ProjetGOPP

Pierre Pinet, Hanen Cherif

Decembre 2023

1 **Preuve de NP-difficulté** 2

2 **Formulations PLNE** 3
- 2.1 Modèle avec périodes de temps . . . 3
- 2.2 Modèle sans périodes de temps . . . 5
  - 2.2.1 Améliorations possibles . . . 5

3 **Décompostition de Benders** 6
- 3.1 Benders Classique . . . 6
- 3.2 Benders Combinatoire . . . 7

4 **Extensions du problème** 10
- 4.1 Plusieurs machines . . . 10
  - 4.1.1 Modèle sans période du temps . . . 10
  - 4.1.2 modèle avec période du temps . . . 11
- 4.2 Consommation d'électricité . . . 12
  - 4.2.1 Modèle avec périodes de temps . . . 12
  - 4.2.2 Modèle Sans Périodes de Temps . . . 13

5 **Résolution heuristique** 14

6 **Expérimentations numériques** 14
- 6.1 Capacité à trouver une solution . . . 14
- 6.2 Qualité des solutions . . . 14
- 6.3 Conclusion . . . 14

1 Preuve de NP-difficulté

Pour monter que notre problème (PMSLP) est NP-difficile on va faire une réduction polynomiale depuis le problème d'ordonnancement $1||\sum T_j$ qui est NP-difficile.

Soit $\mathbb{I}_a$ une instance du problème $1||\sum T_j$. Les données de cette instance sont:

1. un ensemble de tâches I
2. q_i les durées des tâches $i \in I$
3. b_i les dates d'échéance des tâches $i \in I$
4. une valeur cible α

On va construire un instance $\mathbb{I}_b$ du problème PMSLP à partir de $\mathbb{I}_a$. Les données de cette instance sont:

1. un ensemble de tâches $J = I$.
2. un singleton contenant le seul site $K = \{k\}$.
3. un nombre de machines à installer $m = 1$.
4. un unique coût d'installation $c_k = 0$ pour l'unique site.
5. $p_j = q_j$ les durées des tâches $j \in J$.
6. $d_j = b_j$ les dates d'échéance des tâches $j \in J$.
7. $D_{jk} = 0$ les distances des tâches $j \in J$ au site k.
8. une vitesse de transport $u_{jk} = 1$ pour toutes les tâches.
9. un coût de transport $f_{jk} = 0$ pour toutes les tâches.
10. une valeur cible $\beta = \alpha$.

On voit trivialement que la réduction se fait bien en temps polynomial en fonction la taille de l'instance $\mathbb{I}_a$.

On rappelle qu'une instance est positive si le problème de décision associé est vérifié, entre autres qu'il existe une solution de coût $\leq \alpha$ / $\leq \beta$ pour l'instance.

On va montrer que $\mathbb{I}_a$ positive est équivalent à $\mathbb{I}_b$ positive.

On suppose qu'il existe une solution O_a de coût $\leq \alpha$ pour $\mathbb{I}_a$. On définit la solution O_b pour $\mathbb{I}_b$ où on installe la seule machine sur l'unique site k et où on utilise l'ordonnancement de O_a sur cette machine. Les coûts de d'installation et de transport sont nécessairement nuls et les retards sont les même donc les solutions O_a et O_b ont la même valeur. On peut donc exhiber la solution O_b pour prouver que l'instance $\mathbb{I}_b$ est positive.

On peut prouver l'autre sens de l'équivalence (:$\mathbb{I}_b$ positive $\implies$ $\mathbb{I}_a$ positive) trivialement en utilisant l'idée précédente dans l'autre sens. On exhibe une solution de valeur $\leq \beta$ pour $\mathbb{I}_b$ et on utilise le même ordonnancement pour $\mathbb{I}_a$.

2 Formulations PLNE

Pour les données on reprend les même notations que celles de l'énoncé.

2.1 Modèle avec périodes de temps

On définit T l'ensemble des périodes de temps. $|T| = \lceil \max_{j,k}\{r_{jk}\} \rceil + \sum_{j \in J} p_j$
On peut aussi remplacer le nombre de périodes par la plus grande date de fin d'une solution obtenue précédemment.

Pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$y_k = \begin{cases} 1 & \text{si on installe une machine sur le site } k \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ et pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$x_{jk} = \begin{cases} 1 & \text{si la tâche } j \text{ est affectée au site } k \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ et pour chaque période $t \in T$ on a une variable binaire indicatrice:

$$s_{jt} = \begin{cases} 1 & \text{si la tâche } j \text{ commence à la période } t \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$, pour chaque site $k \in K$ et pour chaque période $t \in T$ on a une variable binaire indicatrice:

$$a_{jkt} = \begin{cases} 1 & \text{si la tâche } j \text{ est en cours d'éxécution sur le site } k \text{ à la période t} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ on a une variable réelle:

$$T_j \in \mathbb{R}^+ \text{ le retard de la tâche } j$$

Le modèle est le suivant:

$$
\begin{aligned}
\min \quad & \lambda_1(\sum_{k\in K} y_k c_k) + \lambda_2(\sum_{j\in J}\sum_{k\in K} x_{jk}e_{jk}) + \lambda_3(\sum_{j\in J} T_j) && (1)\\
\text{s.c.} & \sum_{k\in K} x_{jk} = 1 \quad \forall j \in J && (2)\\
& \sum_{k\in K} y_k = m && (3)\\
& x_{jk} \leq y_k \quad \forall j \in J, \forall k \in K && (4)\\
& \sum_{t\in T} s_{jt} = 1 \quad \forall j \in J && (5)\\
& a_{jkt} \geq \sum_{u\in[t-p_j+1;t]} s_{ju} + x_{jk} - 1 \quad \forall j \in J, \forall k \in K, \forall t \in T && (6)\\
& \sum_{j\in J} a_{jkt} \leq 1 \quad \forall k \in K, \forall t \in T && (7)\\
& \sum_{t\in T} s_{jt}.t \geq \sum_{k\in K} x_{jk}r_{jk} \quad \forall j \in J && (8)\\
& T_j \geq \sum_{t\in T} s_{jt}.t + p_j - d_j + \sum_{k\in K} x_{jk}r_{jk} \quad \forall j \in J && (9)\\
& y_k \in \{0;1\} \quad \forall k \in K && (10)\\
& x_{jk} \in \{0;1\} \quad \forall j \in J, \forall k \in K && (11)\\
& s_{jt} \in \{0;1\} \quad \forall j \in J, \forall t \in T && (12)\\
& a_{jkt} \in \{0;1\} \quad \forall j \in J, \forall k \in K, \forall t \in T && (13)\\
& T_j \geq 0 \quad \forall j \in J && (14)\\
& && (15)
\end{aligned}
$$

L'objectif (1) minimise la somme des coûts d'installation, de transport et de retard, chacun respectivement pondéré par un poids.

La contrainte (2) impose que chaque tâche doit être affectée à un site.

La contrainte (3) impose que m machines doivent êtres installées.

La contrainte (4) impose qu'une tâche j ne peut être affectée à un site k que si une machine y a été installée.

La contrainte (5) impose qu'une tâche j doit commencer à une période.

La contrainte (6) impose qu'une tâche j est en cours d'exécution sur le site k à la période t si elle a commencé il y a moins de p_j périodes **et** qu'elle est affectée sur le site k.

La contrainte (7) impose que la machine sur un site k ne peut pas exécuter plus d'une tâche à la période t.

La contrainte (8) impose qu'une tâche j n'est disponible qu'après la fin du trajet vers le site auquel elle est affectée.

La contrainte (9) fixe la valeur du retard d'une tâche j, le temps de retour r_{jk} est pris en compte (avec k le site où est affectée la tâche).

Comme les tâches doivent démarrer sur des périodes entières l'optimum de sera légèrement plus grand que celui d'un modèle sans périodes de temps (on peut borner ce décalage par le nombre de tâches)

2.2 Modèle sans périodes de temps

Pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$y_k = \begin{cases} 1 & \text{si on installe une machine sur le site } k \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ et pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$x_{jk} = \begin{cases} 1 & \text{si la tâche } j \text{ est affectée au site } k \\ 0 & \text{sinon} \end{cases}$$

