traduit par la considération de l'ensemble de commandes noté H_m , dans lequel on cherche les commandes équilibrées dont le C_{max} vaut 1 (c'est $B_1(Q_v)$).

Une commande admissible N appartient à H_m si et seulement si :

$$\sum_{i=1}^n \frac{N(i)}{V(i)} = m$$

Considérons un rectangle maximal de zéros $I_k \times J_k$ et exprimons $T(I_k, J'_k)$ et $T(I'_k, J_k)$ en fonction de N et de V :

$$T(I_k, J'_k) = \frac{\sum_{i \in I_k} \frac{N(i)}{V(i)}}{\text{card}(J'_k)} \quad T(I'_k, J_k) = \frac{\sum_{i \in I'_k} \frac{N(i)}{V(i)}}{\text{card}(J_k)}$$

Par définition de J_k et de J'_k et à l'aide de la définition de H_m , on peut écrire les égalités suivantes :

$$\text{card}(J'_k) = m - \text{card}(J_k) \quad \sum_{i \in I'_k} \frac{N(i)}{V(i)} = m - \sum_{i \in I_k} \frac{N(i)}{V(i)}$$

Dans le cas général, la contrainte (C_k) s'écrit : $T(I_k, J'_k) \leq T(I'_k, J_k)$. Ici, on a :

$$\frac{\sum_{i \in I_k} \frac{N(i)}{V(i)}}{m - \text{card}(J_k)} \leq \frac{m - \sum_{i \in I_k} \frac{N(i)}{V(i)}}{\text{card}(J_k)} \Leftrightarrow \sum_{i \in I_k} \frac{N(i)}{V(i)} \times \text{card}(J_k) \leq \left(m - \sum_{i \in I_k} \frac{N(i)}{V(i)} \right) \times (m - \text{card}(J_k))$$

$$0 \leq m^2 - m \left(\sum_{i \in I_k} \frac{N(i)}{V(i)} + \text{card}(J_k) \right) \Leftrightarrow \sum_{i \in I_k} \frac{N(i)}{V(i)} \leq m - \text{card}(J_k)$$

On retrouve bien l'expression de (C_k) qui a été donnée au paragraphe 4.2.

5.3 Exemples

5.3.1 Les machines sont uniformes

On fait l'hypothèse que la commande peut être perturbée de façon quelconque (dans la limite de son admissibilité), et que les machines sont uniformes. On reprend un exemple déjà traité dans le cas où les perturbations conservaient la charge totale de la commande.

$$v = \begin{bmatrix} 1 & 2 & 0.5 & 1.2 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0.5 & 1.2 \\ 1 & 2 & 0.5 & 1.2 \end{bmatrix} \quad Q = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Les rectangles de zéros sont les mêmes, mais on construit maintenant $B(Q_v)$, et plus seulement $B_1(Q_v)$. La contrainte (C_k) s'écrit :

$$T(I_k, J'_k) \leq T(I'_k, J_k)$$

Avec les expressions de $T(I_k, J'_k)$ et de $T(I'_k, J_k)$:

$$T(I_k, J'_k) = \frac{\sum_{i \in I_k} N(i)}{\sum_{j \in J'_k} V(j)} \quad T(I'_k, J_k) = \frac{\sum_{i \in I'_k} N(i)}{\sum_{j \in J_k} V(j)}$$

Finalement, la contrainte (C_k) s'écrit dans ce cas :

$$(C_k) : \sum_{i \in I_k} N(i) \leq \frac{s - \sum_{j \in J_k} V(j)}{\sum_{j \in J_k} V(j)} \times \sum_{i \in I'_k} N(i)$$

On en déduit l'expression de $B(Q_v)$:

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad B(Q_v) : \begin{cases} N(1) \leq \frac{3}{1.7} (N(2) + N(3)) & I_1 = \{1\} \quad J_1 = \{3,4\} \\ N(1) + N(2) \leq \frac{3.5}{1.2} N(3) & I_2 = \{1,2\} \quad J_2 = \{4\} \\ N(3) \leq \frac{3.2}{1.5} (N(1) + N(2)) & I_3 = \{3\} \quad J_3 = \{1,3\} \\ N(1) + N(3) \leq \frac{4.2}{0.5} N(2) & I_4 = \{1,3\} \quad J_4 = \{3\} \end{cases}$$

$B(Q_v)$ et $B_1(Q_v)$ comptent tous les deux h contraintes parce que leurs contraintes sont issues des mêmes rectangles maximaux de zéros de Q . En revanche, les frontières des contraintes de $B_1(Q_v)$ sont de dimension 1 (parce qu'elles sont incluses dans H_s qui est de dimension 2) alors que celles de $B(Q_v)$ sont de dimension 2 (ce sont des hyperplans d'un espace à trois dimensions). La figure ci-dessous est une représentation dans l'espace de H_s , $B_1(Q_v)$ et de la frontière d'une contrainte de $B(Q_v)$, pour l'exemple étudié :

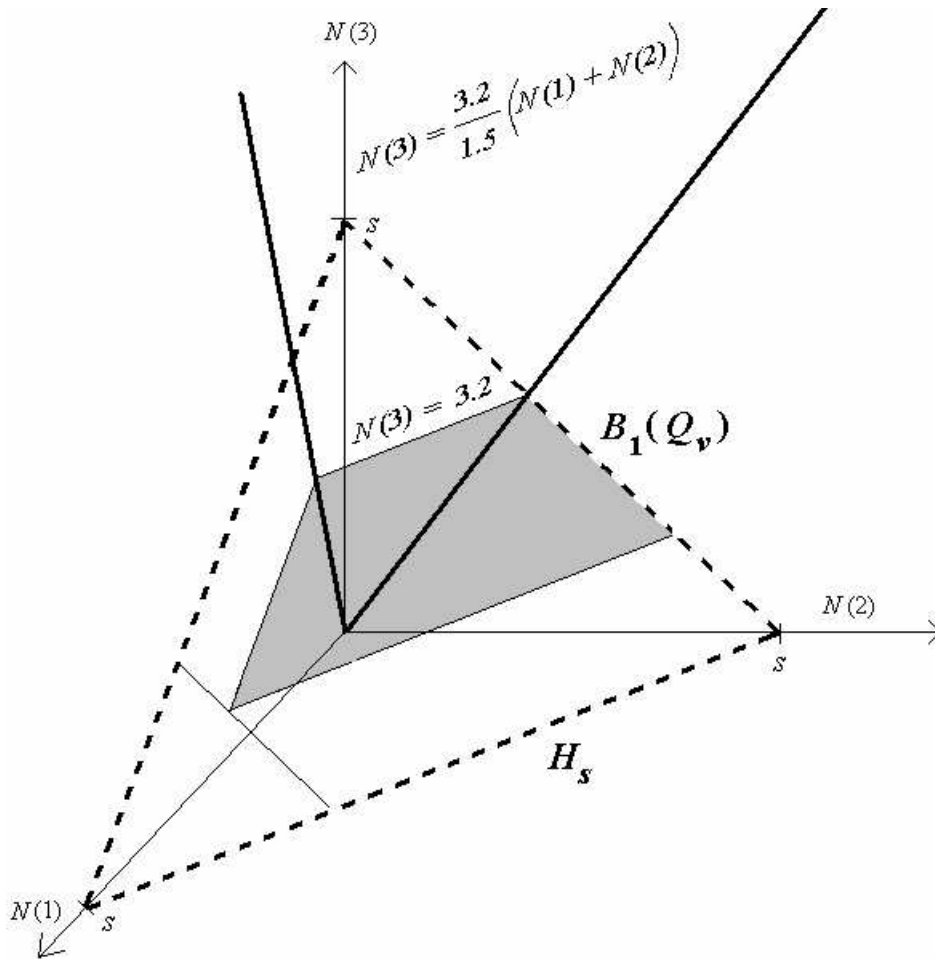


Figure 18 : Représentation spatiale de H_s , $B_1(Q_v)$ et d'une contrainte de $B(Q_v)$

Seule une frontière de $B(Q_v)$ a été représentée, afin de ne pas surcharger le dessin. La Figure 18 est une représentation en trois dimensions de l'espace des commandes. H_s est de dimension 2 : c'est un triangle représenté en traits pointillés. L'ensemble des commandes équilibrées dont le C_{max} est égal à 1 est inclus dans H_s : la zone grisée représente $B_1(Q_v)$, qui est délimitée par 4 frontières issues de 4 contraintes. L'équation de l'une d'elles est portée sur le dessin :

$N(3) \leq 3.2$ est associée au rectangle maximum de zéros $I_3 \times J_3$. Sa frontière est un segment de droite inclus dans H_s : $N(3) = 3.2$ Cette frontière est de dimension 1.

