Définition du problème

Dans le présent chapitre, nous décrivons le problème d'équilibrage de lignes d'usinage de type TLBP/B-P (voir figure page 60). Nous rappelons que ces lignes sont composées de stations dont les unités d'usinage sont activées de façon parallèle. Nous expliquons en détails les données ainsi que les différentes contraintes qui caractérisent ce problème en indiquant l'ensemble des notations utilisées. Pour illustrer l'ensemble de ces notions nous proposons un exemple concret qui nous provient de notre partenaire industriel.

4.1 Définition du problème

Le TLBP/B-P est relatif à l'étude de lignes mono-produit considérant un ensemble fixe d'opérations à effectuer en respectant l'ensemble des contraintes. Il est important de signaler que l'unicité du type de produit considéré est conceptuelle, cette hypothèse n'exclut donc pas la considération physique de plusieurs types de pièces à condition que l'ensemble se comporte comme un seul produit. Plus précisément, si un ensemble fixe d'opérations est à effectuer sur la ligne de manière cyclique en respectant un ensemble de contraintes alors le problème peut être classé comme mono-produit. Par exemple, l'utilisation de palette contenant plusieurs types de pièces illustre bien la notion de mono-produit conceptuel. Dans ce cas, la ligne fabrique plusieurs types de pièces qui sont fixées sur une palette servant de support. La palette peut être alors considérée comme un seul produit car il s'agit d'effectuer les mêmes opérations sur chaque palette.

Dans ce problème, le mode d'activation est en parallèle pour les unités d'usinage sur chacune des stations de manière à ce que le début d'un cycle est désigné par l'enclenchement simultanée de toutes les unités de la ligne et la fin du cycle est marqué par leur terminaison.

Pour résoudre le TLBP/B-P, il faut déterminer la structure de la ligne en termes d'unités d'usinage à mettre en place, sur chaque station, et de nombre de stations à ouvrir en tenant compte de plusieurs contraintes. Une des contraintes est la cadence de la ligne, celle-ci est déterminée par le temps de cycle de la ligne. En pratique, une valeur maximale sur le temps de cycle de la ligne est obtenue à partir de la productivité visée. Cette valeur a pour objectif de poser une limite sur le temps de travail pour chacune des stations (celui de la station goulot déterminant le temps de cycle effectif). Ainsi, le temps de cycle effectif ne doit jamais dépasser ce seuil pour pouvoir obtenir la productivité visée.

Par ailleurs, plusieurs autres types de contraintes sont définies, nous distinguons celles qui concernent directement les opérations, par exemple, les contraintes de précédence, de celles qui sont relatives aux unités d'usinage telles que les contraintes d'incompatibilité (exclusion).

Le problème d'optimisation se pose en termes de sélection d'un sous-ensemble d'unités d'usinage, à partir de l'ensemble d'unités disponibles, de telle façon que toutes les opérations soient effectuées, que toutes les contraintes soient respectées et que le coût total soit minimal. Ce coût correspond à la somme des coûts engendrés par la mise en place des stations et le coût des unités d'usinage choisies.

Concrètement, le problème d'optimisation doit apporter des réponses aux interrogations suivantes :

1. Considérant les coûts des unités d'usinage et des stations de travail, quel est le coût minimum à investir dans l'équipement ?
2. Quelle est la configuration associée à ce coût optimal, c'est-à -dire :
   (a) Combien de stations de travail faut-il mettre en place ?
   (b) Combien et quelles sont les unités d'usinage qui doivent être utilisées ?

Pour ce problème, nous développons une approche basée sur la programmation par contraintes (PPC) et deux programmes linéaires en nombres entiers (PLNE) l'un est orienté blocs et l'autre est orienté opérations. Ci-dessous, nous présentons les notations des données que nous utiliserons pour la formulation de ces modèles. Nous introduisons également les ensembles, les collections ainsi que les graphes que nous employons pour représenter les différentes contraintes du problème.

4.1.1 Les données

Les données du problème sont constituées de l'ensemble des opérations à effectuer sur la ligne, de l'ensemble des unités d'usinages disponibles, des coûts et des temps d'exécution des unités d'usinage et du coût d'ouverture d'une station.

L'ensemble des opérations

Toutes les opérations qui doivent être effectuées sur la ligne sont connues. Nous les désignons par l'ensemble $\mathbf{N} = \{1, \dots, n\}$.

L'ensemble des unités d'usinage (blocs)

Nous employons le terme *bloc* d'opérations pour simplifier le vocabulaire, aussi lorsque nous faisons référence à l'appartenance d'une opération à un bloc nous entendons que l'unité d'usinage correspondante exécute l'opération en question.

L'ensemble des blocs disponibles $\mathbf{B}$ est défini comme suit :

$\forall b \in \mathbf{B}$, tel qu'il existe une unité d'usinage effectuant toutes les opérations de b, notées $\mathcal{N}(b)$ tel que $\mathcal{N}(b) \subseteq \mathbf{N}$.

Ainsi, chaque bloc b est défini par l'ensemble d'opérations $\mathcal{N}(b)$ qu'il contient.