Pour chaque paire orientée de tâche différentes i et j, $i \neq j$ on on a une variable binaire indicatrice:

$$z_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est planifiée avant la tâche } j \text{ sur un même site} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ on a une paire de variables réelles:

$$s_j \in \mathbb{R}^+ \text{ la date de début de la tâche } j$$

$$T_j \in \mathbb{R}^+ \text{ le retard de la tâche } j$$

Le modèle est le suivant:

$$\begin{aligned}
\min \quad & \lambda_1(\sum_{k \in K} y_k c_k) + \lambda_2(\sum_{j \in J}\sum_{k \in K} x_{jk} e_{jk}) + \lambda_3(\sum_{j \in J} T_j) && (16)\\
\text{s.c.} & \sum_{k \in K} x_{jk} = 1 \quad \forall j \in J && (17)\\
& \sum_{k \in K} y_k = m && (18)\\
& x_{jk} \leq y_k \quad \forall j \in J, \forall k \in K && (19)\\
& z_{ij} + z_{ji} \geq x_{ik} + x_{jk} - 1 \quad \forall i \in J, \forall j \in J, i \neq j, \forall k \in K && (20)\\
& s_j \geq \sum_{k \in K} x_{jk} r_{jk} \quad \forall j \in J && (21)\\
& s_i + p_i \leq s_j + M(1 - z_{ij}) \quad \forall i \in J, \forall j \in J, i \neq j && (22)\\
& T_j \geq s_j + p_j - d_j + \sum_{k \in K} x_{jk} r_{jk} \quad \forall j \in J && (23)\\
& y_k \in \{0;1\} \quad \forall k \in K && (24)\\
& x_{jk} \in \{0;1\} \quad \forall j \in J, \forall k \in K && (25)\\
& z_{ij} \in \{0;1\} \quad \forall i \in J, \forall j \in J, i \neq j && (26)\\
& s_j \geq 0 \quad \forall j \in J && (27)\\
& T_j \geq 0 \quad \forall j \in J && (28)\\
& && (29)
\end{aligned}$$

L'objectif (16) minimise la somme des coûts d'installation, de transport et de retard, chacun respectivement pondéré par un poids.

La contrainte (17) impose que chaque tâche doit être faite.

La contrainte (18) impose que m machines doivent êtres installées.

La contrainte (19) impose qu'une tâche j ne peut être affectée à un site k que si une machine y a été installée.

La contrainte (20) impose que si deux tâches différentes i et j sont planifiées sur la même machine k alors l'une doit être planifiée avant l'autre.

La contrainte (21) impose qu'une tâche j n'est disponible qu'après la fin du trajet vers le site auquel elle est affectée.

La contrainte (22) impose que deux tâches ne peuvent pas être faites en même temps sur la même machine. Ici M est la plus grande durée possible entre le début d'une tâche et la fin d'une autre: $\max_{j,k}\{r_{jk}\} + \sum_{j \in J} p_j$

La contrainte (23) fixe la valeur du retard d'une tâche j, le temps de retour r_{jk} est pris en compte (avec k le site où est affectée la tâche).

2.2.1 Améliorations possibles

Pour renforcer la relaxation du modèle on peut ajouter des contraintes comme la transitivité sur les variables z.

Le big M pourrait être remplacé par des M_j plus petits car la date début minimale de la tâche j peut être soustraite.

Les variables s_j peuvent probablement être supprimées en les remplaçant par une expression qui ressemble à $\sum_{i \in J, i \neq j} z_{ij} p_i$, (cette expression est incomplète à cause des dates de disponibilités).

3 Décompostition de Benders

3.1 Benders Classique

a) Une fois que $\overline{K}$ et les $\overline{J_k}$ sont fixés le problème restant se décompose en m sous problèmes indépendants où on doit choisir les dates début s_j de chaque tâche $j \in \overline{J_k}$ de telle sorte à minimiser la somme des retards de ces tâches tout en respectant l'ordre donné par $\overline{J_k}$.

Ce problème est le problème d'ordonnancement $1|r_j;prec|\sum T_j$. (il faut remplacer d_j par $d_j - rjk$ pour avoir la correspondance)
On pose $n =: |\overline{J_k}|$

