

# Généralisation au problème multi-produit

Dans le précédent chapitre, nous avons apporté une première extension du modèle linéaire orienté opérations pour le TLBP/B-P. À présent, nous montrons une seconde généralisation, qui se situe, cette fois-ci, du point de vue du nombre de produits usinés. Plus exactement, nous montrons comment un modèle pour la configuration des lignes mono-produit peut être étendu à un modèle pour configurer une ligne multi-produit. Il s'agit de configurer la ligne de manière à ce que plusieurs types de produits, nécessitant chacun un ensemble propre d'opérations, puissent être usinés. Toutefois, il est nécessaire que les produits à usiner soient des produits proches du point de vue des caractéristiques d'usinage. En particulier, il est supposé qu'ils ont un certain nombre d'opérations communes pour leur fabrication [KL98]. Nous précisons, dans la suite, les conditions dans lesquelles une telle adaptation est envisageable.

Chaque produit est caractérisé par un ensemble des données comportant notamment l'ensemble des opérations à effectuer. De plus, les contraintes relatives aux opérations sont également propres à chaque type de produits, ainsi, les précédences et les contraintes d'inclusion qui sont données pour chaque type. Les contraintes relatives aux unités d'usinage, quant à elles, sont communes à la famille des produits. La démarche proposée revient à agréger l'ensemble des produits pour les considérer comme un produit unique. De cette manière, nous pourrions appliquer le modèle mono-produit au produit agrégé. Dans la section qui suit, nous décrivons plus précisément les hypothèses que nous avons adoptées dans ce cas.

## 8.1 Hypothèses

Les hypothèses inchangées par rapport à celles prises pour le TLBP/B-P, sont les suivantes :

1. Une opération ne doit pas être exécutée plus d'une seule fois par les unités sélectionnées sur la ligne,

2. Les coûts des stations et des unités d'usinage, ainsi que les temps d'exécution sont connus.
3. les unités d'usinages disponibles sont connues à l'avance, toutefois à la différence des problèmes précédents, le choix d'une unité n'implique pas forcément l'exécution de toutes les opérations qu'elle est capable d'exécuter. Ainsi, nous autorisons l'exécution partielle des opérations. En pratique, cette hypothèse est applicable dans le cas des machines reconfigurables dont les broches peuvent être modifiées (ajoutés ou supprimés).
4. les contraintes de précédence sont *non strictes* (de type  $\leq$ ), cette hypothèse résulte de la prise en compte des circuits dans le graphe de précédence agrégé. Plus exactement, nous transformons les opérations qui constituent un circuit en macro-opération qui sera alors effectuée par un outil combiné (outil à étages). C'est pourquoi les contraintes de précédence qui relient les opérations initiales doivent être moins strictes car nous autorisons leur exécution en même temps. Dans le cas non strict, l'interdiction imposée par une contrainte de précédence du couple  $(i, j)$  porte sur l'impossibilité d'exécuter  $j$  avant  $i$  uniquement, ce qui fait qu'il est possible d'exécuter  $i$  avant  $j$  ou bien en même temps. Il est important de signaler que nous pouvons également considéré le cas stricte en éliminant de  $\mathbf{B}$  tous les blocs  $b$  contenant simultanément les deux opérations  $i$  et  $j$ .

Dans la section qui suit, nous introduisons quelques notations nécessaires à la formulation du modèle multi-produits.

## 8.2 Notations

1.  $P$  est une famille de produits à fabriquer dont le nombre de produits est donnée par sa cardinalité  $|P|$ ,
2.  $N_p$  est l'ensemble des opérations du produit  $p = 1, \dots, |P|$ ,
3.  $\mathbf{N} = \bigcup_{p=1}^{|P|} N_p$  est l'ensemble des opérations de la famille qui comporte tous les types de produits,
4.  $\mathbf{B}$  est l'ensemble des blocs disponibles pour effectuer l'ensemble des opérations  $\mathbf{N}$ . En pratique, il est obtenu en fusionnant l'ensemble des blocs pour chaque type de produit et peut être éventuellement complété en ajoutant des outils combinés,
5.  $q_b$  est le coût du bloc  $b$  et  $t_b$  est son temps d'exécution,
6.  $C$  est coût moyen de l'ouverture d'une station,
7.  $m_0$  est le nombre maximal de stations possibles et  $n_0$  est le nombre maximal de blocs par station,
8.  $\Delta_t$  est l'intervalle de temps pour lequel la demande en produits de type  $p$  est connue et égale à  $d_p$ ,  $p = 1, \dots, |P|$ .

Afin de considérer les contraintes de tous les types de produits nous proposons de les agréger. Dans la section qui suit nous expliquons en détails la démarche adoptée.

## 8.3 L'agrégation des contraintes

L'agrégation des contraintes est un des moyens possibles pour la prise en compte simultanée de l'ensemble des contraintes qui concernent tous les types de produits. Nous l'avons appliquée pour obtenir l'ensemble total des contraintes de précédence et d'inclusion qui concernent les opérations. Suite à cette étape, nous déduisons des blocs à supprimer, au vue d'hypothèses que nous avons fournies, et finalement nous mettons à jour l'ensemble total des contraintes d'exclusion pour rester cohérent avec la réduction qui a été faite. Dans ce qui suit, nous décrivons d'abord de façon plus précise l'agrégation des contraintes liant les opérations et ensuite nous indiquons les substitutions nécessaires pour l'ensemble de blocs et des contraintes d'exclusion.

### 8.3.1 Les contraintes de précédence

Pour chaque type de produit  $p = 1, \dots, |P|$ , il existe un graphe de précédence  $G_p^{or} = (\mathbf{N}_p, D_p^{or})$ . Afin que l'ensemble des contraintes de la famille  $P$  soient respectées il suffit de fusionner l'ensemble des graphes de précédence pour obtenir un graphe total (agrégé) :

$$G^{or} = \bigcup_{p=1}^{|P|} G_p^{or},$$

Comme nous l'avons déjà évoqué, les contraintes de précédence sont supposées être *non strictes*, dans ce cas l'exécution d'opérations liées par une contrainte de précédence peut se faire en parallèle. Plus exactement, s'il existe  $(i, j) \in D_p^{or}$  alors il y a deux possibilités d'exécution pour la paire  $(i, j)$ , à savoir : l'opération  $i$  est exécutée strictement avant l'opération  $j$  ou celles-ci sont exécutées simultanément par une même tête d'usinage. Cela se traduit par le fait d'autoriser l'utilisation d'unités d'usinage avec outils étagés pouvant effectuer plusieurs opérations liées par des contraintes de précédence.

Le graphe global engendré par la fusion des graphes de précédence des différents types doit être cohérent en ce sens qu'il ne doit pas comporter de circuit.