Dans le cas d'une conception préliminaire, cet ensemble est construit suite à une étude de disponibilité sur le marché ou à une analyse préalable de faisabilité des têtes d'usinage, tandis que pour la reconfiguration d'une ligne une partie de $\mathbf{B}$ correspondra aux blocs qui sont déjà mis en fonctionnement sur la ligne. Selon les besoins nécessitant la reconfiguration, cet ensemble est éventuellement complété par des blocs qui effectueront des opérations supplémentaires, si de telles opérations doivent être introduites après la reconfiguration. De la même façon, si la ligne, suite à la reconfiguration, ne doit plus effectuer certaines opérations, alors les unités correspondant à ces opérations sont à supprimer de l'ensemble des disponibilités $\mathbf{B}$. L'ensemble $\mathbf{B}$ peut toutefois être mis à jour de façon à intégrer de nouvelles têtes d'usinages qui sont apparues sur le marché ou qui sont conçues par le bureau d'études de l'entreprise elle-même.

En général dans $\mathbf{B}$, il existe plusieurs blocs possibles pour chaque opération, aussi une décision du problème revient à déterminer quel est le bloc à retenir pour chaque opération i de $\mathbf{N}$.

Pour le TLBP/B-P, la sélection d'un bloc entraîne l'exécution de l'intégralité des opérations appartenant à ce bloc. Autrement dit, il n'est pas possible de choisir un bloc pour exécuter seulement un sous-ensemble de ses opérations. Cette hypothèse n'est pas restrictive vu qu'il est toujours possible de considérer des sous-ensembles d'opérations de l'unité correspondante en tant que des éléments de $\mathbf{B}$ à part entière.

Coût des blocs

Le coût du bloc b est donné par q_b qui représente, selon le cas, soit le coût d'acquisition de l'unité d'usinage correspondante, soit une pondération à ajuster par le concepteur lui-même. Par exemple, en accordant un coût relativement bas à une unité, le décideur est en mesure de formuler sa préférence pour le choix de cette unité. Cette pondération permet d'intégrer de nombreux paramètres, entre autres, l'amortissement des unités qui sont déjà en fonctionnement sur la ligne ou le coût de leur exploitation. Nous ne nous attarderons pas sur l'ajustement ou le calcul de ces coûts. Il est toutefois à noter que cette question peut faire, à elle seule, l'objet de plusieurs études.

Coût des stations

Le coût pour la mise en place d'une nouvelle station est noté C. Nous supposons que ce coût est le même pour toute station car il s'agit du coût des éléments de base tels que le bâtis. Nous employons indifféremment les termes *établir* ou *ouvrir* pour l'introduction d'une nouvelle station dans la ligne. À chaque ouverture d'une nouvelle station le coût C est ajouté au coût total. De même, le coût des unités d'usinage qui vont équiper cette station est ajouté au coût total de la ligne.

Temps d'exécution des blocs

Chaque bloc est caractérisé par son coût mais aussi par son temps d'exécution. Ce temps est fourni lors de l'acquisition du bloc et correspond, en pratique, au temps nécessaire pour effectuer toutes les opérations du bloc. Ainsi, pour les unités multi-broches, il peut être estimé, à une constante près[1], au plus grand temps d'exécution de ses opérations [DFG+06b].

4.1.2 Les contraintes

Différentes contraintes sont prises en compte, notamment le temps de cycle qui assure une productivité voulue de la ligne, les contraintes classiques de précédence entre les opérations ainsi que des contraintes d'inclusion pour les opérations et d'exclusion pour les blocs.

Le temps de cycle (productivité)

T_0 représente le temps de cycle maximum ou cadence maximale de la ligne (capacité de production). Du fait que les lignes sont synchrones, le temps de cycle effectif correspond au temps écoulé entre le moment où les stations activent leurs unités et le moment où elles terminent toutes leurs opérations. Autrement dit, le temps de cycle correspond au temps qui sépare la sortie de deux pièces de la ligne. Dans le cas du TLBP/B-P, le temps de cycle est déterminé par l'unité qui a le plus grand temps d'exécution (se référer à la section 1.5.4) auquel il faut ajouter le temps constant qui est nécessaire au mécanisme de transfert pour déplacer une pièce d'une station à une autre. Afin que le temps de cycle soit respecté, il faut que $T \leq T_0$, tel que T est le temps de cycle effectif de la ligne. Pour la cas parallèle cette contrainte peut être traité avant la résolution du problème. En effet, il suffit d'éliminer chacun des blocs ayant un temps d'exécution supérieur à T_0.

[1] La constante permet de prendre en compte l'avance à vide des têtes d'usinage et le temps de transport.