La contrainte de $B(Q_v)$ associée à ce même rectangle est :

$$N(3) \leq \frac{3.2}{1.5}(N(1) + N(2))$$

La frontière de cette contrainte est une portion de plan passant par l'origine, et dont on a représenté en traits épais l'intersection avec le plan $N(1) = 0$ d'une part, et l'intersection avec le plan $N(2) = 0$ d'autre part. Cette frontière est de dimension 2. On peut remarquer que toute frontière (C_k) de $B_1(Q_v)$ est incluse dans la frontière (C_k) de $B(Q_v)$. De même, $B_1(Q_v)$, qui est une portion de plan, est inclus dans $B(Q_v)$ qui est une portion d'espace de dimension 3 qui n'est pas entièrement représentée sur la Figure 18, mais qui est un cône dont le sommet est l'origine, et dont la base est un trapèze (c'est $B_1(Q_v)$).

5.3.2 Les machines sont identiques

On fait l'hypothèse que la commande peut être perturbée de façon quelconque (dans la limite de son admissibilité), et que les machines sont identiques. On reprend un exemple déjà traité dans le cas où les perturbations conservaient le temps de traitement brut de la commande :

$$v = \begin{bmatrix} 1.3 & 1.3 & 1.3 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 1.5 & 1.5 & 1.5 & 1.5 \end{bmatrix} \quad V = \begin{bmatrix} 1.3 \\ 0.2 \\ 1.5 \end{bmatrix} \quad \text{et} \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

Les rectangles de zéros ne dépendent que de Q qui n'a pas changé, mais on construit maintenant $B(Q_v)$, et plus seulement $B_1(Q_v)$. La contrainte (C_k) s'écrit en général :

$$T(I_k, J'_k) \leq T(I'_k, J_k)$$

Avec les expressions de $T(I_k, J_k)$ et de $T(I'_k, J_k)$:

$$T(I_k, J'_k) = \frac{\sum_{i \in I_k} \frac{N(i)}{V(i)}}{\text{card}(J'_k)} \quad T(I'_k, J_k) = \frac{\sum_{i \in I'_k} \frac{N(i)}{V(i)}}{\text{card}(J_k)}$$

Finalement, la contrainte (C_k) s'écrit dans ce cas :

$$(C_k): \sum_{i \in I_k} \frac{N(i)}{V(i)} \leq \left(\frac{m - \text{card}(J_k)}{\text{card}(J_k)} \right) \times \sum_{i \in I'_k} \frac{N(i)}{V(i)}$$

On en déduit l'expression de $B(Q_v)$:

$$Q_v = \begin{bmatrix} 1.3 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 1.5 & 1.5 & 0 & 1.5 \end{bmatrix} \quad B(Q_v): \begin{cases} \frac{N(1)}{1.3} \leq \frac{1}{3} \times \left(\frac{N(2)}{0.2} + \frac{N(3)}{1.5} \right) & I_1 = \{1\} \quad J_1 = \{2, 3, 4\} \\ \frac{N(1)}{1.3} + \frac{N(2)}{0.2} \leq \frac{3}{1} \times \frac{N(3)}{1.5} & I_2 = \{1, 2\} \quad J_2 = \{4\} \\ \frac{N(1)}{1.3} + \frac{N(3)}{1.5} \leq \frac{3}{1} \times \frac{N(2)}{0.2} & I_3 = \{1, 3\} \quad J_3 = \{3\} \end{cases}$$

Comme dans le chapitre précédent, il est nécessaire de définir des distances pour mesurer le rayon de stabilité, et donc la robustesse. Le paragraphe suivant est consacré à ces mesures qui se font cette fois dans \mathfrak{R}_+^n .

5.4 Distances dans \mathfrak{R}_+^n

La définition de la distance entre deux commandes et entre une commande et une frontière est différente dans \mathfrak{R}_+^n de ce qu'elle est dans H_s ou H_m , parce que l'on considère l'ensemble des commandes admissibles. D'autre part, les frontières des contraintes de $B(Q_v)$ dans \mathfrak{R}_+^n sont des hyperplans passant par l'origine, ce qui n'était pas le cas des frontières de $B_1(Q_v)$ dans H_s ou H_m . La distance d'une commande à une telle frontière sera présentée dans le cas général, c'est-à-dire aussi bien pour les machines uniformes que pour les machines identiques.

5.4.1 Distance séparant deux commandes de \mathfrak{R}_+^n

La distance de deux éléments quelconques A et B de \mathfrak{R}_+^n s'exprime comme dans H_s :

$$d(A, B) = \|B - A\|_1$$

Il ne sera plus nécessaire, lorsque les machines sont identiques, de considérer une distance en norme T . La distance utilisée est la distance « standard » associée à la norme 1.

5.4.2 Distance d'une commande à une frontière dans \mathfrak{R}_+^n

On a vu que pour les machines uniformes comme pour les machines identiques, les contraintes (C_k) étaient de la forme :

$$(C_k): \sum_{i \in I_k} u(i) \times N(i) \leq \sum_{i \in I'_k} u(i) \times N(i)$$

Où les $u(i)$ sont des coefficients réels strictement positifs. Notons aussi que ni I_k , ni I'_k ne peuvent être des ensembles vides, par définition d'un rectangle maximum de zéros. Il en résulte que la frontière de (C_k) , notée (F_k) s'écrit :

$$(F_k): \sum_{i \in I_k} u(i) \times N(i) - \sum_{i \in I'_k} u(i) \times N(i) = 0$$

(F_k) est donc un hyperplan de \mathfrak{R}_+^n passant par l'origine, dont l'équation peut aussi s'écrire :

$$(F_k): \sum_{i \in I} w(i) \times N(i) = 0$$

Où les $w(i)$ sont non nuls pour tout i , et où il existe au moins un $w(i)$ strictement positif et un $w(i)$ strictement négatif. Ceci garantit que l'intersection de (F_k) avec \mathfrak{R}_+^n n'est pas réduite à l'origine.

5.4.2.1 Distance d'une commande à une frontière dans \mathfrak{R}^n

Définition 20: la distance d'une commande N à la frontière (F_k) dans \mathfrak{R}^n s'écrit :

$$d(N, F_k) = \left| \frac{\sum_{i=1}^n w(i) \times N(i)}{w(i_0)} \right|$$

Où l'indice i_0 est défini par : $w(i_0) = \max_i (w(i))$

Démonstration : il est clair que si N appartient à (F_k) , sa distance à (F_k) est nulle. Supposons que N ne soit pas un élément de (F_k) . On a donc :

$$\sum_{i=1}^n (w(i) \times N(i)) = y \neq 0 \text{ soit encore } \sum_{i \neq i_0} (w(i) \times N(i)) + w(i_0) \times N(i_0) = y \neq 0$$

Où i_0 est défini par la formule $w(i_0) = \max_{i \in I} (w(i))$. On définit le point P appartenant à \mathfrak{R}^n par :

$$\begin{cases} P(i) = N(i) & (\forall i \neq i_0) \\ P(i_0) = \frac{-1}{w(i_0)} \times \left(\sum_{i \neq i_0} (w(i) \times N(i)) \right) = \frac{y - w(i_0) \times N(i_0)}{-w(i_0)} = N(i_0) - \frac{y}{w(i_0)} \end{cases}$$

On peut aisément vérifier que P appartient à l'hyperplan (F_k) . Calculons maintenant la distance (en norme 1) de N à P :

$$d(N, P) = \|N - P\|_1 = |N(i_0) - P(i_0)| = \left| \frac{y}{w(i_0)} \right|$$

Démontrons par l'absurde que cette distance est minimale. Pour cela, on fait l'hypothèse qu'il existe un point X appartenant à (F_k) tel que la distance de N à X est strictement inférieure à la distance de N à P . Cela se traduit par :

$$\|N - X\|_1 < \|N - P\|_1 \text{ donc } \sum_{i=1}^n |N(i) - X(i)| < |N(i_0) - P(i_0)|$$

Cela signifie qu'il existe une suite de réels $\{x_i\}_i$ avec i dans I telle que :

$$\begin{cases} X(i) = N(i) + x_i & (\forall i \in I) \\ \sum_{i=1}^n |x_i| < |N(i_0) - P(i_0)| \\ \sum_{i=1}^n [w(i) \times (N(i) + x_i)] = 0 \end{cases}$$

La première ligne est l'expression de X en fonction de N , la seconde ligne traduit l'hypothèse que X est à une distance de N inférieure à la distance qui sépare N de P , et la troisième ligne rappelle que X est un élément de (F_k) . Développons la troisième ligne :

$$\sum_{i=1}^n (w(i) \times N(i)) = - \sum_{i=1}^n (w(i) \times x_i) \text{ soit encore } |y| = \left| \sum_{i=1}^n (w(i) \times x_i) \right|$$

Par définition de $w(i_0)$ on peut écrire :

$$|y| \leq |w(i_0)| \times \sum_{i=1}^n |x_i| \text{ sachant que } \sum_{i=1}^n |x_i| < |N(i_0) - P(i_0)|, \text{ on peut majorer strictement } |y| :$$

$$|y| < |w(i_0)| \times d(N, P) \text{ et comme } d(N, P) = \left| \frac{y}{w(i_0)} \right|, \text{ on obtient } |y| < |y|, \text{ ce qui est absurde.}$$

On a donc démontré la formule exprimant la distance d'un point N de \mathfrak{R}^n à un hyperplan (F_k) .