Le graphe global  $G^{or}$  est construit, plus exactement, en appliquant les règles suivantes :

1. tout sommet appartenant à un des graphes  $G_p^{or}$  est rapporté dans le graphe global  $G^{or}$  ainsi que l'ensemble de ses arcs,
2. les redondances sont éliminées en supprimant tout arc reliant deux sommets pour lesquels il existe déjà un chemin,
3. les circuits doivent être identifiés et les opérations qui les composent sont considérées comme des macro-opérations qui sont remplacées par de simples sommets (il faut vérifier qu'il existe des blocs exécutant ces macro-opérations).

### 8.3.2 Les contraintes d'inclusion et d'exclusion

Pour les contraintes d'inclusion chaque type  $p$  est caractérisé par un ensemble  $D_p^{in}, p = 1, \dots, |P|$ . L'agrégation est effectuée en fusionnant tous les ensembles  $D_p^{in}$  pour obtenir un ensemble global  $D^{in}$ . La fusion des opérations formant un circuit dans le graphe global  $G^{or}$  imposent des modifications sur l'ensemble des autres contraintes, notamment, sur l'ensemble des contraintes d'inclusion. Plus précisément, lorsqu'une opération est à l'origine d'un circuit et qu'elle appartient désormais à la macro-opération, de plus si celle-ci est contenue dans un sous-ensemble d'inclusion quelconque alors ce sous-ensemble d'inclusion est complété avec les opérations manquantes de la macro-opération. De la même manière que dans le modèle mono-produit, il est alors possible de réduire la taille des éléments de l'ensemble  $D^{in}$  en fusionnant en un seul sous-ensemble tous ceux qui ont une intersection non vide.

Comme pour les contraintes d'inclusion, l'ensemble total  $D^{ex}$  est obtenu en fusionnant l'ensemble des exclusions de chaque type, soit :  $D^{ex} = \bigcup_{p=1}^{|P|} D_p^{ex}$

Il est également possible de réduire l'ensemble agrégé d'exclusion en éliminant les contraintes qui concernent les blocs qui ont été supprimés.

### 8.3.3 L'ensemble des blocs

Cet ensemble doit contenir l'ensemble des unités pour tous les types de produits, c'est donc l'union de tous les ensembles de blocs disponibles,  $\mathbf{B} = \bigcup_{i=1}^{|P|} \mathbf{B}_i$

De cet ensemble il faut supprimer tous les blocs qui n'exécutent qu'un sous-ensemble des opérations d'une macro-opération détectée précédemment. Des blocs complémentaires peuvent éventuellement être ajoutés pour les macro-opérations qui ne sont contenues dans aucun bloc existant (si c'est possible), autrement le problème serait irréalisable.

## 8.4 Un exemple

Nous reprenons l'exemple présenté dans notre article [BBD06] pour illustrer l'ensemble des notions introduites pour le modèle multi-produit. Pour commencer, nous décrivons les données et contraintes du problème puis nous déroulons la phase d'agrégation en appliquant les différents pré-traitements proposés.

### 8.4.1 Description des données et contraintes

Pour cet exemple, la famille comporte trois types de produits ( $|P| = 3$ ), notés :  $p_1, p_2$  et  $p_3$ . L'ensemble des opérations à effectuer, pour chaque type, est donné dans le tableau 8.1.

$p_i$	$ N_i $	$N_i$
$p_1$	10	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
$p_2$	12	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
$p_3$	10	$\{1, 2, 5, 7, 8, 9, 10, 11, 12, 13\}$

TAB. 8.1 – L'ensemble des opérations pour la famille de produits

L'ensemble total des opérations est donné par  $N = \bigcup_{i=1}^{|P|} N_i = \{1, \dots, 13\}$ , et les opérations communes sont  $\bigcap_{i=1}^{|P|} N_i = \{1, 2, 5, 7, 8, 9, 10\}$ .

Pour chaque type  $p_i$ , un graphe de précedence  $G_i^{or}$  est fourni, nous les rapportons dans la figure 8.1.

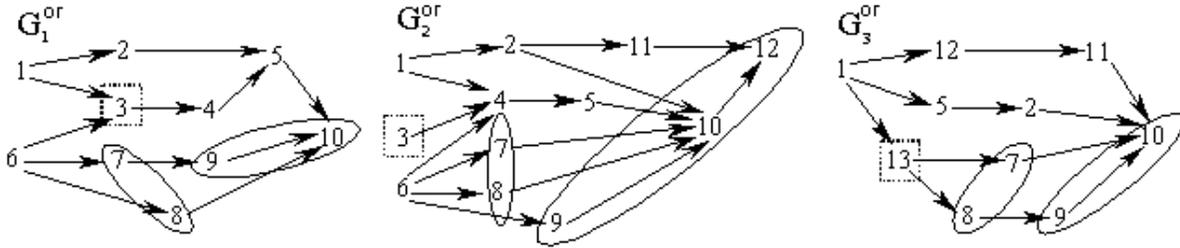


FIG. 8.1 – Les graphes de précedence de la famille de produits.

Pour chaque type de produit  $p_i$ , l'ensemble des contraintes d'inclusion est donné par  $D_i^{in}$ , comme suit :

$$D_1^{in} = \{\{7, 8\}, \{9, 10\}\}, D_2^{in} = \{\{7, 8\}, \{9, 10, 12\}\} \text{ et } D_3^{in} = \{\{7, 8\}, \{9, 10\}\}$$

De plus, les opérations  $D_s^{in} = \{\{3, 13\}\}$  doivent être exécutées sur la même station. Ces contraintes doivent être considérées en plus des contraintes d'inclusion de chaque type car elles n'ont pas pu être considérées dans un des  $D_i^{in}$ . Plus exactement, aucun des ensemble  $N_i$  ne contient les deux opérations en même temps. Ainsi, pour le type  $p_1$  et  $p_2$  c'est seulement l'opération 3 qui est présente dans  $N_1$  tandis que pour  $p_3$  c'est uniquement l'opération 13 qui est considérée.

Concernant les contraintes d'exclusion elles sont données par un seul ensemble  $D^{ex}$  car elles concernent les blocs disponibles et sont, de ce fait, communes à tous les types de produits. Elles sont décrites comme suit :

$$D^{ex} = \{\{b_1, b_6, b_7\}, \{b_1, b_9\}, \{b_5, b_{11}\}\}$$

Les blocs disponibles constituant  $\mathbf{B}$  sont décrits dans le tableau 8.2.