Contraintes de précédence

D^{or} est l'ensemble réunissant les contraintes de précédence liant certaines opérations de $\mathbf{N}$. Ses éléments sont des couples d'opérations tels que tout couple $(i, j) \in D^{or}$ représente une contrainte d'antériorité entre l'opération i et l'opération j. Plus exactement, l'opération j ne peut commencer son exécution avant que l'opération i ne termine la sienne. Notons que ces contraintes peuvent être représentées par un graphe orienté $G^{or} = (\mathbf{N}, D^{or})$. Dans le cas d'une activation parallèle des unités multi-broche, toutes les opérations effectuées sur une même station sont exécutées simultanément. Ainsi, aucun couple $(i, j) \in D^{or}$ ne peut être assigné à une même station. Plus exactement, l'opération i doit être affectée à une station qui précède celle de l'opération j.

Contraintes d'inclusion

D^{in} est la collection des contraintes d'inclusion entre les opérations où chaque élément est un sous-ensemble $d \subset \mathbf{N}$ d'opérations qui doivent être exécutées sur la même station. Pour représenter graphiquement ces contraintes, nous utilisons le graphe non orienté $G^{oi} = (\mathbf{N}, A_1)$ tel que les nœuds représentent les opérations et un arc de A_1 est construit entre tout couple d'opérations appartenant à un sous-ensemble d'inclusion $d \in D^{in}$. Il est intéressant de noter que tout ensemble d appartenant à D^{in} correspond forcément à une clique dans le graphe G^{oi} et que la relation d'inclusion est transitive.

Contraintes d'exclusion

D^{ex} représente la collection des contraintes d'incompatibilité entre blocs, nous utilisons le terme contraintes d'exclusion pour y faire référence. La collection D^{ex} comporte des sous-ensembles $E \subset \mathbf{B}$ tels que chaque sous-ensemble correspond à un ensemble de têtes qui ne peuvent, en aucun cas, être installées sur une même station. Cependant, l'interdiction porte sur l'affectation de l'intégralité des éléments de E et n'affecte pas ses propres sous-ensembles E' (des sous-ensemble $E' \subset E$ peuvent donc être affectées à la même station, c'est-à-dire E est un ensemble minimal). Par exemple, si : $D^{ex} = E = \{b_1, b_2, b_5\}$ alors $\forall E' \subset E$, E' peut être affecté sur une même station si les autres contraintes ne sont pas violées. Ainsi, $\{b_1, b_2\}$, $\{b_1, b_5\}$ ou $\{b_2, b_5\}$ peuvent être assignés dans une même station. Une condition nécessaire pour pouvoir affecter un ensemble quelconque de blocs E sur la même station est que E ne contienne aucun sous-ensemble appartenant à la collection D^{ex}, c'est-à-dire $\nexists E' \subset E$ tel que $E' \in D^{ex}$.

Ces contraintes peuvent être représentées par un graphe $G^{ex} = (\mathbf{B}, A_2)$ où un arc appartenant à A_2 relie deux blocs incompatibles. Il est également à noter que la relation d'incompatibilité n'est pas transitive, ainsi lorsque un bloc b est incompatible avec un bloc b' qui lui-même n'est pas compatible avec b'', dans ce cas b et b'' ne sont pas forcément incompatibles à moins que le couple (b, b'') appartienne à D^{ex}.

Limitation sur le nombre de stations

m_0 est le nombre maximum de stations que peut comporter la ligne.

Limitation sur le nombre de blocs par station

n_0 est le nombre maximum d'unités d'usinage qui peuvent être installées sur une station, ce qui correspond souvent au nombre d'emplacements d'axes possibles.

4.1.3 Solutions réalisables

Une solution du problème correspond à une configuration réalisable, si et seulement si les conditions suivantes sont remplies :

- Chacune des opérations de $\mathbf{N}$ doit être exécutée exactement une seule fois. C'est-à-dire qu'aucune opération ne doit être omise ni dupliquée. Donc, aucune intersection entre les blocs d'opérations n'est autorisée car une solution avec deux blocs non disjoints correspond à une configuration avec des unités d'usinage effectuant les mêmes opérations.
- Le temps de travail de chaque station ne doit pas dépasser le temps de cycle maximum T_0.
- Toutes les autres contraintes sont également respectées : inclusion, exclusion, précédence et limitations sur le nombre de stations et sur le nombre d'unités d'usinage par station.

Parmi toutes les configurations réalisables celle dont la somme des coûts des unités d'usinage retenues pour équiper les stations et des coûts engendrés par la mise en place des stations est minimale est la solution optimale. Il est cependant courant d'avoir plusieurs solutions optimales, celles-ci ont le même coût mais pas forcément la même structure. Ces structures peuvent être différentes car elles ne contiennent pas les mêmes blocs ou parce que l'affectation des blocs aux stations est différente.

4.2 Un exemple numérique

Afin d'illustrer toutes les données et les contraintes décrites précédemment, nous présentons un exemple industriel concret qui s'est posé lors de la conception d'une ligne d'usinage pour la fabrication de la pièce décrite dans la figure 4.1.