Ici comme l'ordre des tâches est une donnée on peut résoudre ce problème en fixant récursivement les dates de début:
$j(1) = r_{jk}$
$j(l) = \max\{r_{jk}; j(l-1) + p_j\}, l \in [2;n]$

b) Pour un sous problème avec k fixé on a:

$$\min \quad \sum_{j \in \overline{J_k}} T_j \tag{30}$$
$$s_j \geq r_{jk} \quad \forall j \in \overline{J_k} \tag{31}$$
$$s_{j(i+1)} \geq s_{j(i)} + p_{j(i)} \quad \forall i \in [1;n-1] \tag{32}$$
$$T_j \geq s_j + p_j - d_j + r_{jk} \quad \forall j \in \overline{J_k} \tag{33}$$
$$s_j \geq 0 \quad \forall j \in \overline{J_k} \tag{34}$$
$$T_j \geq 0 \quad \forall j \in \overline{J_k} \tag{35}$$
$$\tag{36}$$

Pour chaque tâche $j \in \overline{J_k}$, s_j et T_j sont respectivement la date début et le retard de j.
Si $\overline{J_k} = \varnothing$ on considère que le minimum est 0.
On nomme ce problème $SP[\overline{J_k}]$.

c) On définit $i(j)$ comme la position occupée par la tâche j dans $SP[\overline{J_k}]$.
Pour appliquer la décomposition de Benders on commence par écrire le dual de $SP[\overline{J_k}]$:

$$\max \quad \sum_{j \in \overline{J_k}} \alpha_j r_{jk} + \sum_{j \in \overline{J_k}} \gamma_j (p_j - d_j + rjk) + \sum_{1}^{n-1} \beta_i p_{j(i)} \tag{37}$$
$$\alpha_j + \beta_{i(j)-1} - \beta_{i(j)} - \gamma_j \leq 0 \quad \forall j \in \overline{J_k}, j \neq j(1) \tag{38}$$
$$\alpha_j - \beta_1 - \gamma_j \leq 0 \quad j = j(1) \tag{39}$$
$$\gamma_j \leq 1 \quad \forall j \in \overline{J_k} \tag{40}$$
$$\alpha_j \geq 0 \quad \forall j \in \overline{J_k} \tag{41}$$
$$\beta_i \geq 0 \quad \forall i \in [1;n-1] \tag{42}$$
$$\gamma_j \geq 0 \quad \forall j \in \overline{J_k} \tag{43}$$
$$\tag{44}$$

On nomme ce problème $DSP[\overline{J_k}]$.
γ_j, β_i et α_j sont respectivement les variables duales associées aux contraintes (31), (32) et (33) de $SP[\overline{J_k}]$.
On définit ϕ_k comme l'ensemble des points extrêmes définis par les contraintes de $DSP[\overline{J_k}]$.

En appliquant la décomposition de Benders sur notre modèle sans périodes de temps on obtient:

$$
\begin{aligned}
\min \quad & \lambda_1(\sum_{k\in K} y_k c_k) + \lambda_2(\sum_{j\in J}\sum_{k\in K} x_{jk}e_{jk}) + \lambda_3(\sum_{k\in K}\theta_k) && (45)\\
\text{s.c.} & \sum_{k\in K} x_{jk} = 1 \quad \forall j\in J && (46)\\
& \sum_{k\in K} y_k = m && (47)\\
& x_{jk} \le y_k \quad \forall j\in J, \forall k\in K && (48)\\
& z_{ij} + z_{ji} \ge x_{ik} + x_{jk} - 1 \quad \forall i\in J, \forall j\in J, i\neq j, \forall k\in K && (49)\\
& \theta_k \ge \sum_{j\in J} x_{jk}\alpha_j r_{jk} + \sum_{j\in J} x_{jk}\gamma_j(p_j - d_j + rjk) + \sum_{1}^{n-1}\beta_i p_{j(i)} \quad \forall k\in K, \forall(\alpha,\beta,\gamma)\in\phi_k && (50)\\
& y_k \in \{0;1\} \quad \forall k\in K && (51)\\
& x_{jk}\in\{0;1\} \quad \forall j\in J, \forall k\in K && (52)\\
& z_{ij}\in\{0;1\} \quad \forall i\in J, \forall j\in J, i\neq j && (53)\\
& \theta_k \ge 0 \quad \forall k\in K && (54)\\
& && (55)
\end{aligned}
$$

Si tout ϕ_k est énuméré la variable θ_k est la valeur du retard des tâches sur la machine du site k. Si ϕ_k est énuméré partiellement on a une borne supérieure par dualité faible.