$b$	opérations	temps opératoire ( $t_b$ )	coût ( $q_b$ )
$b_1$	{1,3,6,13}	9	250
$b_2$	{1,3,13}	8	170
$b_3$	{1,2,7,8}	6	281
$b_4$	{2,5,9}	10	150
$b_5$	{2,5,7,8,11}	9	275
$b_6$	{2,6,9,10}	11	230
$b_7$	{4,6,8,10}	13	211
$b_8$	{4,7,8}	9	160
$b_9$	{4,7,8,9}	10	215
$b_{10}$	{5,12,13}	6	158
$b_{11}$	{10,11,12,13}	12	230
$b_{12}$	{2,5,10,11,12}	11	260

TAB. 8.2 – L'ensemble des blocs pour la famille de produits

Les temps de cycle  $T_{0_i}$ , pour chaque type  $p_i$ , sont déduits de la période commune  $\Delta_t = 48300$  et de la demande pour chaque type, donnée respectivement par :  $d_1 = 2100$ ,  $d_2 = 1932$  et  $d_3 = 2300$ , en utilisant l'expression  $T_{0_i} = \Delta_t/d_i$ . Ainsi,  $T_{0_1} = 23$ ,  $T_{0_2} = 25$  et  $T_{0_3} = 21$ .

### 8.4.2 Agrégation et pré-traitements

L'agrégation des contraintes de précédence de l'ensemble des graphes présentées dans la figure 8.1 permet de détecter deux circuits, à savoir :  $2 \mapsto 5 \cup$  et  $11 \mapsto 10 \mapsto 12 \cup$ . Chaque circuit est supprimé et remplacé par une macro-opérations substituant aux opérations formant le circuit. Dans ce cas, nous obtenons deux macro-opérations que nous désignons par les sommets  $a = \{2, 5\}$  et  $b = \{10, 11, 12\}$  dans le graphe agrégé de la figure 8.2.

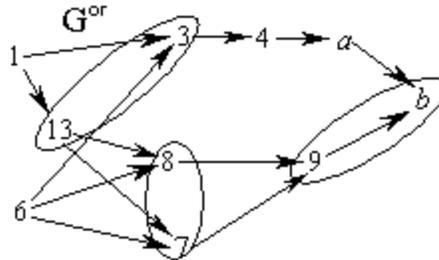


FIG. 8.2 – Le graphe de précédence pour la famille de produits.

L'introduction des macro-opérations engendrent des modifications quant à la composition de l'ensemble des opérations de chaque type de produit. Nous indiquons ces repercussions dans le tableau 8.3

La présence de ces macro-opérations engendre également des mises à jour de l'ensemble des blocs disponibles  $\mathbf{B}$ . Plus précisément, nous appliquons les deux règles suivantes :

$p_i$	$ \mathbf{N}_i $	$\mathbf{N}_i$
$p_1$	9	$\{1, a, 3, 4, 6, 7, 8, 9, b\}$
$p_2$	9	$\{1, a, 3, 4, 6, 7, 8, 9, b\}$
$p_3$	7	$\{1, a, 7, 8, 9, b, 13\}$

TAB. 8.3 – La modification de l’ensemble des opérations pour la famille de produits

1. chaque bloc ne considérant qu’un sous-ensemble d’une macro-opération quelconque est à supprimer de l’ensemble  $\mathbf{B}$ .
2. lorsqu’il existe un circuit dans le graphe de précedence reliant les blocs, il faut éliminer ce circuit en supprimant un des blocs qui le forme. Nous avons fait le choix de supprimer le bloc le plus coûteux par opération, toutefois nous pourrions également appliquer d’autres règles si le problème devient infaisable, par exemple, ne pas supprimer les blocs qui sont l’unique possibilité d’exécution de certaines opérations.

L’application de la première règle sur l’ensemble  $\mathbf{B}$  induit l’élimination des blocs  $b_3, b_5, b_6, b_7$  et  $b_{10}$ .

Quant à l’application de la seconde règle, elle implique d’éliminer un des blocs qui forment le circuit montré dans la figure 8.3.

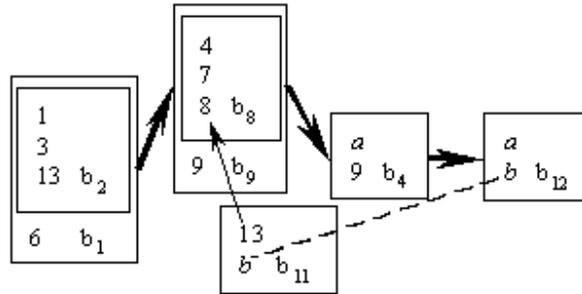


FIG. 8.3 – Le graphe de précedence pour les blocs.

Le bloc choisi parmi  $b_4, b_8, b_{11}$  et  $b_{12}$  est  $b_{11}$  car c’est celui dont le coût par opération est le plus élevé (57,5). Ce qui ramène l’ensemble des blocs disponibles à :

$$\mathbf{B} = \{b_1, b_2, b_4, b_8, b_9, b_{12}\}.$$

L’ensemble des contraintes d’inclusion doit également être mis à jour en considérant les macro-opérations.

$$D_1^{in} = \{\{7, 8\}, \{9, b\}\}, D_2^{in} = \{\{7, 8\}, \{9, b\}\} \text{ et } D_3^{in} = \{\{7, 8\}, \{9, b\}\}$$

L’union de l’ensemble de ces contraintes plus la contrainte additionnelle entre produits :

$$D^{in} = \{\{7, 8\}, \{9, b\}, \{3, 13\}\}.$$

De la même manière que pour les autres ensembles, les contraintes d’exclusion doivent

être modifiées dans le sens où il faut prendre en compte l'élimination de certains blocs. Du fait qu'un sous-ensemble d'exclusion traduit l'interdiction de l'affectation simultanée des blocs correspondant, il faut alors supprimer tous les sous-ensembles de blocs faisant intervenir un des blocs supprimés.

Pour cet exemple, cela revient à garder uniquement  $\{b_1, b_9\}$  dans  $D^{ex}$ .

## 8.5 Conclusion

L'intérêt du problème multi-produit que nous avons étudié dans le présent chapitre est incontestable. Malgré cette réalité industrielle, ce problème n'a pas encore été investi dans le contexte de l'usinage ; notre contribution s'inscrit dans ce contexte en généralisant les travaux effectués dans la seconde partie de cette thèse. En effet, le noyau de l'approche proposée pour le TLBP/B-M ou encore pour le problème multi-produit réside dans la formulation du modèle pour le TLBP/B-P étudié dans la seconde partie de ce mémoire. Plusieurs extensions deviennent alors possibles même si cela demande parfois de redéfinir certaines hypothèses. En particulier, nous avons supposé que les contraintes de précédence dans le cas multi-produit ne sont plus au sens strict telles qu'elles ont été définies pour le TLBP/B-P, et ce en raison de la présence de macro-opérations introduites suite à l'agrégation des contraintes. En effet, cette considération au sens non strict permet de contourner le problème de circuit qui peut se poser en envisageant d'utiliser des outils combinés capables d'effectuer certaines opérations liées par des précédences sans les dissocier.