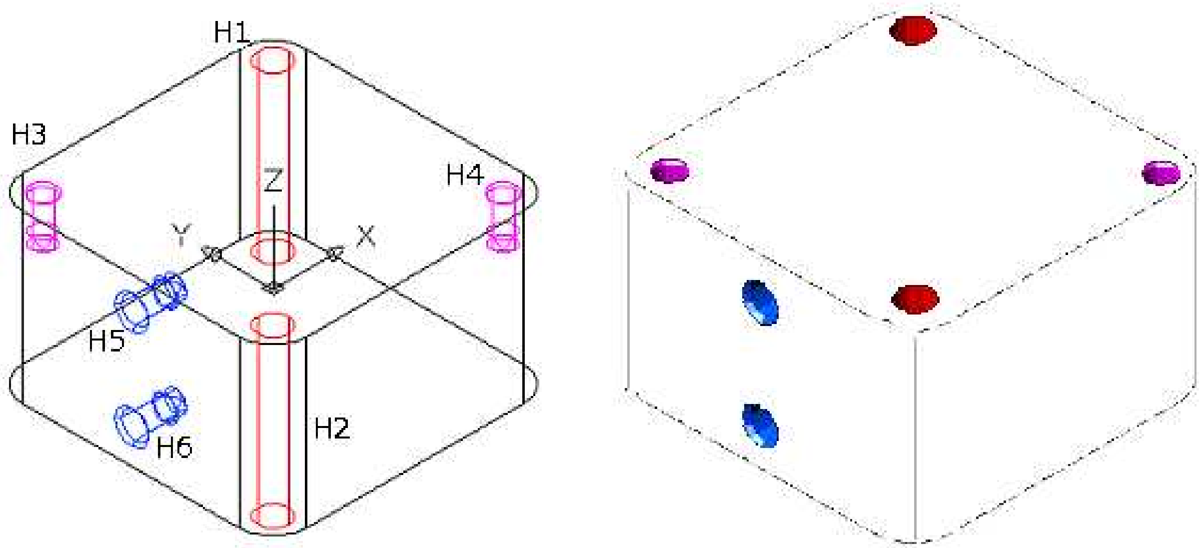

FIG. 4.1 – Deux vues de la pièce

Pour des raisons de confidentialité, les coûts rapportés ont été modifiés, mais toutes les autres données et contraintes sont authentiques.

Pour la fabrication de la pièce, 14 opérations sont indispensables, elles sont décrites dans le tableau 4.1. Pour ce faire, on dispose de 21 têtes d'usinage qui sont décrites dans le tableau 4.2. Le coût pour mettre en place une station est 15500 unités monétaires. De plus, le nombre maximum de blocs par stations $n_0 = 2$ et le nombre maximum de stations sur la ligne $m_0 = 8$.

opération	description	paramètres	temps opératoire
1	Perçage de H1	⌀ 14	169,60
2	Perçage de H2	⌀ 14	169,60
3	Perçage de H3	⌀ 9,8	48,60
4	Chambrage du chanfrein H3	⌀ 12	7,30
5	Alésage H3	⌀ 10	13,71
6	Perçage de H4	⌀ 9,8	48,60
7	Chambrage du chanfrein H4	⌀ 12	7,30
8	Alésage H4	⌀ 10	13,71
9	Perçage H5	⌀ 10,2	52,42
10	Chambrage du chanfrein H5	⌀ 13,2	8,04
11	Filetage H5	M12-6G	3,21
12	Perçage H6	⌀ 10,2	52,42
13	Chambrage du chanfrein H6	⌀ 13,2	8,04
14	Filetage H6	M12-6G	3,21

TAB. 4.1 – L'ensemble **N** des opérations

blocs	opérations	coût	temps
b_1	1,3,6	12200	169,6
b_2	2,4,7	10100	169,6
b_3	1,2,3,6	14400	169,6
b_4	4,7	7950	7,30
b_5	9,12	10200	52,42
b_6	10,13	7450	8,04
b_7	5,8	9000	13,71
b_8	11	9750	3,21
b_9	14	9750	3,21
b_{10}	2,3,6	12300	169,6
b_{11}	1,4,7	10150	169,6

blocs	opérations	coût	temps
b_{12}	1,2,4,7	14150	169,6
b_{13}	3,6	10000	48,60
b_{14}	1,6	10200	169,6
b_{15}	1	10900	169,6
b_{16}	2	10900	169,6
b_{17}	1,2	13800	169,6
b_{18}	3	9000	48,60
b_{19}	6	9000	48,60
b_{20}	5	8500	13,71
b_{21}	8	8500	13,71

TAB. 4.2 – L'ensemble **B** des unités d'usinage

L'ensemble des **contraintes de précédence** sont données comme suit :

$D^{or} = \{(3,4),(4,5),(6,7),(7,8),(9,10),(1,11),(2,11),(5,11),(8,11),(10,11),(13,11),$
$(12,13),(1,14),(2,14),(5,14),(8,14),(10,14),(13,14)\}$.

La figure 4.2 montre le graphe $G^{or} = (\mathbf{N}, D^{or})$ représentant les contraintes de précédence.