Problème: dans la contrainte (50) $pj(i)$ dépend de $\overline{J_k}$ et donc des variables de décisions, je suspecte que c'est parce que j'ai écrit $SP[\overline{J_k}]$ avec des $j\in\overline{J_k}$, il aurait fallu écrire des $j\in J$ et introduire les variables de décisions qui déterminent $\overline{J_k}$ sous forme de coefficients (en tant que données) qui auraient été utilisées dans les contraintes.

3.2 Benders Combinatoire

a) Sur une machine d'un site $k\in K$ le problème à résoudre est le problème d'ordonnancement $1|r_j|\sum T_j$ où les date de disponibilité sont r_{jk}, les durées p_j et les dates d'échéance sont $d_j - r_{jk}$ avec les tâches $j\in\tilde{J}_k$.
On nomme ce problème $P[\tilde{J}_k]$.

b) La dernière date de fin la plus petite possible est obtenue en démarrant avec la tâche qui a la date de disponibilité la plus petite et en démarrant chaque tâche dès que la précédente se termine sans temps mort.
On note cette date de fin $C(\tilde{J}_k) = \min_{j\in\tilde{J}_k}\{r_{jk}\} + \sum_{j\in\tilde{J}_k} p_j$

Le retard de la dernière tâche $\tilde{j}$ est $T_{\tilde{j}} \ge c_{\tilde{j}} - (d_{\tilde{j}} - r_{\tilde{j}k})$

On a:

$$c_{\tilde{j}} \ge C(\tilde{J}_k) \text{ (*)}$$

On a aussi:

$$d_{\tilde{j}} - r_{\tilde{j}k} \le \max_{l\in\tilde{J}_k}\{d_{\tilde{l}} - r_{\tilde{l}k}\}$$
$$-(d_{\tilde{j}} - r_{\tilde{j}k}) \ge -\max_{l\in\tilde{J}_k}\{d_{\tilde{l}} - r_{\tilde{l}k}\} \text{ (**)}$$

En sommant (*) et (**) on a:

$$c_{\tilde{j}} - (d_{\tilde{j}} - r_{\tilde{j}k}) \ge C(\tilde{J}_k) - \max_{l\in\tilde{J}_k}\{d_{\tilde{l}} - r_{\tilde{l}k}\} \quad (56)$$

$$c_{\tilde{j}} - (d_{\tilde{j}} - r_{\tilde{j}k}) \ge \min_{j\in\tilde{J}_k}\{r_{jk}\} + \sum_{j\in\tilde{J}_k} p_j - \max_{l\in\tilde{J}_k}\{d_{\tilde{l}} - r_{\tilde{l}k}\} \quad (57)$$

$$T_{\tilde{j}} \ge \min_{j\in\tilde{J}_k}\{r_{jk}\} + \sum_{j\in\tilde{J}_k} p_j - \max_{l\in\tilde{J}_k}\{d_{\tilde{l}} - r_{\tilde{l}k}\} \quad (58)$$

Cette expression donne une borne inférieure sur le retard de la dernière tâche et donc aussi une borne inférieure sur la somme des retards de toute solution.

$$T(\tilde{J}_k) \ge T_{\tilde{j}} \ge \min_{j\in\tilde{J}_k}\{r_{jk}\} + \sum_{j\in\tilde{J}_k} p_j - \max_{j\in\tilde{J}_k}\{d_{\tilde{j}} - r_{\tilde{j}k}\}$$

c) On suppose que $T(\tilde{J_k'}) < T(\tilde{J_k})$, cela implique qu'il existe une solution de $P[\tilde{J_k'}]$ de valeur strictement plus petite que toutes les solutions de $P[\tilde{J_k}]$.

Si on prend cette solution de $P[\tilde{J_k'}]$ et qu'on retire les tâches $j \notin J_k$ on obtient une solution réalisable pour $P[\tilde{J_k}]$ dont le coût n'a pas augmenté car aucun début de tâche n'a été augmenté.
Il existerai donc une solution de $P[\tilde{J_k}]$ de valeur plus petite où égale que $T(\tilde{J_k'})$, on a une contradiction donc notre supposition est fausse.