# Conclusion générale et perspectives

Les travaux effectués dans cette thèse s'inscrivent dans le cadre de l'étude et du développement des systèmes d'aide à la décision pour la configuration des lignes d'usinage à partir d'équipements standard. Plus précisément, la démarche proposée fournit au concepteur des outils efficaces, fondés sur des méthodes d'optimisation pour répondre au problème de conception et reconfiguration de ces lignes.

Nous avons souligné la complexité du problème posé et les caractéristiques essentielles de ces lignes imposant la recherche de nouvelles approches pour le problème de leur configuration. Nous avons évoqué notamment les regroupements des opérations en blocs généralisant ainsi le problème d'équilibrage de lignes d'assemblage (SALBP). Ces blocs correspondent physiquement à des unités d'usinage standard effectuant chacune un sous-ensemble d'opérations. L'ensemble de ces unités est déterminé suite à une étude réalisée en amont. La nécessité de la prise en compte simultanée de tous ces éléments a engendré une difficulté supplémentaire par rapport aux travaux déjà étudiés dans la littérature. Ainsi, les méthodes existantes ne sont pas à même de résoudre ce nouveau problème.

Cette thèse a permis de proposer une modélisation efficace pour ce nouveau problème, mais aussi plusieurs algorithmes améliorant les performances des approches suggérées.

Dans la seconde partie de ce mémoire, nous avons testé une approche basée sur la programmation par contraintes. Nous avons opté pour l'approche PPC en raison de la nature fortement contrainte du problème traité et de la disponibilité d'outils tels que ILOG Solver. Dans ce cadre, nous avons apporté une formulation générique et des règles de propagation de certaines contraintes. Suite aux tests expérimentaux effectués, l'approche PPC s'est avérée insuffisante pour la résolution des instances de moyenne et grande taille. En effet, l'approche a atteint ses limites assez rapidement même si elle apporte des solutions réalisables assez vite. Nous pensons que ce comportement est dû à la nature de l'objectif qui est une somme pondérée de plusieurs variables. Nous avons proposé d'utiliser une borne de type relaxation linéaire en appliquant la propagation des différentes contraintes du problème. Au vu des résultats, ce schéma doit être amélioré en envisageant de définir une contrainte globale pour permettre une meilleure propagation sur l'objectif. En outre, nous pensons qu'il faudrait développer nos propres outils pour appliquer la formulation et les techniques de propagation suggérées. En effet, ILOG Solver ne permet pas de gérer l'espace mémoire lors de la construction de l'arbre de branchements, ce qui réduit la taille des instances qui peuvent être traitées.

Ensuite, nous nous sommes intéressés à la programmation linéaire en nombres entiers pour laquelle nous avons proposé deux modèles. Le premier modèle est orienté blocs car

les contraintes sont exprimées en fonction des blocs d'opérations. Quant au modèle orienté opérations, il est formulé en tenant compte de la nature des contraintes, c'est-à-dire que les contraintes qui sont relatives aux opérations sont exprimées en fonction de ces dernières. Dans les deux cas, nous avons proposé d'abord d'analyser les données afin d'optimiser leur représentation. Pour ce faire, nous avons introduit différents éléments de modélisation avant d'affiner la formulation du modèle final.

Par la suite, nous avons proposé différents algorithmes pour la réduction de la taille des modèles. Plus précisément, ces algorithmes exploitent la structure des contraintes du problème pour en déduire des impossibilités d'affectation pour chaque bloc d'opérations. Ceci revient à définir un intervalle contenant les indices des stations dans lesquelles chaque bloc peut être assigné. Ainsi, en réduisant ces intervalles nous avons diminué le nombre de variables binaires des modèles. Nous avons montré dans la phase expérimentale l'apport de ces algorithmes qui ont permis de réduire de façon considérable les temps de calcul. Il est également intéressant de signaler que les algorithmes les plus spécifiques sont les plus performants en termes de réduction du nombre de variables, mais pas forcément en termes de temps de calcul. Ainsi, nous avons opté pour l'Algorithme 2 car il a un bon rapport nombre de variables éliminées / temps de calcul.

Dans la phase expérimentale, nous avons réalisé d'abord une analyse comparative entre les deux modèles linéaires pour en évaluer les performances. Ces résultats ont permis de constater la supériorité du modèle orienté opérations résolvant le plus grand nombre d'instances dont la taille atteignait les 80 opérations. Le modèle a permis notamment de réduire de façon significative le temps de résolution pouvant le diviser par 100, voire plus pour certaines instances de grande taille. Ainsi, l'intérêt de la formulation orientée opérations est d'autant plus important pour les familles difficiles.

Nous avons poursuivi l'étude du modèle orienté opérations pour mesurer l'impact de certaines caractéristiques du problème sur les performances du modèle. Il a été décelé notamment que les instances les plus difficiles à résoudre étaient celles ayant une faible densité du graphe de précedence. Nous pensons que cela est dû au fait que le nombre de permutations pour l'affectation des blocs devient important lorsque la densité est faible. Nous avons également montré que le nombre de blocs disponibles a une influence forte vu que les instances ayant le nombre le plus important de blocs ont requis jusqu'à 100 fois plus de temps de calcul. Pour finir, nous avons étudié l'impact du nombre d'opérations par bloc et du nombre de stations maximum. Cette étude a également permis de conclure que les valeurs numériques de ces caractéristiques avaient une incidence directe sur le temps de calcul.

Dans la troisième partie de cette thèse, nous avons présenté deux extensions du problème initial. Dans la première extension, nous avons suggéré une généralisation du modèle orienté opérations pour le problème de configuration des lignes dont le mode d'activation des unités est mixte. Ce problème est plus complexe que le précédent car plusieurs unités d'usinage peuvent être mise en parallèle dans un même étage et une station peut être composées de plusieurs étages séquentiels. L'étude de ce mode a nécessité l'introduction de nouvelles contraintes pour tenir compte des incompatibilités d'unités pouvant être affectées à un même étage. Ensuite, une seconde extension a été étudiée en adaptant le modèle orienté opérations pour les lignes multi-produit. L'approche qui a été proposée se base sur l'agrégation des

contraintes et données pour chaque type de produit.

Nous avons ainsi montré que l'étude du problème initial (noté TLBP/B-P) considérant le mode d'activation parallèle a rendu possible différentes extensions définissant des problèmes plus complexes tels que celui où le mode d'activation est mixte pour les unités d'usinage ou encore celui considérant plusieurs types de produits dans la ligne. Les approches proposées pour la modélisation et la résolution de ces extensions s'appuient ainsi essentiellement sur le modèle proposé pour le problème initial.