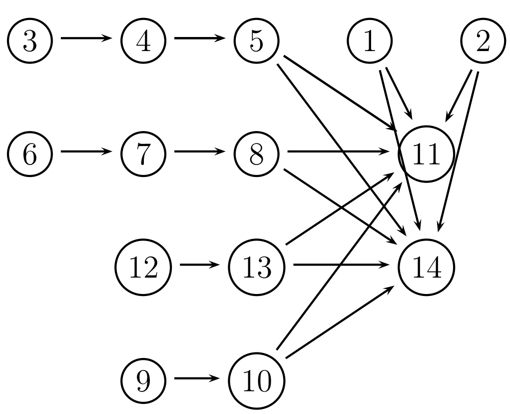

FIG. 4.2 – Le graphe de précédence G^{or}

Les **contraintes d'inclusion** sont décrites comme suit :

$D^{in} = \big\{\{1,2\},\{3,6\}\big\}$.

Le graphe G^{in} permet de représenter les contraintes d'inclusion comme indiquée dans la figure 4.3.

L'ensemble des **contraintes d'exclusion** est donné par la collection D^{ex} qui ne contient que des couples de blocs incompatibles. Toutefois, le modèle prend en compte le cas général où plus de deux blocs peuvent être incompatibles (le graphe correspondant est donné par

4.4).

FIG. 4.3 – Le graphe d'inclusion G^{in}

$D^{ex} = \big\{\{5,8\},\{6,8\},\{9,8\},\{1,7\},\{2,7\},\{3,7\}, \{4,7\},\{5,7\},\{6,7\},\{10,7\},\{11,7\},$
$\{13,7\},\{14,7\} ,\{15,7\},\{16,7\},\{17,7\}, \{18,7\},\{19,7\},\{20,1\},\{20,2\},\{20,3\},\{20,4\},$
$\{20,6\},\{20,10\},\{20,11\},\{20,12\} \{20,13\},\{20,14\},\{20,15\},\{20,16\},\{20,17\},$
$\{20,18\},\{20,19\},\{21,1\},\{21,2\},\{21,3\},\{21,4\},\{21,5\},\{21,6\},\{21,10\},\{21,11\},$
$\{21,12\},\{21,13\},\{21,14\},\{21,15\},\{21,16\},\{21,17\},\{21,18\},\{21,19\},\{20,5\}\big\}$

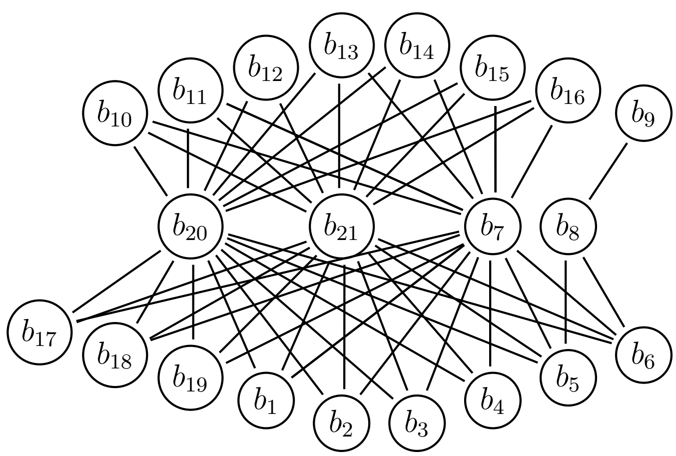

FIG. 4.4 – Le graphe d'exclusion G^{ex}

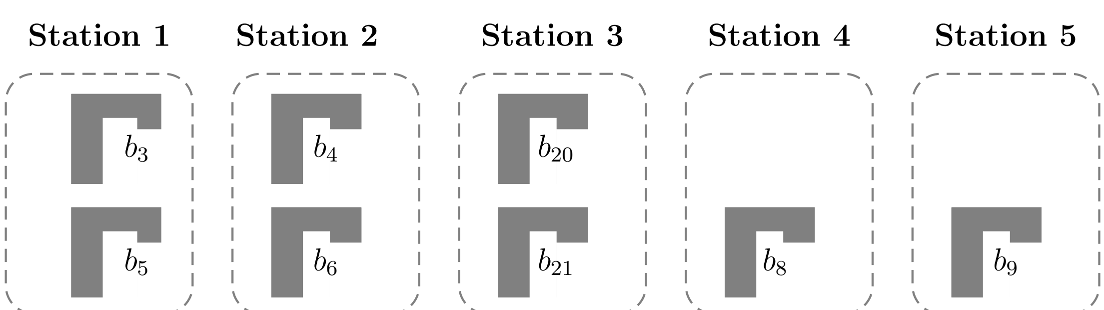

FIG. 4.5 – Une affectation réalisable des blocs aux stations

Une solution réalisable est donnée dans la figure 4.5 ayant un temps de cycle de 169,6. Elle est composée de 5 stations : la première est équipée des deux blocs b_3 et b_5, la seconde contient b_4 et b_6, la troisième est composée de b_{20} et b_{21}, la quatrième comporte b_8 et la dernière b_9. Par conséquent, le coût total est de 154000. Il est intéressant de constater que les quatre premières stations sont établies en raison des contraintes de précédence. Par exemple, le bloc b_{20} et b_8 ne peuvent pas partager la même station car b_8 contient une opération successeur de b_{20}. De même b_9 est successeur de b_{20} et b_{21}. Une dernière station doit être ouverte pour contenir b_9 car d'une part b_9 est successeur de b_{20} et b_{21}et d'autre part b_9 ne peut partager une station avec b_8 en raison de l'existence d'une incompatibilité qui les lie (voir l'arc b_8,b_9 dans le graphe G^{ex}).