5.4.2.2 Distance d'une commande à une frontière dans \mathfrak{R}_+^n

Examinons le cas particulier suivant. On se place dans un espace de commandes à $n = 3$ dimensions, et on considère la commande N_e et l'hyperplan frontière (F_k) définis par :

$$N_e = \begin{bmatrix} 3 \\ 2 \\ 0.5 \end{bmatrix} \quad (F_k): \quad N(1) - N(2) + 5 \times N(3) = 0$$

(F_k) est bien un hyperplan de \mathfrak{R}^n passant par zéro, ses coefficients $w(i)$ sont tous non nuls (ils valent 1, -1 et 5), il en existe un strictement négatif et un strictement positif, donc l'intersection de (F_k) avec \mathfrak{R}_+^n n'est pas réduite à l'origine. La commande N_e est admissible : elle appartient bien à \mathfrak{R}_+^n .

Si l'on calcule la distance de N_e à (F_k) dans \mathfrak{R}^n , on trouve 0.7. En effet, le point P_0 appartient à (F_k) et sa distance à N_e vaut 0.7 :

$$P_0 = \begin{bmatrix} 3 \\ 2 \\ -0.2 \end{bmatrix}$$

Cependant, on remarque que $P_0(3)$ est strictement négatif, donc P_0 n'est pas une commande admissible. La distance d'une commande N à une commande non admissible n'a pas de sens pour notre problème : aucune perturbation dN ne peut faire perdre l'admissibilité de la commande réelle N définie par $N = N_{ref} + dN$. Il est donc nécessaire de calculer la distance d'une commande (admissible) à une frontière dans \mathfrak{R}_+^n , et non dans \mathfrak{R}^n . On propose pour cela une méthode consistant à construire le point P appartenant à (F_k) le plus proche de N (où N est une commande admissible). On construit le point P par itération. On initialise la méthode en posant $P = N$, et on change un coefficient $P(i)$ à chaque itération de manière à rapprocher le point P de l'hyperplan (F_k) tout en restant dans \mathfrak{R}_+^n . On s'arrête dès que P appartient à (F_k) . Cette méthode est décrite formellement par l'algorithme ci-dessous :

Poser $P = N$

$$y = \sum_{i=1}^n (w(i) \times P(i))$$

Classer les coefficients $w(i)$ par valeur absolue décroissante

$i = 1$

Tant que y est non nul, faire

$$P(i) = \max \left(0, N(i) - \frac{y}{w(i)} \right)$$

$$y = \sum_{i=1}^n (w(i) \times P(i))$$

$$i = i + 1$$

Fin tant que

$$d(N, F_k) = \|N - P\|_1$$

Le coefficient $P(i)$ est modifié de sorte que P appartienne à (F_k) dans la boucle « Tant que ». En interdisant les valeurs strictement négatives, on est assuré que P reste admissible.

Illustrons cet algorithme avec la commande N_e et le plan (F_k) définis ci-dessus. Après avoir posé $P = N_e$, on trouve $y = 3.5$. La première itération fixe $P(3) = \max(0, -0.2) = 0$, ce qui conduit à la nouvelle valeur de y : $y = 1$. Pour la seconde itération, on fixe $P(2) = \max(0, 3) = 3$, ce qui conduit alors à $y = 0$ et à la sortie de la boucle. Le point P est alors défini par :

$$P = \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} \quad d(N_e, P) = \|N_e - P\|_1 = 0 + 1 + .5 = 1.5$$

Le point P appartient bien à (F_k) et à \mathfrak{R}_+^n . Finalement, la distance de la commande N_e à la frontière (F_k) vaut 0.2 dans \mathfrak{R}^n , et 1.5 dans \mathfrak{R}_+^n . La Figure 19 représente cette situation dans le plan de coupe $N(1) = 3$:

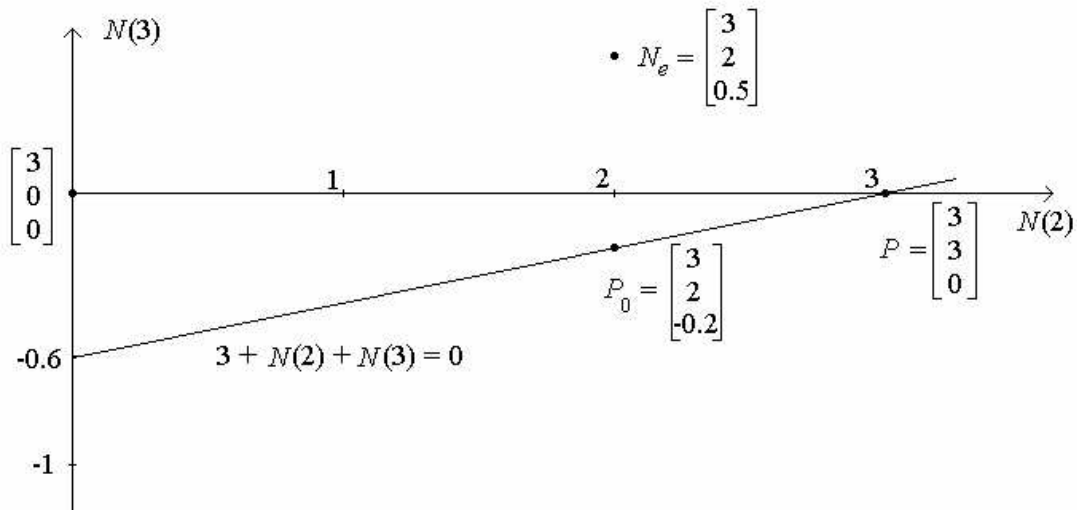


Figure 19 : Représentation du point N_e et du plan dans le plan de coupe $N(1) = 3$

Par conséquent, on calculera les distances dans \mathfrak{R}_+^n , et non dans \mathfrak{R}^n .

5.5 Rayon de stabilité

La définition du rayon de stabilité ne change pas : une fois que $B(Q_v)$ est connu, la valeur du rayon de stabilité est égale à la distance de N_{ref} à la plus proche frontière de $B(Q_v)$. Dans les deux paragraphes qui suivent, deux exemples sont traités dans le cas où les perturbations sur la commande sont quelconques.

5.5.1 Les machines sont uniformes

5.5.1.1 Calcul du rayon de stabilité

Soient la matrice Q_v et la commande de référence N_{ref} définies par :

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad N_{ref} = \begin{bmatrix} 1 \\ 2 \\ 1.7 \end{bmatrix}$$

Connaissant $B(Q_v)$, le vecteur Ω rassemble les distances de N_{ref} à chaque frontière :

$$B(Q_v) : \begin{cases} N(1) \leq \frac{3}{1.7}(N(2) + N(3)) & I_1 = \{1\} & J_1 = \{3,4\} \\ N(1) + N(2) \leq \frac{3.5}{1.2} N(3) & I_2 = \{1,2\} & J_2 = \{4\} \\ N(3) \leq \frac{3.2}{1.5}(N(1) + N(2)) & I_3 = \{3\} & J_3 = \{1,3\} \\ N(1) + N(3) \leq \frac{4.2}{0.5} N(2) & I_4 = \{1,3\} & J_4 = \{3\} \end{cases} \quad \Omega(Q_v, N_{ref}) = \begin{bmatrix} 3.133 \\ 0.671 \\ 2.203 \\ 1.678 \end{bmatrix}$$

Le rayon de stabilité vaut donc :

$$r(Q_v, N_{ref}) = 0.671$$

5.5.1.2 Interprétation géométrique et comparaison avec H_s

Pour les mêmes valeurs de Q_v et N_{ref} , on avait trouvé un rayon de stabilité valant 1 lorsque les perturbations conservaient la charge de N_{ref} . Ce résultat est cohérent : comme H_s est un

sous ensemble de \mathfrak{R}_+^n , la plus petite distance de N_{ref} à la contrainte (C_2) dans H_s est nécessairement supérieure ou égale à la valeur du rayon calculé dans \mathfrak{R}_+^n .