Donc $T(\tilde{J_k'}) \geq T(\tilde{J_k})$

d) Pour un $k \in K$ fixé:

$$
\begin{aligned}
z_{jk} \leq &\quad 1 \quad \forall j \in \tilde{J}_k && (59)\\
\sum_{j \in \tilde{J}_k} z_{jk} \leq &\sum_{j \in \tilde{J}_k} 1 && (60)\\
\sum_{j \in \tilde{J}_k} z_{jk} \leq &\quad |\tilde{J}_k| && (61)\\
\sum_{j \in \tilde{J}_k} z_{jk} - |\tilde{J}_k| \leq &\quad 0 && (62)\\
\sum_{j \in \tilde{J}_k} z_{jk} - |\tilde{J}_k| + 1 \leq &\quad 1 && (63)\\
& && (64)
\end{aligned}
$$

Donc $\forall x \geq 0, \quad x \geq x(\sum_{j \in \tilde{J}_k} z_{jk} - |\tilde{J}_k| + 1)$ (*)

On a $\pi_k \geq T(\tilde{J}_k)$ car $T(\tilde{J}_k)$ est le retard total minimum. En utilisant (*) et $T(\tilde{J}_k) \geq 0$ on obtient:

$$\pi_k \geq T(\tilde{J}_k) \geq T(\tilde{J}_k)(\sum_{j \in \tilde{J}_k} z_{jk} - |\tilde{J}_k| + 1)$$

e) Les variables yk indiquent si on place une machine sur le site $k \in K$.

Les variables z_{jk} indiquent si on affecte la tâche $j \in J$ au site $k \in K$ (ces variables étaient nommées x_{jk} dans nos précédents modèles)

Les variables π_k donnent une borne supérieure sur le coût des retards des tâches affectées au site k.

On définit $\hat{j}$ comme l'ensemble des sous ensemble de J, $\hat{j}$ contient donc toutes les affectations possibles sur une machine.

$$\min \quad \lambda_1(\sum_{k \in K} y_k c_k) + \lambda_2(\sum_{j \in J} \sum_{k \in K} z_{jk} e_{jk}) + \lambda_3(\sum_{k \in K} \pi_k) \tag{65}$$

$$\text{s.c.} \sum_{k \in K} z_{jk} = 1 \quad \forall j \in J \tag{66}$$

$$\sum_{k \in K} y_k = m \tag{67}$$

$$z_{jk} \leq y_k \quad \forall j \in J, \forall k \in K \tag{68}$$

$$\pi_k \geq T(\tilde{J}_k)(\sum_{j \in \tilde{J}_k} z_{jk} - |\tilde{J}_k| + 1) \quad \forall k \in K, \forall \tilde{J}_k \in \hat{j} \tag{69}$$

$$y_k \in \{0;1\} \quad \forall k \in K \tag{70}$$

$$z_{jk} \in \{0;1\} \quad \forall j \in J, \forall k \in K \tag{71}$$

$$\pi_k \geq 0 \quad \forall k \in K \tag{72}$$

f)

4 Extensions du problème

4.1 Plusieurs machines

4.1.1 Modèle sans période du temps

Adaptons le modèle PLNE sans contraintes de temps à la nouvelle condition où chaque site k peut contenir jusqu'à m_k machines avec $1 \leq m_k \leq m$

On définit T l'ensemble des périodes de temps. $|T| = \lceil \max_{j,k}\{r_{jk}\} \rceil + \sum_{j \in J} p_j$
On peut aussi remplacer le nombre de périodes par la plus grande date de fin d'une solution obtenue précédemment.

Pour chaque site $k \in K y_k$ Devient une variable entière indiquant le nombre de machines installées sur le site k avec.

$$1 \leq y_k \leq m_k$$

Pour chaque tâche $j \in J$ et pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$x_{jkl} = \begin{cases} 1 & \text{si la tâche } j \text{ est affectée au site } k\text{à la machine l} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ et pour chaque période $t \in T$ on a une variable binaire indicatrice:

$$s_{jt} = \begin{cases} 1 & \text{si la tâche } j \text{ commence à la période } t \\ 0 & \text{sinon} \end{cases}$$