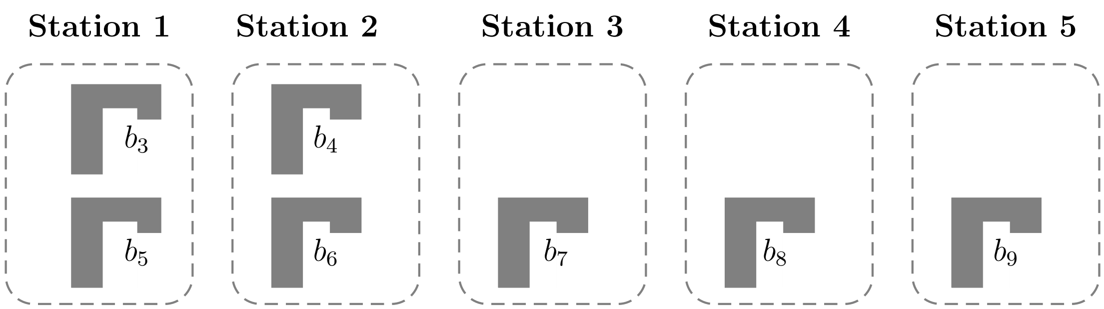

FIG. 4.6 – La meilleure affectation des blocs aux stations

La solution optimale est montrée dans la figure 4.6 où le choix du bloc b_7 se substitue aux blocs b_{20} et b_{21} diminuant le coût total à 146000.

Il est important de souligner que pour respecter le temps de cycle dans le cas du TLBP-B/P il suffit de filter l'ensemble **B** en écartant les blocs ayant un temps d'exécution strictement supérieur au temps de cycle maximum.

4.3 Pré-traitements

Dans cette section, nous proposons deux types de pré-traitements pour le modèle présenté ci-dessus, en premier lieu nous présentons une analyse de la consistance des données qui va permettre d'éliminer certains éléments de l'ensemble **B** et des contraintes d'inclusion et d'exclusion. Par la suite, nous proposons d'effectuer quelques tests (avant d'entamer le processus de résolution) afin de détecter des infaisabilités, évitant ainsi de lancer une recherche assurément infructueuse [BDGL05a, BDGL06].

4.3.1 Consistance

L'étude de la consistance des données a pour objectif de réduire l'espace de recherche en soustrayant les éléments qui engendrent des tentatives infructueuses. Plus précisément, nous appliquons des règles d'inconsistance pour éviter de faire participer des éléments provoquant systématiquement une violation d'une des contraintes. Cette étude sert à diminuer l'effort de recherche en gardant indemne l'ensemble des solutions réalisables et en éliminant le maximum d'éléments inconsistants. Nous proposons d'appliquer des procédures basées sur l'étude de la consistance des données d'une part et sur la dominance de certaines contraintes d'autre part. Afin de modérer les efforts, nous préférons employer ces techniques en tant que pré-traitements, c'est-à-dire avant même d'entamer la résolution du problème au lieu de les effectuer à chaque nœud de l'arbre de résolution.

Nous décrivons d'abord les pré-traitements que nous proposons pour réduire l'ensemble des blocs disponibles $\mathbf{B}$ en expliquant, dans chaque cas, les raisons qui ont permis de conclure sur l'inconsistance des blocs à écarter. Ensuite, nous appliquons quelques règles pour vérifier la faisabilité du problème.

Inconsistance des blocs

1. Si le temps d'exécution du bloc b est strictement supérieur au temps de cycle T_0 alors ce bloc est inconsistant et peut être supprimé de l'ensemble des blocs disponibles $\mathbf{B}$.
   En effet, comme l'activation des blocs se fait en parallèle, il en résulte que le temps de cycle est déterminé par le bloc qui a le plus grand temps d'exécution. Par conséquent, tout bloc avec un temps supérieur à T_0 n'assurerait pas la productivité minimale requise et sa sélection conduirait à la violation du temps de cycle. Ceci est une condition suffisante pour éliminer ce bloc afin d'assurer la consistance de $\mathbf{B}$.
   Il en résulte que les contraintes d'exclusion qui concernent ce bloc deviennent caduques car elles font intervenir un bloc non existant, elles sont donc à supprimer. Toutefois, les contraintes qui font intervenir les opérations effectuées par ce bloc doivent, quant à elles, être maintenues[2], car ces contraintes sont propres aux opérations qui peuvent appartenir, par ailleurs, à d'autres blocs[3]. Ainsi, on écrit :

   $\forall b \in \mathbf{B}$,
   Si
   $t_b > T_0$
   alors
   $\mathbf{B} := \mathbf{B} - \{b\}$
2. Dés lors les opérations d'un même bloc sont effectuées de façon simultanée par la même tête d'usinage multi-broche tout bloc contenant des opérations reliées par une

[2] Le problème engendré, dans le cas de la suppression des contraintes liées aux opérations du bloc b, serait une relaxation du problème original car des contraintes intrinsèques aux opérations auraient été omises. En terme d'espace de solutions, les problèmes sont différents.