Sur la Figure 20, on a représenté une coupe de l'espace des commandes à $N(1) = 1$. Dans ce plan de coupe, H_s est un segment d'équation $1 + N(2) + N(3) = s$. $B_1(Q_v)$ est un sous segment de H_s qui n'a pas été représenté. La commande N_{ref} est sur le segment H_s . La contrainte (C_2) de l'expression de $B(Q_v)$ est un hyperplan passant par l'origine, dont l'intersection avec le plan de coupe $N(1) = 1$ a pour équation :

$$(C_2): 1 + N(2) = \frac{3.5}{1.2} N(3)$$

Il s'agit d'une droite qui ne passe pas par l'origine. (C_2) coupe H_s au point noté N_s : c'est le point de H_s appartenant à (C_2) dont la distance à N_{ref} est minimale. On peut le vérifier en s'assurant que la distance de N_{ref} à N_s vaut bien 1. Il s'agit de la valeur du rayon de stabilité obtenue dans H_s sur la contrainte (C_2) .

On note $r(Q_v, N_{ref})$ la valeur du rayon de stabilité dans \mathfrak{R}_+^n . Il vaut 0.671. Il est représenté par la flèche descendant de N_{ref} vers une commande notée N_r . N_r est la commande de (C_2) la plus proche de N_{ref} : on peut vérifier que sa distance à N_{ref} est égale à $r(Q_v, N_{ref})$.

On peut maintenant comparer le rayon de stabilité dans H_s et le rayon de stabilité dans \mathfrak{R}_+^n . Pour cela, il suffit de comparer les distances de N_{ref} à N_s et de N_{ref} à N_r . La distance de N_{ref} à N_s vaut 1, et elle est représentée par deux flèches partant de N_{ref} et aboutissant à N_s . On souligne ainsi que la distance séparant deux vecteurs en norme 1 est égale à la différence (en valeur absolue) de leurs composantes. Or, on ne passe d'un point de H_s à un autre point même infiniment proche qu'en modifiant au moins deux composantes du point. En effet, la somme des composantes des points de H_s doit rester égale à s .

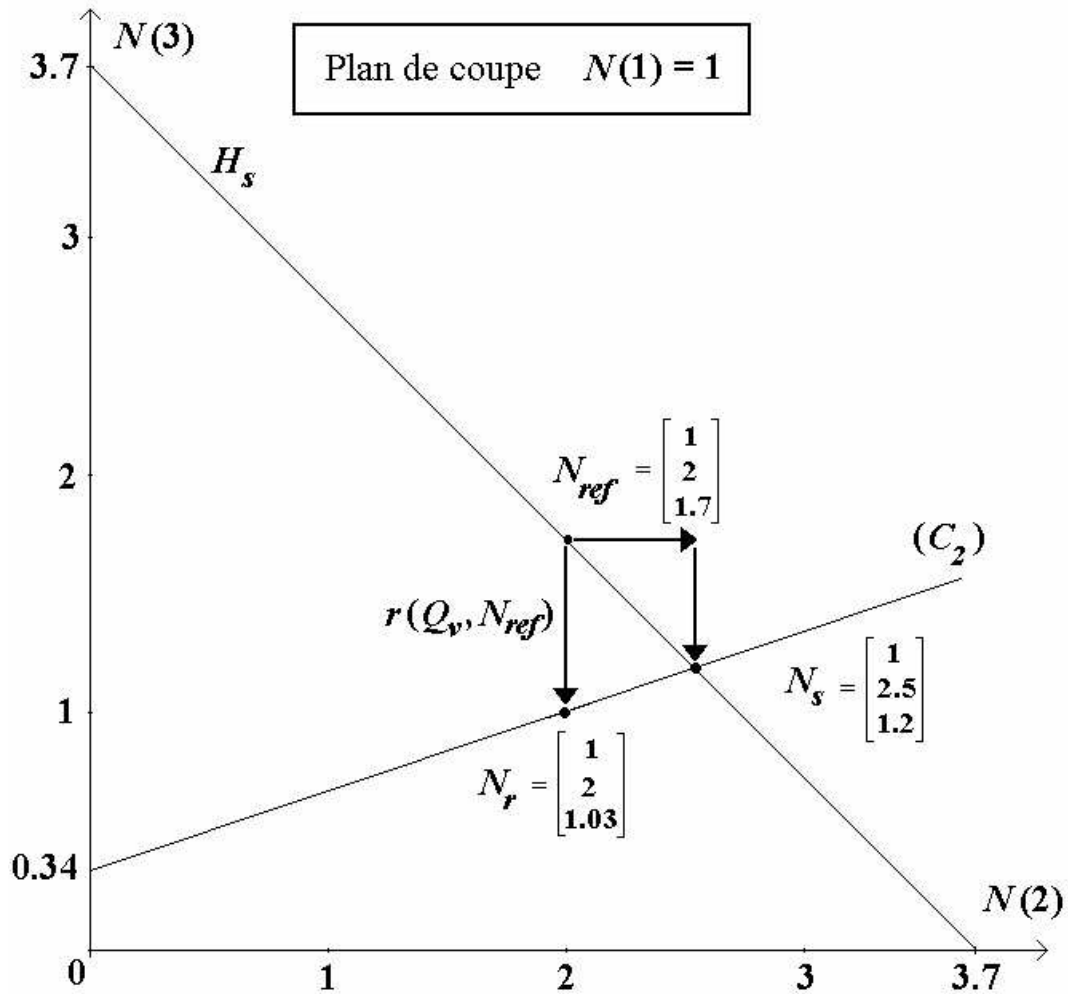


Figure 20 : Comparaison des rayons de stabilité dans H_s et dans \mathfrak{R}_+^n

La contrainte (C_2) est critique dans H_s comme dans \mathfrak{R}_+^n . On résume la comparaison des deux rayons de stabilité en affirmant que trouver le rayon de stabilité dans H_s consiste à déterminer le point de (C_2) le plus proche de N_{ref} dans H_s (ici c'est N_s), alors que trouver le rayon de stabilité dans \mathfrak{R}_+^n , c'est déterminer le point de (C_2) le plus proche de N_{ref} dans \mathfrak{R}_+^n (en l'occurrence, c'est N_r). Comme H_s est inclus dans \mathfrak{R}_+^n , il est logique que le rayon de stabilité dans \mathfrak{R}_+^n soit plus petit (ou aussi grand) que le rayon de stabilité dans H_s .

La Figure 21 est un zoom de la Figure 20 : pour ne pas surcharger le dessin, on n'a représenté que les ensembles H_s et (C_2) dans le plan de coupe $N(1) = 1$. La zone grisée est la boule de centre N_{ref} et de rayon $r(Q_v, N_{ref}) = 0.671$ (plus précisément, il s'agit de son intersection dans le plan de coupe). Dans l'espace à trois dimensions, cette boucle serait un polyèdre régulier ayant 6 sommets. Le point N_f est un point situé sur la frontière de cette boule. Tout point N_f de cette frontière vérifie l'équation $N_f = N_{ref} + dN_f$, avec :

$$\|dN_f\|_1 = r(Q_v, N_{ref}) = 0.671$$

Le segment en trait noir épais dont N_{ref} est le milieu est la boule de centre N_{ref} et de rayon 1 de H_m . La vue dans le plan de coupe $N(1) = 1$ ne permet pas de voir qu'il s'agit de l'hexagone grisé représenté sur la Figure 15.

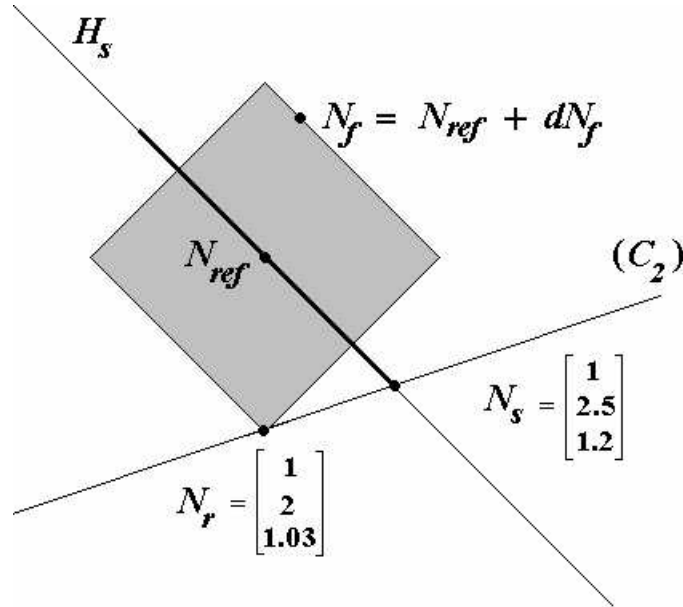


Figure 21 : Boule de rayon $r(Q_v, N_{ref})$ dans le plan de coupe $N(1) = 1$

La commande N_s n'est pas à l'intérieur de la boule de rayon $r(Q_v, N_{ref})$, mais elle est à la frontière de la boule de rayon 1 de H_s .