Par ailleurs, ces travaux ont permis d'ouvrir de nouvelles perspectives pour les études futures. Par exemple, il serait notamment intéressant d'introduire le TLBP/B2, par analogie à la notation du SALBP-2, où l'objectif à minimiser serait le temps de cycle au lieu du coût de la ligne. Cette étude pourrait s'inscrire, comme pour le SALBP, dans une approche incrémentale globale optimisant d'abord le temps de cycle en considérant le TLBP/B2, ce qui permet d'obtenir une valeur du temps de cycle pour pouvoir lancer, ensuite, l'étude du TLBP/B optimisant le coût. De plus, une autre généralisation de ces deux problèmes peut être envisagée où le critère à optimiser serait l'efficacité de la ligne (définie dans ce cas par la minimisation du temps de cycle et du coût).

L'intérêt pratique du TLBP/B2 est incontestable particulièrement pour les lignes à reconfigurer. Plus exactement, lorsque la ligne est déjà mise en service et que la demande croît, il faut alors reconfigurer cette ligne de manière à minimiser le temps de cycle. Ainsi, à partir des unités et stations mises en place, il s'agit bien de retrouver une structure de ligne minimisant le temps de cycle.

Pour conclure, nous avons proposé des outils d'aide à la décision basés sur des méthodes de programmation mathématique pour la configuration des lignes d'usinage. Qu'il s'agisse de conception préliminaire ou de reconfigurations de ces lignes, l'intérêt industriel et les enjeux économiques sont capitaux. Nous avons souligné notamment les sommes colossales investies dans la mise en place de ce type de lignes, qui se chiffrent en millions d'euros, rendant l'usage d'outils d'optimisation indispensable. D'autre part, le gain réalisé grâce à l'emploi d'une telle démarche se répercute directement sur le coût de revient unitaire, aussi une étude minutieuse devient stratégique à ce stade.



# Notations

## Pour une ligne de transfert

$T$	temps de cycle effectif de la ligne
$T_0$	temps de cycle maximum de la ligne
$m$	nombre de stations composant la ligne
$m_0$	nombre maximum de stations dans la ligne
$m^*$	borne inférieure sur le nombre de stations du <i>TLBP</i>

## Pour les stations

$S_k$	la $k^{\text{ème}}$ station de la ligne
$C$	coût moyen de la mise en place d'une station
$n_0$	nombre maximum d'unités d'usinage par station
$y$	variable de décision déterminant le nombre de stations établies (modèle orienté blocs)
$y_k$	variable binaire déterminant l'ouverture ou non de la $k^{\text{ème}}$ station
$TS_k$	le temps de travail de la $k^{\text{ème}}$ station pendant chaque cycle

## Pour les étages

$q_0$	le nombre maximum d'étages par station
$s_j^k$	le $j^{\text{ème}}$ étage de la $k^{\text{ème}}$ station
$TS_j^k$	le temps de travail du $j^{\text{ème}}$ étage la $k^{\text{ème}}$ station pendant chaque cycle
$t_j^k$	la variable de décision délimitant le temps de travail effectif du $j^{\text{ème}}$ étage de la $k^{\text{ème}}$ station

## Pour les opérations

$i, j, o, o'$	des opérations
$\mathbf{N}$	ensemble des opérations
$D^{or}$	ensemble des contraintes de précédence entre les opérations
$\Gamma^-(b)$	ensemble des opérations prédécesseurs du bloc $b$
$D^{in}$	collection de sous-ensembles représentant les contraintes d'inclusion
$d$	sous-ensemble de contraintes d'inclusion appartenant à $D^{in}$
$G^{or}$	graphe de précédence pour les opérations
$G_{inv}^{or}$	graphe de précédence inversé pour les opérations
$g$	un niveau d'opérations
$rank(j)$	rang de l'opération $j$ dans le graphe de précédence $G^{or}$
$\overline{rank}(j)$	rang de l'opération $j$ dans le graphe de précédence inversé $G_{inv}^{or}$

$G^{in}$  graphe d'inclusion pour les opérations

**Pour les blocs**

$\mathbf{B}$  ensemble des blocs disponibles (unités d'usinage)  
 $B, B'$  sous-ensembles de blocs disponibles appartenant à  $\mathbf{B}$   
 $b, b'$  des blocs appartenant à  $\mathbf{B}$   
 $t_b$  temps opératoire du bloc  $b$   
 $q_b$  coût du bloc  $b$   
 $\mathcal{N}(b)$  ensemble des opérations contenues dans le bloc  $b$   
 $x_{bk}$  variable de décision pour l'affectation du bloc  $b$  à la station  $k$   
 $x_{bjk}$  variable de décision pour l'affectation du bloc  $b$  au  $j^{\text{ème}}$  étage de la  $k^{\text{ème}}$  station  
 $D^{ex}$  ensemble des contraintes d'exclusion (incompatibilité)  
 $E$  sous-ensemble de blocs incompatibles  
 $D^{pb}$  collection des contraintes de parallélisme  
 $G^{EB}$  graphe de précédence pour les blocs  
 $head_b$  la station au plus tôt du bloc  $b$   
 $tail_b$  la station au plus tard du bloc  $b$

## Bibliographie

- [Aig95] M. Aigner. *Turán's Graph Theorem*. Amer. Math. Monthly 102, 1995.
- [Ame00a] M. Amen. An exact method for cost-oriented assembly line balancing. *International Journal of Production Economics*, 64 :187–195, 2000.
- [Ame00b] M. Amen. Heuristic methods for cost-oriented assembly line balancing: a survey. *International Journal of Production Economics*, 68 :1–14, 2000.
- [Ame01] M. Amen. Heuristic methods for cost-oriented assembly line balancing: a comparison on solution quality and computing time. *International Journal of Production Economics*, 69 :255–264, 2001.
- [Arc66] A.L. Arcus. COMSOAL: a computer method of sequencing operations for assembly lines. *International Journal of Production Research*, 4 :259–277, 1966.
- [AS93] R.G. Askin and C.R. Standridge. *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, Inc, 1993.
- [Bay86] I. Baybars. A survey of exact algorithms for the simple line balancing problem. *Management Science*, 32 :909–932, 1986.
- [BBD06] S. Belmokhtar, A.I. Bratcu, and A. Dolgui. *Modular Machining Line Design and Reconfiguration: Some Optimization Methods*. In *Manufacturing the Future: Concepts-Technologies-Visions*, pages 125 –152. 3 -86611-198-3. pro literatur Verlag, Mammendorf, Germany, 2006.
- [BD62] R. Bellman and S.E. Dreyfus. *Applied Dynamic Programming*. Princeton University Press, Princeton, 1962.
- [BDDI06] S. Belmokhtar, A. Dolgui, X. Delorme, and I. Ihnatsenka. Optimizing modular machining design problem. In *Information Control Problems in Manufacturing (INCOM06 Symposium) de l'IFAC*. Elsevier, 2006.
- [BDG<sup>+</sup>04] S. Belmokhtar, A. Dolgui, N. Guschinsky, I. Ihnatsenka, and G. Levin. Optimization of transfer lines with constraint programming approach. In *Proceedings of the 34<sup>th</sup> International Conference on Computers and Industrial Engineering*, San Francisco, 14-16 Nov, 2004.
- [BDGL05a] S. Belmokhtar, A. Dolgui, N. Guschinsky, and G. Levin. Un programme linéaire pour la conception de lignes de transfert à partir d'un ensemble fixe de têtes d'usinage. *Contributions sélectionnées du congrès de la ROADEF, Presses Universitaires François Rabelais*, pages 259–276, 2005.