[3] Si le problème admet une solution réalisable, il est forcé que ces opérations appartiennent à d'autres blocs.

relation de précédence peut être supprimé de **B**. En effet, si un tel bloc est considéré la contrainte d'antériorité ne pourra être respectée. Plus formellement, on écrit :

$\forall b \in \mathbf{B}$,
Si
$\exists i, j \in \mathcal{N}(b)$ tel que : $(i, j) \in D^{or}$
alors
$\mathbf{B} := \mathbf{B} - \{b\}$

3. Si un bloc b contient au moins une opération en commun avec un ensemble $d \in D^{in}$ alors la sélection de ce bloc implique l'exécution des opérations de $\mathcal{N}(b) \bigcup d$ dans la même station. En effet, les opérations de $d \in D^{in}$ sont indissociables et les opérations appartenant à un même bloc le sont également. Si l'intersection de $\mathcal{N}(b)$ et d est non vide alors c'est l'intégralité de l'union qui devient indissociable et doit, donc, être exécutée sur la même station. Par conséquent, le choix de b ne peut amener à une solution faisable lorsqu'une relation de précédence existe entre certaines opérations de cette union. Il en résulte, que tout bloc ayant une intersection avec un $d \in D^{in}$ et contenant une opération précédant ou succédant à une opération dans d est inconsistant. Plus formellement :

$\forall b \in \mathbf{B}, \forall d \in D^{in}$,
Si
$\exists i \in \mathcal{N}(b), \exists j \in d \mid \mathcal{N}(b) \cap d \neq \varnothing \wedge ((i, j) \in D^{or} \vee (j, i) \in D^{or})$
alors
$\mathbf{B} := \mathbf{B} - \{b\}$

4. Lorsque un bloc b contient au moins deux opérations et que ces dernières sont liées par des contraintes de précédence aux opérations d'un sous-ensemble d'inclusion ; dans ce cas, si le sens des deux relations d'antériorité est contradictoire alors, quelle que soit l'affectation du bloc b, les deux contraintes ne pourraient être satisfaites simultanément. Plus formellement, ce cas s'écrit de la façon suivante :

$\forall b \in \mathbf{B}, \forall d \in D^{in}$
Si
$\exists k, l \in d, \exists i, j \in \mathcal{N}(b) \mid (i, k) \in D^{or} \wedge (l, j) \in D^{or}$
alors
$\mathbf{B} := \mathbf{B} - \{b\}$

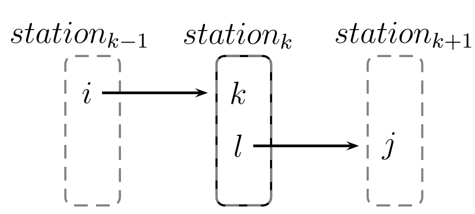

FIG. 4.7 – Inconsistance des blocs contenant i et j

En effet, dans ce cas, les opérations i et j doivent être exécutées sur deux stations distinctes et séparées par au moins une autre station, car les opérations k et l sont indissociables vu qu'elles appartiennent au même sous-ensemble d'inclusion. En raison des contraintes de précédence (i,k) et (l,j), le bloc b ne peut appartenir à une solution sans enfreindre une de ces contraintes. Le schéma donné dans la figure 4.7 montre clairement qu'il faut au minimum une station entre celle où l'opération i est affectée et celle de j pour respecter les contraintes de précédence.

5. Lorsqu'il n'existe aucun sous-ensemble $B' \subset \mathbf{B}$ contenant le bloc b tel que toutes les conditions suivantes soient vérifiées :
   a) tous les blocs de B' sont mutuellement disjoints ;
   b) l'union des opérations des blocs de B' coincide avec l'ensemble $\mathbf{N}$;
   c) la cardinalité de B' ne dépasse pas le nombre maximale des blocs qui peuvent être assignés sur la ligne : $m_0 n_0$.

   Plus précisément, le bloc b est éliminé, si $b \in B'$ et :

   $\exists i \in \mathbf{N} - \{\mathcal{N}(b)\}$ tel que $\forall b' \in \mathbf{B} - B', i \in \mathcal{N}(b') \Rightarrow \mathcal{N}(b) \cap \mathcal{N}(b') \neq \varnothing$

   En d'autres termes, si le bloc b est choisi l'opération i ne pourra pas être effectuée car tout bloc b' contenant l'opération i est mutuellement exclusif avec le bloc b car ils ont des opérations communes. Par conséquent, le bloc b est inconsistant et peut être supprimé.