5.5.2 Les machines sont identiques

5.5.2.1 Calcul du rayon de stabilité

Soient la matrice Q_v et la commande de référence N_{ref} définies par :

$$Q_v = \begin{bmatrix} 1.3 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 1.5 & 1.5 & 0 & 1.5 \end{bmatrix} \quad N_{ref} = \begin{bmatrix} 0.65 \\ 0.3 \\ 3 \end{bmatrix}$$

Connaissant $B(Q_v)$, le vecteur Ω rassemble les distances de N_{ref} à chaque frontière :

$$B(Q_v) : \begin{cases} \frac{N(1)}{1.3} \leq \frac{1}{3} \times \left(\frac{N(2)}{0.2} + \frac{N(3)}{1.5} \right) & I_1 = \{1\} & J_1 = \{2,3,4\} \\ \frac{N(1)}{1.3} + \frac{N(2)}{0.2} \leq \frac{3}{1} \times \frac{N(3)}{1.5} & I_2 = \{1,2\} & J_2 = \{4\} \\ \frac{N(1)}{1.3} + \frac{N(3)}{1.5} \leq \frac{3}{1} \times \frac{N(2)}{0.2} & I_3 = \{1,3\} & J_3 = \{3\} \end{cases} \quad \Omega(Q_v, N_{ref}) = \begin{bmatrix} 0.5167 \\ 0.8 \\ 0.133 \end{bmatrix}$$

Le rayon de stabilité est : $r(Q_v, N_{ref}) = 0.133$. Dans H_m , on avait trouvé un rayon de stabilité valant 0.5, mais le rayon de stabilité était basé sur une distance en norme T . Le point N_s de H_m appartenant à la contrainte critique (C_3) et le plus proche de N_{ref} est :

$$N_s = \begin{bmatrix} 0.65 \\ 0.2 \\ 3.75 \end{bmatrix}$$

Calculons la distance de N_{ref} à N_s en norme 1 :

$$d(N_{ref}, N_s) = \|N_{ref} - N_s\|_1 = 0.85$$

Comme dans le cas où les vitesses ne dépendent que des machines, on vérifie que le rayon de stabilité est plus petit dans \mathcal{R}_+^n que dans tout sous ensemble (H_m en l'occurrence) : on a bien $0.133 \leq 0.85$.

5.6 Conclusion

Le théorème permettant d'associer un rectangle maximum de zéros dans la matrice de configuration Q à une contrainte délimitant l'ensemble des commandes équilibrées a été démontré dans le cas général (v est quelconque). Cependant, ce résultat n'a pu être exploité que dans les deux cas particuliers de distribution des vitesses sur le parc de machines. Cela est dû au fait qu'en général, le temps $T(I_k, J_k)$ ne s'exprime simplement en fonction de v et de N . Comme $T(I_k, J_k)$ est le temps minimum passé par les machines J_k à traiter les produits de type I_k , ce temps est obtenu en résolvant (PL 1) dans le cas de l'équilibre simple. Or, sauf cas particuliers, il n'est pas possible d'exprimer littéralement la solution optimale d'un programme linéaire.

6 Conclusions et perspectives

L'optimisation en contexte incertain et ses enjeux ont été présentés dans la première partie de cette thèse. Les définitions proposées pour la robustesse et la flexibilité visent à fournir des critères pertinents pour évaluer le comportement d'une solution destinée à être appliquée en contexte incertain. Cependant, aucune méthode générale de synthèse de solutions robustes ou flexibles n'est proposée. Les problèmes d'ordonnancement sont extrêmement variés, et les méthodes destinées à les résoudre en contexte certain sont elles aussi à l'image de cette diversité. S'il n'existe pas de méthode générale permettant d'obtenir des solutions performantes (au sens de la performance locale) en ordonnancement, il ne faut pas être surpris de l'absence de méthode de synthèse générale en ordonnancement robuste. Outre le fait que les diverses définitions de la robustesse ne concordent pas toujours (et c'est encore pire pour la flexibilité !), la difficulté vient de la nature des problèmes d'ordonnancement pour lesquels il ne semble pas exister de méthode de résolution générale.

6.1 Problème de la configuration *ex-nihilo*

L'objet de la deuxième partie de ce mémoire a été de s'attacher à appliquer les métriques de robustesse à un problème de machines parallèles multifonctions dans un premier temps, puis à mettre au point une méthode d'augmentation de la robustesse adaptée à ce problème dans un deuxième temps. Il n'a pas été question de réactivité, ni donc de flexibilité, dans la mesure où la modification de la configuration de ce parc de machines a des conséquences lourdes sur son exploitation (immobilisation des équipements et tests). On s'est donc concentré sur des méthodes prédictives robustes. En second lieu, et toujours pour des raisons d'exploitation et de faisabilité, le problème traité tient davantage de la reconfiguration que de la configuration. Il semble que le problème de génération de matrices de qualification robustes puisse être traité avec les outils qui ont été développés dans cette thèse. On pourrait par exemple envisager de partir d'une matrice de qualification « admissible minimale », c'est-à-dire une matrice pour laquelle aucun « 1 » ne peut être retiré sans que la matrice en question ne perde sa qualité d'admissibilité. On peut montrer facilement qu'une telle matrice a exactement $\max(n, m)$ « 1 ». L'algorithme d'ajout de « 1 » présenté dans ce mémoire pourrait alors servir à ajouter itérativement des « 1 », tout en minimisant le nombre de « 1 » à ajouter pour atteindre le niveau de robustesse désiré. A ce niveau, deux questions fondamentales restent ouvertes. Premièrement, comment construire la meilleure matrice de qualification « admissible minimale » ? On peut montrer que le nombre de matrices de qualification « admissibles minimales » est égal à $(\min(n, m))^{n-m}$. Ce nombre est d'autant plus grand que n et m sont différents. La deuxième question qui se pose concerne la pertinence de l'application successive de la méthode d'ajout de « 1 ». Autrement dit, l'ajout successif de p « 1 » est-il équivalent à un algorithme plaçant simultanément p « 1 » dans la matrice Q (pour un entier $p > 1$) ? En général, la réponse aux questions du type « un raisonnement local peut-il être appliqué globalement ? » est non. Cependant, aucun contre-exemple mettant en défaut la méthode proposée lors de l'ajout successif de plusieurs « 1 » n'a pu être trouvé pour le moment, ce qui bien sûr ne prouve rien.

6.2 Problème de la suppression des « 1 »

Il faut également noter que l'on ne peut envisager d'implanter la méthode d'augmentation de la robustesse d'une configuration courante proposée dans cette thèse sans lui adjoindre une méthode permettant d'enlever des « 1 » dans la matrice de qualification (donc les qualifications). En effet, la méthode proposée ne peut conduire, à terme, qu'à remplir la matrice de qualification de « 1 ». Or il est impératif de maîtriser les coûts de maintenance, directement proportionnels au nombre de « 1 » que compte la matrice de qualification. Intuitivement, il paraît intéressant de supprimer les « 1 » dont l'impact se limite à des contraintes (C_k) dont la frontière est éloignée de N_{ref} . Cependant, la détermination des contraintes définissant $B(Q_v)$ n'apporte des informations que sur les rectangles de zéros, et pas sur les coefficients valant « 1 ». Partant de l'idée simple que l'ajout d'un « 1 » suffit à « casser » un rectangle de zéros, il est peut-être intéressant de s'intéresser aux rectangles de « 1 » de la matrice de qualification et de s'interroger sur leur signification en terme de robustesse. Il sera peut-être possible d'appliquer la méthode d'ajout de « 1 » présentée dans ce mémoire pour augmenter le rayon de stabilité, à l'ajout de « 0 » dans le but de minimiser la diminution du rayon de stabilité.