- [BDGL05b] S. Belmokhtar, A. Dolgui, N. Guschinsky, and G. Levin. A new model for transfer line with simultaneous activation of multi-spindle heads at workstations. In *Proceedings of the 35<sup>th</sup> International Conference on Computers and Industrial Engineering*, <http://www.umoncton.ca/cie/>, Turkey, 19-22 Juin, 2005.
- [BDGL06] S. Belmokhtar, A. Dolgui, N. Guschinsky, and G. Levin. Integer programming models for logical layout design of modular machining lines. *Computers and Industrial Engineering*, 51 :502–518, 2006.
- [Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, 1957.
- [BLPN03] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. Kluwer Academic Publisher, London, 2003.
- [BR03] J. Bukchin and J. Rubinovitz. A weighted approach for assembly line design with station paralleling and equipment selection. *IIE Transactions*, 35 :73–85, 2003.
- [Bru93] A.M Bruckstein. Why the ant trails look so straight and nice. *The Mathematical Intelligence*, 15(2) :59–62, 1993.
- [BS78] K.R. Baker and L.E. Schrage. Finding an Optimal Sequence by Dynamic Programming: An Extension to Precedence-Related Tasks. *Operations Research*, 26 :111–120, 1978.
- [BSW73] G.M. Buxey, N.D. Slack, and R. Wild. Production flow line system design—a review. *AIIE Transactions*, 5 :37–48, 1973.
- [BT00] J. Bukchin and M. Tzur. Design of flexible assembly line to minimize equipment cost. *IIE Transactions*, 32 :585–598, 2000.
- [Car82] J. Carlier. The one-machine sequencing problem. *European Journal of Operational Research*, 11 :42–47, 1982.
- [Chi98] W.C. Chiang. The application of a tabu search metaheuristic to the assembly line balancing problem. *Operations Research*, 11 :209–227, 1998.
- [Col02] Alain Colmerauer. *Cours de DEA sur la programmation logique*. <http://www.lim.univ-mrs.fr/>, 2002.
- [Dan51] G. Dantzig. *Maximization of a linear function of variables subject to linear inequalities*. In: *Activity Analysis of Production and Allocation*. Wiley, New-York, 1951.
- [Dan66] S. Dano. *Industrial Production Model*. Vienna, Springer, 1966.
- [Das03] A. Dashchenko. *Manufacturing Technologies for Machines of the Future 21<sup>st</sup> Century Technologies*. Springer, 2003.
- [Dav91] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

- [DDX89] Y. Dallery, R. David, and X. Xie. Approximate analysis of transfer lines with unreliable machines and finite buffers. *IEEE Transactions on Automatic Control*, 34 :943–953, 1989.
- [DES06] A. Dolgui, A.V. Ereemeev, and V.S. Sigaev. Hybrid algorithm for buffer allocation in tandem production lines HBBA. *Journal of Intelligent Manufacturing*, 2006. (In Press).
- [DFG<sup>+</sup>05] A. Dolgui, B. Finel, N. Guschinsky, L. Levin, and F. Vernadat. An heuristic approach for transfer lines balancing. *Journal of Intelligent Manufacturing*, 16(2) :159–171, 2005.
- [DFG<sup>+</sup>06a] A. Dolgui, B. Finel, O. Guschinskaya, N. Guschinsky, L. Levin, and F. Vernadat. Balancing large-scale machining lines with multi-spindle heads using decomposition. *International Journal of Production Research*, 2006. (in Press).
- [DFG<sup>+</sup>06b] A. Dolgui, B. Finel, N. Guschinsky, L. Levin, and F. Vernadat. MIP approach to balancing transfer lines with blocks of parallel operations. *IIE Transactions*, 38 :869–882, 2006.
- [DG89] Y. Dallery and S.B. Gershwin. Manufacturing flow line systems: a review of models and analytical results,. *Queueing Systems Theory and Applications, Special Issue on Queueing Models of Manufacturing Systems*, 12 :3–94, 1989.
- [DG97] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1 :53–66, 1997.
- [DGL<sup>+</sup>04] A. Dolgui, N. Guschinsky, G. Levin, M. Louly, and S. Belmokhtar. Balancing of transfer lines with simultaneously activated spindles. In *Information Control Problem in Manufacturing: A Proceedings Volume of the IFAC Symposium*. Elsevier, April 5-7 2004.
- [DGL06] A. Dolgui, N. Gushinsky, and G. Levin. A special case of transfer lines balancing by graph approach. *European Journal of Operational Research*, 168 :732–746, 2006.
- [DI05] A. Dolgui and I. Ihnatsenka. Balancing transfer lines with mixed activation of spindle heads. Technical Report 500-002, Ecole des Mines de St Etienne, 2005.
- [DIB05] A. Dolgui, I. Ignatsenko, and S. Belmokhtar. Machining lines with multi-spindle stations: a new optimisation problem. *The International Journal of Ingenium*, 2 :153–162, 2005.
- [DM67] E.V. Denardo and L.G. Mitten. Elements of sequential decision processes. *Journal of Industrial Engineering*, 1967.
- [DMM97] M. Dell’Amico, F. Maffioli, and S. Martello. *Annotated bibliographies in combinatorial optimization*. Wiley and Sons, 1997.
- [DP06] A. Dolgui and J.M. Proth. *Les systèmes de production modernes*. Lavoisier, 2006.
- [ES98] E. Erel and S.C. Sarin. A survey of the assembly line balancing procedures. *Production Planning and Control*, 9(5) :414–434, 1998.