Dans le premiers cas c'est le temps d'exécution du bloc qui le rend inconsistant. Dans les trois cas suivants, nous supprimons des blocs en raison des violations des relations d'antécédence. Quant au dernier cas, il concerne les blocs dont le choix engendre l'absence d'une ou de plusieurs opérations. Dans ce qui suit, nous présentons d'autres règles de pré-traitement.

Dominance de contraintes

Une analyse de dominance de certaines contraintes par rapport à d'autres permet de détecter des contraintes moins fortes ou redondantes. Ainsi, les contraintes dominantes sont conservées alors que les contraintes dominées peuvent être omises car elles sont en quelque sorte contenues dans les premières.

Par exemple, un sous-ensemble E peut être supprimé de l'ensemble D^{ex} lorsque sa cardinalité est supérieure au nombre maximum de blocs par station, *i.e.* $|E| > n_0$. En effet, le nombre de blocs qui peuvent être affectés à la même station étant limité à n_0, il n'est plus nécessaire de conserver l'interdiction exprimée par E car cette contrainte devient redondante.

Par ailleurs, le pré-traitement concernant la collection D^{ex} peut être affiné davantage en supprimant les sous-ensembles $E \in D^{ex}$ qui correspondent au cas suivant :

$$\exists b, b' \in E \text{ tel que } \mathcal{N}(b) \cap \mathcal{N}(b') \neq \varnothing.$$

Dans ce cas, du fait que les blocs b et b' ont des opérations communes, ils deviennent mutuellement exclusifs. Ainsi, les contraintes interdisant leur affectation à la même station deviennent superflues car elles seront déjà prises en compte par les inéquations interdisant l'affectation de plus d'un bloc parmi ceux qui exécutent les mêmes opérations (voir (5.19)).

Réduction de la collection d'inclusion D^{in}

Les sous-ensembles de D^{in} dont l'intersection est non vide peuvent être fusionnés en un seul sous-ensemble. Par définition, un sous-ensemble de D^{in} est un sous-ensemble contenant les opérations de $\mathbf{N}$ qui doivent impérativement être exécutées sur la même station. La relation d'inclusion est transitive c'est-à-dire s'il existe des opérations qui doivent être exécutées avec une certaine opération et que celle-ci elle-même doit être effectuée avec d'autres alors c'est l'union de toutes ces opérations qui devient indissociable.

À présent, nous proposons quelques tests nécessaires mais non suffisants à la faisabilité des instances. Ces tests ont pour objectif de détecter les instances irréalisables avant même d'entamer leur résolution.

4.3.2 Vérification de la faisabilité

Afin qu'une instance du problème admette une solution réalisable il est nécessaire que les conditions suivantes soient vérifiées :

1. Il faut que l'union des blocs de $\mathbf{B}$ coincide avec l'ensemble des opérations, c'est-à-dire il doit y avoir au moins un bloc pour chaque opération, donc :

$$\bigcup_{b \in \mathbf{B}} \mathcal{N}(b) = \mathbf{N}$$

2. Le graphe de précédence G^{or} doit être acyclique. C'est-à-dire que le graphe ne doit pas comporter d'arc liant un sommet j avec un sommet i s'il existe un chemin allant de i vers j.

3. Pour tout sous-ensemble d'inclusion $d \in D^{in}$, il doit exister un sous-ensemble $B' \subseteq \mathbf{B}$ de blocs mutuellement disjoints tel que :

   (a) le nombre de blocs de B' ne dépasse pas le nombre maximum de blocs par station soit : $|B'| \leq n_0$;

   (b) l'intégralité des opérations appartenant à d doit être contenue dans les blocs de B', soit : $d \subseteq \bigcup_{b \in B'} \mathcal{N}(b)$;

   (c) l'ensemble des blocs de B' ne doit pas contenir de blocs incompatibles, c'est-à-dire : $\forall E \subseteq B', E \notin D^{ex}$;

   (d) aucune précédence ne doit lier deux opérations appartenant aux blocs de B', soit : $\forall i, j \in \bigcup_{b \in B'} \mathcal{N}(b), (i, j) \notin D^{or}$.

4.4 Conclusion

Dans le présent chapitre nous avons introduit le problème d'optimisation TLBP/B-P, en précisant les notations ainsi que les données et les contraintes. Nous avons également rapporté un exemple concret, qui nous a été transmis par un de nos partenaires industriels, afin d'illustrer les différentes données et contraintes du problème et de mieux cerner ses caractéristiques.

Nous avons également proposé plusieurs pré-traitements pour réduire la taille des données en éliminant quelques éléments inconsistants. De plus, nous avons suggéré quelques tests qui permettent de détecter à l'avance certaines instances infaisables.

Nous proposons dans le chapitre suivant trois formulations pour modéliser et résoudre ce problème. La première est basée sur la programmation par contraintes et les deux autres sont des programmes linéaires en nombres entiers. Nous proposons également deux bornes inférieures, l'une obtenue par relaxation linéaire d'un des modèles proposés et la seconde par relaxation à un problème particulier de set partitioning.