6.2.1 Exemple

On reprend l'exemple introduit au paragraphe 5.5.1.1. Celui-ci n'est destiné qu'à montrer qu'il est parfois possible de remplacer un « 1 » par un « 0 » sans provoquer de diminution du rayon de stabilité. On avait :

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad N_{ref} = \begin{bmatrix} 1 \\ 2 \\ 1.7 \end{bmatrix} \quad \Omega(Q_v, N_{ref}) = \begin{bmatrix} 3.133 \\ 0.671 \\ 2.203 \\ 1.678 \end{bmatrix} \quad \text{et } r(Q_v, N_{ref}) = 0.671$$

$$B(Q_v) : \begin{cases} N(1) \leq \frac{3}{1.7} (N(2) + N(3)) & I_1 = \{1\} \quad J_1 = \{3, 4\} \\ N(1) + N(2) \leq \frac{3.5}{1.2} N(3) & I_2 = \{1, 2\} \quad J_2 = \{4\} \\ N(3) \leq \frac{3.2}{1.5} (N(1) + N(2)) & I_3 = \{3\} \quad J_3 = \{1, 3\} \\ N(1) + N(3) \leq \frac{4.2}{0.5} N(2) & I_4 = \{1, 3\} \quad J_4 = \{3\} \end{cases}$$

On peut voir que remplacer $Q(1,1)$ par « 0 » va affecter la contrainte (C_1) dont le rectangle de zéros s'agrandit, mais aussi les contraintes (C_3) et (C_4) dont les rectangles de zéros sont alors confondus : on décide d'éliminer la contrainte (C_4) qui est égale à (C_3). On note Q_v - la matrice de vitesse correspondante :

$$Q_{v^-} = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad B(Q_{v^-}) : \begin{cases} N(1) \leq \frac{2}{2.7}(N(2) + N(3)) & I_1 = \{1\} & J_1 = \{1,3,4\} \\ N(1) + N(2) \leq \frac{3.5}{1.2}N(3) & I_2 = \{1,2\} & J_2 = \{4\} \\ N(1) + N(3) \leq \frac{3.2}{1.5}N(2) & I_3 = \{1,3\} & J_3 = \{1,3\} \end{cases}$$

$$\Omega(Q_{v^-}, N_{ref}) = \begin{bmatrix} 1.741 \\ 0.671 \\ 0.893 \end{bmatrix} \text{ et } r(Q_{v^-}, N_{ref}) = 0.671$$

On constate que le rayon de stabilité reste inchangé malgré la suppression d'un « 1 » dans la matrice de qualification. Mais aucune méthode formelle n'est proposée pour déterminer les « 1 » pouvant être supprimés sans dégradation du rayon de stabilité.

6.3 Problème de l'impact des pannes survenant dans l'atelier

Du point de vue opérationnel, la prise en compte de l'incertain ne se limite pas aux perturbations que subit la commande en entrée de l'atelier : des pannes de machine peuvent se produire à l'intérieur de cet atelier. La prise en compte de ces pannes est d'autant plus importante que l'atelier de photolithographie occupe une position centrale dans le système de production de circuits intégrés. Dans l'hypothèse où la machine j tombe en panne, il est possible d'évaluer la capacité des machines restantes à assurer l'équilibre des commandes avec la méthode proposée dans cette thèse. Mais est-ce que l'équilibre de la commande reste un critère pertinent en situation de crise (en cas de panne) ? Pour parer aux pannes, les industriels conservent des qualifications « cachées » appelées *back up* : une machine j est en *back up* pour le produit i si elle est physiquement qualifiée pour ce type de produit mais que $Q(i,j) = 0$. Le choix des machines à « mettre en *back up* » en prévision des pannes des machines de l'atelier est un problème intéressant qui compléterait avantageusement les travaux présentés ici.

6.4 Problèmes de l'ordonnancement des produits sur les machines

Dans la seconde partie de cette thèse, il n'a été question que du choix de la configuration du parc de machines. Pour un parc de machines identiques ou uniformes représenté par une configuration préexistante et une commande prévisionnelle données, on peut garantir qu'il existe une solution de (PL 2) équilibrée. Or en réalité, il n'est possible d'affecter qu'un nombre entier de produits à une machine. Les programmes linéaires (PL 1) et (PL 2) auraient dû, en toute rigueur, être des programmes linéaires en nombres entiers. Ils sont en nombres réels pour plusieurs raisons : le nombre de produits traversant l'atelier est supposé suffisamment grand pour que l'écart entre une solution en nombres entiers et une solution en nombres réels ne soit pas significatif en terme de différence de charge des machines. D'autre part, si l'on considère les données industrielles présentées au chapitre 4.1.13.3, on constate que les machines ayant toutes une vitesse unitaire, il ne peut exister de solution équilibrée que

si le nombre total de produits est un multiple de m . Dans la version en nombres entiers de (PL 1) et de (PL 2) le critère d'équilibre de la charge n'aurait de sens que dans des cas particuliers très marginaux. Or en pratique, cette notion est très utile et il serait dommage de s'en passer au prétexte que le nombre de produits n'est pas rigoureusement un multiple de m . Un intéressant problème d'ordonnancement des produits sur les machines (pour une configuration et une commande donnée) pouvant constituer une perspective de ces travaux de thèse est présenté dans le paragraphe suivant.

6.4.1 Problème de la trajectoire de commande

Considérons la matrice de vitesses Q_v et la commande N_{ref} définies ci-après.

$$Q_v = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad N_{ref} = \begin{bmatrix} 8 \\ 6 \end{bmatrix} \quad B(Q_v) : \begin{cases} N(1) \leq 2 \times N(2) \\ N(2) \leq 2 \times N(1) \end{cases}$$

Le traitement de la commande N_{ref} par les $m = 3$ machines de ce parc peut être fait en assurant un taux d'occupation des machines de 100 % : on peut vérifier aisément que N_{ref} appartient à $B(Q_v)$. Si l'on charge uniformément les 3 machines, on constate que ce parc permet de consommer au maximum deux fois plus de produits d'un type donné que du type restant (il n'y a que deux types de produits). De ce fait, l'ensemble des états de commande atteignables, c'est-à-dire l'ensemble des états du stock d'entrée de l'atelier en partant de N_{ref} , est limité. Plus précisément, en chargeant uniformément les machines du parc avec pour commande N_{ref} , les produits restant à traiter ne peuvent pas sortir d'un cône de sommet N_{ref} dans l'espace des commandes. La trajectoire de commande, c'est-à-dire l'état du stock d'entrée de l'atelier, sera nécessairement comprise dans le cône grisé de la Figure 22 si toutes les machines sont chargées à l'instant t . Or, le cône n'est pas entièrement inclus dans $B(Q_v)$. Ceci montre que même si N_{ref} appartient à $B(Q_v)$, il est possible d'affecter les produits aux machines de sorte que toutes les machines n'aient pas la même charge.

Si un opérateur charge chaque machine à raison d'un produit par unité de temps, en affectant un produit de type 1 aux machines 1 et 2, et un produit de type 2 à la machine 3, il ne sera plus possible de garantir la même charge aux machines à $t = 4$: à cet instant, il ne restera que deux produits de type 2 et aucun produit de type 1. Or la machine 1 ne peut traiter que des produits de type 1, donc cette machine ne peut plus être chargée alors que le traitement de la commande n'est pas terminé. Au contraire, si on avait choisi de charger la machine 1 avec un produit de type 1 et les machines 2 et 3 avec un produit de type 2, il n'aurait plus été possible de garantir la même charge aux machines à partir de $t = 3$. A cet instant, il serait resté cinq produits de type 1 à traiter, mais plus aucun produit de type 2. La machine 3 aurait été affamée. Ces deux trajectoires de commande sont représentées sur la Figure 22 par les deux flèches partant de N_{ref} et atteignant les axes $N(1) = 0$ et $N(2) = 0$. Il faut également signaler qu'on a fait l'hypothèse d'une consommation linéaire des produits par les machines : si on raisonnait en nombres entiers, les trajectoires seraient « en escalier ».