- [Fal93] E. Falkenauer. The grouping genetic algorithms: widening the scope of gas. *JORBEL-Belgian Journal of Operations Research, Statistics and Computer Sciences*, 33(1-2) :79–102, 1993.
- [Fal96] E. Falkenauer. A hybrid grouping algorithm for bin packing. *Journal of heuristics*, 2 :5–30, 1996.
- [Fin04] B. Finel. *Structuration de lignes d'usinage : méthodes exactes et heuristiques*. PhD thesis, Université de Metz, 2004.
- [GG89] S Ghosh and R.J. Gagnon. A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, 4 :637–670, 1989.
- [GHR88] S.C. Graves and C. Holmes Redfield. Equipment selection and task assignment for multi-product assembly system design. *International Journal of Flexible Manufacturing Systems*, 1 :31–50, 1988.
- [GJ79] M.R. Garey and D.S. Johnson. *Computers and intractability: A guide to the theory of NP-Completeness*. Bell Laboratories Murray Hill, New Jersey, 1979.
- [GL83] S.C. Graves and B.W. Lamar. An integer programming procedure for assembly design problems. *Operations Research*, 31(3) :522–545, 1983.
- [GL97] F. Glover and M. Laguna. *Tabu Search*. Kluwer academic publishers, Boston, 1997.
- [GN64] A.L. Gutjhar and G.L. Nemhauser. An algorithm for the line balancing problem. *Management Science*, 11 :308–315, 1964.
- [Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Addison Wesley, 1989.
- [GP78] W.V. Gehrlein and J.H. Patterson. Balancing single model assembly lines: comments on a paper by E.M. Dar-El (Mansoor). *AIIE Transactions*, 10 :109–112, 1978.
- [HB61] W.B. Helgeson and D.P. Birnie. Assembly line balancing using the ranking positional weight technique. *Journal of Industrial Engineering*, 12 :394–394, 1961.
- [Hit96] K. Hitomi. *Manufacturing system engineering*. Taylor&Francis, 1996.
- [HKS63] M. Held, R.M. Karp, and R. Shareshian. Assembly line balancing dynamic programming with precedence constraints. *Operations Research*, 11 :442–459, 1963.
- [HM78] S.W. Haider and C.L. Moodie. An investigation of the use of an interactive computer model for balancing paced assembly lines. *Computers and Industrial Engineering*, 2 :83–89, 1978.
- [Hof63] T.R. Hoffman. Assembly line balancing with a precedence matrix. *Management Science*, 9 :551–562, 1963.
- [Hof92] T.R. Hoffman. A hybrid system for assembly line balancing. *Management Science*, 38 :39–47, 1992.

- [HS00] W. Hopp and M.L Spearman. *Factory Physics: Foundations of Manufacturing Management*. Irwin/McGraw-Hill, 2000.
- [ILO] ILOG. *ILOG Solver 6.0: User's Manuel*.
- [Jac56] J.R. Jackson. A computing procedure for a line balancing problem. *Management Science*, 2 :261–271, 1956.
- [Joh88] J.R. Johnson. Optimally balancing large assembly lines with FABLE. *Management Science*, 34 :240–253, 1988.
- [Joh06] Ulf Johansson. Statistiques en bref : industrie, commerce et services. Technical Report KS-NP-06-010-FR-N, Communauté Européenne, 2006.
- [Kar72] R.M. Karp. *Reducibility among combinatorial problems* in Complexity of Computer Computation, pages 85–103. Plenum Press, New York, 1972.
- [KHJ+99] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritchow, H. Van Brussel, and A.G. Ulsoy. Reconfigurable manufacturing systems. *CIRP Annals*, 48(2) :527–598, 1999.
- [KKK96] Y.K. Kim, Y.J. Kim, and Y. Kim. Genetic algorithm for assembly line balancing problem with various objectives. *Computers and Industrial Engineering*, 30(3) :397–409, 1996.
- [KL98] A. K. Kamrani and R. Logendran. *Group technology and cellular manufacturing: methodologies and applications*. Gordon and Breach, 1998.
- [Kle63] M. Klein. On line balancing problem. *Operations Research*, 11 :274–281, 1963.
- [KM72] V. Klee and G.J. Minty. *How good is the simplex algorithm?*. Academic Press, New York, 1972.
- [Kow79] R. Kowalski. Algorithm = logic + control. *Communications of the ACM*, 22(7) :424–436, 1979.
- [Koz92] J. Koza. *Genetic Programming*. Cambridge, Mass., MIT Press, 1992.
- [KQ82] E.P.C. Kao and M. Queyranne. Dynamic programming methods for assembly line balancing. *Operations Research*, 30 :375–390, 1982.
- [LRS06] S.D. Lapierre, A. Ruiz, and P. Soriano. Balancing assembly lines with tabu search. *European Journal of Operational Research*, 168(3) :826–837, 2006.
- [Man67] E.M. Manssor. Improvement on Gutjhar and Nemhauser's algorithm for line balancing problem. *Management Science*, 14 :250–254, 1967.
- [Mas06] S. Masood. Line balancing and simulation of an automated production transfer line. *Assembly Automation*, 26(1) :69–74, 2006.
- [Min83] M. Minoux. *Programmation mathématique : théorie et algorithmes (2 tomes)*. Dunod, Paris, 1983.
- [MT99] A. Matta and T. Tolio. Flexible and productive manufacturing system architecture. In *15th Int. Conf. on Production Research*, Limerick, Ireland, Août 1999.

- [MUK00] M.G. Mehrabi, A.G. Ulsoy, and Y. Koren. Reconfigurable manufacturing systems: key to futur manufacturing. *Journal of Intelligent Manufacturing*, 11 :403–419, 2000.
- [MUKH02] M.G. Mehrabi, A.G. Ulsoy, Y. Koren, and P. Heytler. Trends and perspectives in flexible and reconfigurable manufacturing systems. *Journal of Intelligent Manufacturing*, 13 :135–146, 2002.
- [Mul93] H. Mulenbein. *Evolutionary Algorithms: Theory and Applications*. Wiley, New York, 1993.
- [NW98] G. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience Publication, 1998.
- [OL96] I.H. Osman and G. Laporte. Metaheuristics: a bibliography. *Annals of Operations Research*, 63 :513–623, 1996.
- [Pas05] V. Paschos. *Optimisation combinatoire 1, concepts fondamentaux*. Hermes Science Publusing, 2005.
- [PCI06] PCI. Site web, 2006. <http://www.pci.fr>.
- [PCPV03] H. Pierreval, C. Caux, J.L. Paris, and F. Viguier. Evolutionary approaches to the design and organization of manufacturing systems. *Computers and Industrial Engineering*, 44 :339–364, 2003.
- [PDK83] P.A. Pinto, D.G. Dannenbring, and B.M. Khumawala. Assembly line balancing with processing alternatives: an application. *Management Science*, 29 :817–830, 1983.
- [RDDDB02] B. Rekiek, A. Dolgui, A. Dechambre, and A. Bratcu. State of art of assembly lines design optimisation. *Annual Reviews in Control*, 26(2) :163–174, 2002.
- [Sak84] M. Sakarovitch. *Optimisation combinatoire, méthodes mathématiques et algorithmiques*. Hermann, 1984.
- [SB87] M.J. Saltzman and I. Baybars. A two-process implicit enumeration algorithm for the simple assembly line balancing problem. *European Journal of Operational Research*, 32 :118–129, 1987.
- [SB06] A. Scholl and C. Becker. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3) :666–693, 2006.
- [Sch99] A. Scholl. *Balancing and Sequencing of Assembly Lines*. Physica-Verlag Heidelberg., 1999.
- [SK97] A. Scholl and R. Klein. SALOME-A Bi-directional Branch and Bound Procedure for Assembly Line Balancing. *INFORMS Journal on Computing*, 9 :319–334, 1997.
- [SS77] R.M. Stallman and G.J. Sussman. Forward reasoning and dependency directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, 9(2) :135–196, 1977.

- [Sza97] J. Szadkowski. Critical path concept for multi-tool cutting processes optimization. In *Manufacturing, Modeling, Management and Control IFAC Symposium*, pages 393–398, Vienna, Austria, 1997. Elsevier.
- [Tai95] E.D. Taillard. Comparaison of iterative searches for the quadratic assignment. *Location science*, 3 :87–105, 1995.
- [TPG86] F.B. Talbot, J.H. Paterson, and W.V. Gehrlein. A comparative evaluation of heuristic line balancing techniques. *Management Science*, 32 :430–454, 1986.
- [TYHWK03] L. Tang, D.M. Yip-Hoi, W. Wang, and Y. Koren. Concurrent line-balancing, equipment selection and throughput analysis for multi-part optimal line design. In *CIRP 2nd International Conference on Reconfigurable Manufacturing (Michigan)*, 2003.
- [Wik06] Wikipédia. Site web, 2006. <http://fr.wikipedia.org/wiki/Accueil>.
- [WM86] T.S. Wee and M.J. Magazine. Assembly Line Balancing as Generalized Bin Packing. *Operations Research Letters*, 1 :56–58, 1986.
- [YUA02] A. Yigit, A.G. Ulsoy, and A. Allahverdi. Optimizing modular product design for reconfigurable manufacturing. *Journal of Intelligent Manufacturing*, 13 :309–316, 2002.
- [ZZX02] G.W. Zhang, S.C. Zhang, and Y. S. Xu. Research on flexible transfer line schematic design using hierarchical process planning. *Journal of Materials Processing Technology*, 129 :629–633, 2002.

## École Nationale Supérieure des Mines de Saint Etienne

N° d'ordre : 424 GI

<b>Prénom</b>	Sana
<b>Nom</b>	Belmokhtar
<b>Titre de la thèse</b>	<i>Machining lines with standard equipment : modelization, configuration and optimization</i>
<b>Spécialité</b>	Industrial engineering
<b>Mots clés</b>	machining lines, integer programming, constraint programming
<b>Résumé</b>	<p>The study investigated in this thesis proposes methods to design and optimize the configuration of machining lines from standard multi-spindle units. The problem of designing such lines is defined as the selection of a subset of spindle units from the set of all available units and their assignment to workstations such that all constraints are satisfied and the total cost is minimized. The production rate is insured by imposing a maximal cycle time to not exceed. Moreover, several constraints related to operations as precedence and inclusion constraints are taken into account. Restrictions related to spindle units are also considered such maximal number of units per station and incompatibilities between some spindles. The main part of the thesis was devoted to the design of machining line with parallel activation mode of spindle units at stations. In this case, all spindle-units of the line are activated simultaneously. Several models were proposed to solve this problem, the first one was used in a constraint programming approach for which two lower bound were suggested. Two integer linear programming models were also provided. An experimental study was done in order to compare the performances of the proposed approaches on one hand and to evaluate the different algorithms proposed to reduce the models size on the other hand. In addition, several series of tests have been achieved to study the influence of numerical characteristics of the instances on the performances of the models.</p> <p>Two generalizations have been investigated : the mixed activation mode for spindle units at workstation which is a more complex problem than the parallel one, and the multi-product machining lines which considers several product at the same time.</p>

## École Nationale Supérieure des Mines de Saint Etienne

N° d'ordre : 424 GI

<b>Prénom</b>	Sana
<b>Nom</b>	Belmokhtar
<b>Titre de la thèse</b>	<i>Lignes d'usinage avec équipement standard : modélisation, configuration et optimisation</i>
<b>Spécialité</b>	Génie industriel
<b>Mots clés</b>	Lignes d'usinage, programmation linéaire en nombres entiers, programmation par contraintes
<b>Résumé</b>	<p>Les travaux de cette thèse s'inscrivent dans le cadre du développement d'outils d'aide à la décision pour la configuration des lignes d'usinage modulaires à partir d'équipements standard. Le problème de configuration se pose en termes de sélection d'un sous-ensemble d'unités d'usinage (équipements) à partir d'un ensemble connu et de leur affectation aux postes de travail définissant ainsi la structure (configuration) de la ligne. Le problème revient à trouver la meilleure solution en termes de coût de mise en oeuvre en prenant en compte différents types de contraintes : certaines sont liées à la productivité minimum à assurer, d'autres sont intrinsèques aux opérations et aux équipements. Nous nous sommes essentiellement intéressés au problème de configuration des lignes avec un mode d'activation parallèle des unités d'usinage dans les stations. Dans ce cas, le début d'un cycle est marqué par l'enclenchement simultané de toutes les unités d'usinage de la ligne. Pour ce problème, nous avons proposé trois modèles : un modèle générique que nous avons utilisé dans une approche par programmation par contraintes et deux modèles linéaires en nombres entiers. Nous avons également fourni deux bornes inférieures pour ce problème et proposé plusieurs algorithmes afin de réduire le nombre de variables des modèles et améliorer leurs performances. Une phase expérimentale a été menée pour comparer les performances des approches proposées et étudier l'impact des algorithmes de réduction. De plus, une étude de l'influence de nombreuses caractéristiques numériques des problèmes traités est également rapportée. Nous nous sommes ensuite intéressés à l'extension du plus performant de nos modèles pour deux problèmes plus généraux. La première généralisation du modèle est relative au mode d'activation des unités d'usinage où nous permettons un mode d'activation mixte et non plus seulement parallèle. La seconde généralisation porte sur le nombre de produits usinés en s'intéressant aux lignes multi-produit.</p>