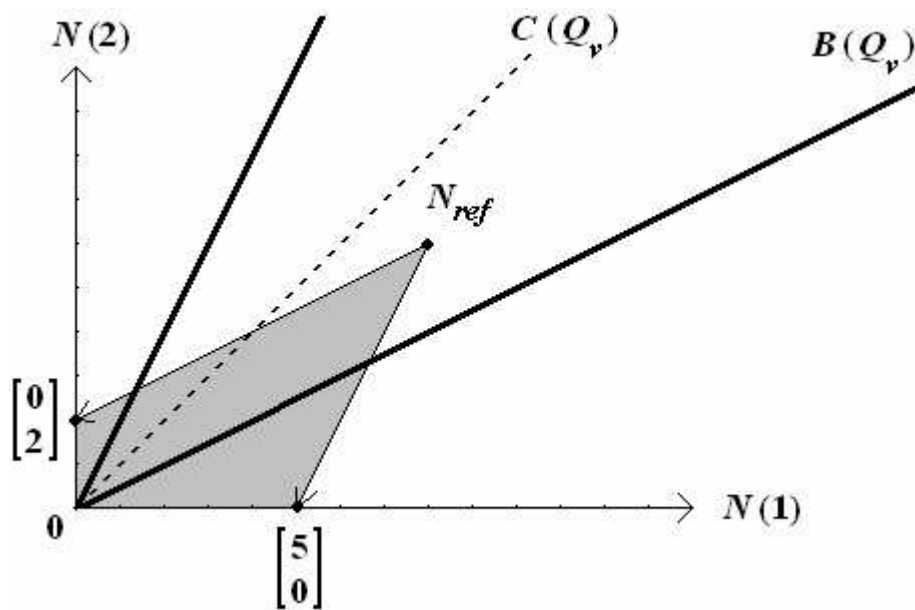


Figure 22 : Représentation du lieu des trajectoires de commandes

La ligne pointillée de la Figure 22 représente l'ensemble des commandes centrales, noté $C(Q_v)$. Les commandes centrales d'une matrice de vitesse Q_v sont les commandes pour lesquelles le rayon de stabilité est maximum pour un volume (ou un temps) de production fixé. Par définition, une commande centrale est à la distance maximale de l'ensemble des frontières de $B_1(Q_v)$. Pour choisir une « bonne » trajectoire de commande, on pourrait envisager de répartir les produits sur les machines de manière à se rapprocher de l'ensemble des commandes centrales. Cette idée a un autre avantage important : en cas d'arrivée de produits en temps réel dans le stock d'entrée de l'atelier, on est assuré de pouvoir absorber ces perturbations en exploitant au mieux la configuration courante à condition de se situer dans l'ensemble des commandes centrale au moment où survient cette arrivée de produits. Si aucun produit ne vient alimenter le stock, les produits restant dans le stock d'entrée peuvent être traités en garantissant l'équilibre de la charge des machines.

6.4.2 Ensemble des solutions optimales de (PL 1)

Un problème très voisin du dernier problème présenté concerne l'ensemble des solutions optimales de (PL 1). Il semble que le nombre et la structure des solutions optimales de (PL 1), pour une matrice de vitesse Q_v fixée et une commande N , dépende de la position de N par rapport aux frontières de $B(Q_v)$ (ou de $B_1(Q_v)$ lorsqu'on travaille à volume ou à temps de production constant). Le fait que (PL 1) ait un grand nombre de solutions est intéressant pour la gestion de l'atelier sur le terrain. Plus le nombre de solutions est grand, plus on dispose de flexibilité pour ordonnancer les produits à traiter. Il est donc intéressant de savoir dans quelles zones de l'espace de commande le maximum de flexibilité est disponible pour choisir une trajectoire de commande. Pour illustrer ces phénomènes, reprenons la matrice de vitesse définie au paragraphe 5.5.1.1, et considérons deux commandes notées N_1 et N_2 . (C_1) et (C_2), les deux premières contraintes définissant $B(Q_v)$ ont été rappelées.

$$Q_v = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 0.5 & 0 \\ 0 & 2 & 0 & 1.2 \end{bmatrix} \quad N_1 = \begin{bmatrix} 1 \\ 2 \\ \frac{36}{35} \end{bmatrix} \quad N_2 = \begin{bmatrix} \frac{5}{2} \\ \frac{235}{564} \\ 1 \end{bmatrix}$$

$$(C_1): \quad N(1) \leq \frac{3}{1.7}(N(2) + N(3)) \quad I_1 = \{1\} \quad J_1 = \{3, 4\}$$

$$(C_2): \quad N(1) + N(2) \leq \frac{3.5}{1.2} N(3) \quad I_2 = \{1, 2\} \quad J_2 = \{4\}$$

La commande N_2 se trouve à l'intersection des frontières de (C_1) et de (C_2) . On constate qu'il existe une unique solution optimale à (PL 1), notée R_{T1} pour le couple (Q_v, N_1) :

$$R_{T1} = \begin{bmatrix} \frac{5}{6} & \frac{5}{6} & 0 & 0 \\ 0 & 0 & \frac{5}{6} & 0 \\ 0 & 0 & 0 & \frac{5}{6} \end{bmatrix}$$

La commande N_1 se trouve à la frontière de (C_2) mais vérifie strictement (C_1) . Il existe une infinité de solutions optimales à (PL 1) pour le couple (Q_v, N_2) :

$$R_{T2}(\varepsilon) = \begin{bmatrix} 2\varepsilon & \frac{1}{2} - \varepsilon & 0 & 0 \\ \frac{6}{7} - 2\varepsilon & \frac{6}{7} - \frac{1}{2} + \varepsilon & \frac{6}{7} & 0 \\ 0 & 0 & 0 & \frac{6}{7} \end{bmatrix} \quad \left(\forall \varepsilon \quad t.q \quad 0 \leq \varepsilon \leq \min\left(\frac{1}{2}, \frac{3}{7}\right) \right)$$

On peut remarquer qu'il existe un « cycle » dans $R_{T2}(\varepsilon)$ (donc aussi dans Q_v) : il est possible de passer d'une solution à une autre en procédant à un échange de produits, dans la limite fixée par le réel ε . Cette limite provient du fait que les quantités de produit à affecter aux machines ne doivent pas être strictement négatives. Les observations faites sur de nombreux exemples font supposer que la variété des solutions est liée à l'existence de ces cycles, ainsi qu'à la commande N qui permet ou non d'exploiter ces cycles pour passer d'une solution optimale de (PL 1) à une autre : avec la commande N_1 , aucun cycle de Q_v n'a pu être exploité, d'où l'unicité de la solution de (PL 1). A l'instar de R_{T2} , On aurait pu indiquer la

solution R_{T1} par ε , mais avec la condition ($0 \leq \varepsilon \leq 0$). On peut définir un cycle plus formellement comme suit :

Soient I_c un ensemble ordonné de types de produits deux à deux distincts et J_c un ensemble ordonné de machines deux à deux distinctes. On suppose que I_c et J_c ont le même nombre d'éléments. Ce nombre, noté c est donc inférieur ou égal au minimum de n et de m . Soit k un entier défini modulo c : $k = 0, 1, \dots, c-1$, avec $(c-1) + 1 = 0$, et $0 - 1 = c-1$.

(I_c, J_c) est un cycle de Q_v si et seulement si on a pour tout k :

$$\begin{cases} Q_v(I_c(k), J_c(k)) > 0 \\ Q_v(I_c(k-1), J_c(k)) > 0 \\ Q_v(I_c(k), J_c(k+1)) > 0 \end{cases}$$

On peut remarquer que la matrice Q_v n'a qu'un seul cycle. Lorsqu'il existe au moins un cycle, on peut trouver une infinité de solutions pour (PL 1). Il semble que le nombre de degrés de liberté permettant de passer d'une solution à une autre (lorsque la commande N permet de les exploiter) soit égal au nombre de cycles de Q_v .

Du point de vue de la géométrie, on peut considérer $B(V)$ comme une intersection finie de demi-espaces (une contrainte (C_k) définit un demi-espace), donc comme un polyèdre. On rappelle qu'une k -face d'un polyèdre désigne une frontière de dimension k . Une 0-face est un sommet. L'exemple proposé tend à illustrer la conjecture suivante :

Soit N une commande de $B(Q_v)$. Si N appartient à une k -face (où k est l'entier modulo c défini plus haut), alors k cycles de Q_v sont exploitables pour passer d'une solution optimale de (PL 1) à une autre : il existe k degrés de liberté pour construire la matrice de répartition de N sur le parc défini par Q_v .

Encore une fois, il ne s'agit que d'une conjecture « vraisemblable » : une étude plus approfondie serait nécessaire pour la démontrer, ce qui constitue une autre perspective de ces travaux.

Cependant, les problèmes ayant trait à l'ordonnancement des produits sur les machines (la trajectoire de commande et l'ensemble des solutions optimales de (PL 1)) n'ont pas été approfondis pour des raisons propres au contexte industriel de cette étude. Dans l'unité de production où se situe l'atelier de photolithographie auquel on s'intéresse, les produits à fabriquer sont associés à des priorités déjà établies. Ces priorités reflètent les impératifs commerciaux de l'entreprise, mais aussi la nécessité d'une gestion globale efficace des produits, principalement pour éviter la propagation de « vagues de WIP » à travers l'unité de production. Il en résulte que l'affectation des produits aux machines sur le terrain est un problème considéré comme résolu. Si d'autres recherches devaient être menées à ce sujet, il conviendrait sans doute de ne plus se limiter à l'atelier seul, mais de considérer que les produits sortant de cet atelier sont destinés aux ateliers en aval (les ateliers clients). Il pourrait être intéressant de traiter les produits de l'atelier de photolithographie de manière à assurer qu'aucun atelier en aval n'est affamé, ni saturé. Cette perspective dépasse le cadre de l'étude

présentée dans cette thèse, où l'atelier de photolithographie a été considéré indépendamment de l'unité de production dans laquelle il se situe.

Références

- [Artigues, 1997] C. Artigues « Ordonnancement en temps réel d'ateliers avec temps de préparation des ressources ». Thèse de doctorat, Université Paul Sabatier, Toulouse, 1997.
- [Baker, 1974] K. Baker « Introduction to sequencing and scheduling » Wiley, New York, 1974.
- [Bénassy, 1987] J.Bénassy « La gestion de production », Hermès, Paris, 1987.
- [Bourgade, 1997] V. Bourgade « Real-time no-wait in process scheduling flowshop problems in uncertain environment », Conf. MCPL'97. IFAC-IFIP, pp.330-336. - Campinas - SP – Brazil, 1997.
- [Billaut, 1993] J-C. Billaut « Prise en compte des ressources multiples et des temps de préparation dans les problèmes d'ordonnancement en temps réel », Thèse de doctorat, Université Paul Sabatier, Toulouse, 1993.
- [Browne *et al.*, 1984] J. Browne « Classification of Flexible Manufacturing Systems » The FMS Magazine, vol 2, no. 2, avril, pp. 114-117, 1984.
- [Brucker *et al.*, 1997] P. Brucker B. Jurisch A Krämer « Complexity of scheduling problems with multi-purpose machines » Annals of Operations Research 70(1997)57-73.
- [Carlier et Chrétienne, 1988] J. Carlier, P. Chrétienne, « Problèmes d'ordonnancement : modélisation / complexité / algorithmes », Masson, 1988.
- [Chetouane *et al.*, 1997] F. Chetouane, D. Bouvet, A. Landrieu and Z. Binder « Control of flexible manufacturing system : interaction between production and transportation », MCPL'97, pp.103-108. - Campinas - SP – Brazil, 1997.
- [Chetouane, 1999] F. Chetouane, « Sur la robustesse dans les systèmes de production Application à la conduite d'un atelier de traitement de surface », Institut National Polytechnique de Grenoble, 1999.
- [Dean et Greenwald, 1998] T. Dean and L. Greenwald « A conditional scheduling approach to designing real-time systems », AI Planning systems, 1229-1234, 1998.
- [Demmou, 1977] R. Demmou « Etude de familles remarquables d'ordonnancements en vue d'une aide à la décision » Thèse de doctorat, Université Paul Sabatier, Toulouse, 1977.

[Erol, 1999] M. Erol, « Prise en compte de la flexibilité dans la planification dynamique (application à la flexibilité des ressources humaines) », Institut National Polytechnique de Grenoble, 1999.

[Erol, 1997] M. Erol « Analysis of production planning in case of the random demand », Computers and Industrial Engineering, vol. 37, no 1-2, 99. 21-25, 1997

[Erschler, 1976] J. Erschler « Analyse sous contraintes et aide à la décision pour certains problèmes d'ordonnancement ». Thèse de doctorat d'état, Université Paul Sabatier, Toulouse, 1976.

[Esquirol et Lopez, 1999] P. Esquirol , P. Lopez , « L'Ordonnancement », Economica, 1999.

[Esswein *et al.*, 2003] C. Esswein, V. Strusevich, J-C. Billaut « Qu'a-t-on à perdre en offrant de la flexibilité ? Etude du flow shop à deux machines », 4^{ème} Conférence Francophone de Modélisation et Simulation (MOSIM'03) du 23 au 25 avril 2003, Toulouse.

[Garey et Johnson, 1979] Michael R. Garey, David S. Johnson « Computer and Intractability A Guide to the Theory of NP-Completeness » W.H.Freeman and Company, New York.

[Groupe flexibilité du Gotha, 2002] Groupe Flexibilité du GOTHa., « Flexibilité et Robustesse en Ordonnancement », *Bulletin semestriel de la ROADEF* vol. 9. Disponible sous <http://www.roadef.org/lf/flexordo.ps> , 2002.

[Guinand *et al.*,1999] F. Guinand, A. Moukrim and E. Sanlaville (1999) « Scheduling With Communication Delays and On-Line Disturbances » In Springer Verlag, editor, Proc. EuroPar'99, Toulouse (France), vol 1685 of LNCS, 1999.

[Herroelen et Leus, 2002] W. Herroelen, R. Leus, « Project Scheduling Under Uncertainty - Survey and Research Potentials ». Invited paper to be published in the special issue of the European Journal of Operational Research that will contain selected papers presented at PMS2002, the Eighth International Workshop on Project Management and Scheduling, April 3-5, 2002, Valencia, Spain.

[Kouvelis *et al.*, 2000] P. Kouvelis, R.L. Daniels and G. Vairaktarakis « Robust Scheduling of a two-machine flow shop with uncertain processing times » , IIE Transactions 32, p 421-432, 2000.

[Kouvelis et Yu, 1997] P. Kouvelis and G. Yu « Robust Discrete Optimisation and its Applications » , Kluwer Academic Publisher, 1997.

[Lawler et Labetoulle] E. L. Lawler et J. Labetoulle « On preemptive Scheduling of Unrelated Parallel Processors by Linear Programming », Journal of the Association for Computing Machinery 25, 612-619, 1978.

- [Le Gall, 1989] A. Le Gall « Un système interactif d'aide à la décision pour l'ordonnancement et le pilotage en temps réel d'atelier ». Thèse de doctorat, Université Paul Sabatier, Toulouse, 1989.
- [Mahadev *et al.*, 1998] Mahadev, Pekec, Roberts, « On the meaningfulness of optimal solutions to scheduling problems : can an optimal solution be nonoptimal ? », *Operation Research* vol 46, Supp. No. 3, May-June 1998.
- [Manacher, 1967] G.K. Manacher. « Production and stabilization of real-time task schedules », *Journal of the ACM*, 14(3):439-465, July 1967.
- [Mahjoub, 2000] A. Mahjoub « Etude de la robustesse des algorithmes d'ordonnancement » Mémoire de DEA, Institut National Polytechnique de Grenoble, 2000.
- [Mencerey, 2002] T. Mencerey « Configuration du parc machines d'un atelier de photolithographie : prise en compte des irrégularités d'approvisionnement » Mémoire de DEA, Institut National Polytechnique de Grenoble, 2002.
- [Nourine et Raynaud, 1999] Nourine L., Raynaud O., « A fast algorithm for building lattices », *Information Processing Letters*, vol. 71, p. 199-204, 1999.
- [Penz *et al.*, 2001] B. Penz, C. Rapine, D. Trystram « Sensitivity analysis of scheduling algorithms », à paraître dans *Operational Research*.
- [Rossi, 2000] A. Rossi « Ordonnancement en milieu incertain » Mémoire de DEA, Institut National Polytechnique de Grenoble, 2000.
- [Rossi et Jacomino, 2001] Rossi A., Jacomino M., « Robustness and Flexibility as Performance Guarantee for Scheduling Problems in Uncertain Contexts », *Proceedings of the 3rd World Manufacturing Congress*, Rochester, 2001.
- [Roy, 1985] B. Roy « Méthodologie multicritère d'aide à la décision » Economica, Paris, 1985.
- [Sotskov *et al.*, 1998] Sotskov Y N., Wagelmans A.P.M., Werner F. « On the Calculation of the Stability Radius of an Optimal or an Approximate Schedule ». *Annals of OR*, vol. 83, p. 213 - 252, 1998.
- [Thomas, 1980] V. Thomas « Aide à la décision pour l'ordonnancement en temps réel d'atelier ». Thèse de doctorat, Université Paul Sabatier, Toulouse, 1980.
- [Vincke, 1999a] P. Vincke « Robust and neutral methods for aggregating preferences into an outranking relation », *European Journal of Operational Research*, No 122, pp 405-412, 1999.
- [Vincke, 1999b] P. Vincke « Robust solutions and methods in decision-aid », *Journal of Multi-Criteria Decision Analysis* No 8: pp 181-187, 1999.

[Viswanadham et Narahari, 1992] N. Viswanadham, Y. Narahari, « Performance Modeling of Automated Manufacturing Systems », Prentice-Hall, 1992.

[Zhang et Graves, 1997] H. Zhang, S.C. Graves (1997) « Cyclic scheduling in a stochastic environment », Operation Research vol 45. No. 6, November-December 1997.