Pour chaque paire orientée de tâche différentes i et j, $i \neq j$ on on a une variable binaire indicatrice:

$$z_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est planifiée avant la tâche } j \text{ sur un même site} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ on a une variable réelle:

$$T_j \in \mathbb{R}^+ \text{ le retard de la tâche } j$$

soit $L = [1, m_k]$
Le modèle est le suivant:

$$\min \quad \lambda_1(\sum_{k \in K} y_k c_k) + \lambda_2(\sum_{j \in J}\sum_{k \in K} x_{jkl} e_{jk}) + \lambda_3(\sum_{j \in J} T_j) \tag{73}$$

$$\text{s.c.} \sum_{k \in K}\sum_{l=1}^{m_k} x_{jkl} = 1 \quad \forall j \in J \tag{74}$$

$$\sum_{k \in K} y_k = m \tag{75}$$

$$x_{jkl} \leq y_k \quad \forall j \in J, \forall k \in K, \forall l \in L \tag{76}$$

$$z_{ij} + z_{ji} \geq x_{ikl} + x_{jkl} - 1 \quad \forall i \in J, \forall j \in J, i \neq j, \forall k \in K \forall l \in L \tag{77}$$

$$s_j \geq \sum_{k \in K} x_{jkl} r_{jk} \quad \forall j \in J \tag{78}$$

$$s_i + p_i \leq s_j + M(1 - x_{ikl} * xjkl * z_{ij}) \quad \forall l \in L \quad \forall i \in J, \forall j \in J, i \neq j \tag{79}$$

$$T_j \geq s_j + p_j - d_j + \sum_{k \in K} x_{jkl} r_{jk} \quad \forall j \in J \tag{80}$$

$$1 \leq y_k \leq m_k \quad \forall k \in K \tag{81}$$

$$x_{jkl} \in \{0; 1\} \quad \forall j \in J, \forall k \in K \tag{82}$$

$$z_{ij} \in \{0; 1\} \quad \forall i \in J, \forall j \in J, i \neq j \tag{83}$$

$$s_j \geq 0 \quad \forall j \in J \tag{84}$$

$$T_j \geq 0 \quad \forall j \in J \tag{85}$$

4.1.2 modèle avec période du temps

Pour chaque tâche $j \in J$, pour chaque site $k \in K$ et pour chaque période $t \in T$ on a une variable binaire indicatrice: $a_{jkt} = \{ \begin{array}{ll} 1 & \text{si la tâche } j \text{ est en cours d'éxécution sur le site } k \text{ à la période t} \\ 0 & \text{sinon} \end{array}$

Pour chaque site $k \in K y_k$ Devient une variable entière indiquant le nombre de machines installées sur le site k avec.

$$1 \leq y_k \leq m_k$$

Pour chaque tâche $j \in J$ et pour chaque site $k \in K$ on a une variable binaire indicatrice:

$$x_{jkl} = \begin{cases} 1 & \text{si la tâche } j \text{ est affectée au site } k \text{à la machine l} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ et pour chaque période $t \in T$ on a une variable binaire indicatrice:

$$s_{jt} = \begin{cases} 1 & \text{si la tâche } j \text{ commence à la période } t \\ 0 & \text{sinon} \end{cases}$$

Pour chaque paire orientée de tâche différentes i et j, $i \neq j$ on on a une variable binaire indicatrice:

$$z_{ij} = \begin{cases} 1 & \text{si la tâche } i \text{ est planifiée avant la tâche } j \text{ sur un même site} \\ 0 & \text{sinon} \end{cases}$$

Pour chaque tâche $j \in J$ on a une variable réelle:

$$T_j \in \mathbb{R}^+ \text{ le retard de la tâche } j$$

Le modèle est le suivant:

$$\min \quad \lambda_1(\sum_{k \in K} y_k c_k) + \lambda_2(\sum_{j \in J} \sum_{k \in K} x_{jkl} e_{jk}) + \lambda_3(\sum_{j \in J} T_j) \tag{86}$$

$$\text{s.c.} \sum_{k \in K} \sum_{l=1}^{m_k} x_{jkl} = 1 \quad \forall j \in J \tag{87}$$

$$\sum_{k \in K} y_k = m \tag{88}$$

$$x_{jkl} \leq y_k \quad \forall l \in L \quad \forall j \in J, \forall k \in K \tag{89}$$

$$\sum_{t \in T} s_{jt} = 1 \quad \forall j \in J \tag{90}$$

$$a_{jkt} \geq \sum_{u \in [t-p_j+1;t]} s_{ju} + \sum_{l \in [1;m_k]} x_{jkl} - 1 \quad \forall j \in J, \forall k \in K, \forall t \in T \tag{91}$$

$$\sum_{j \in J} a_{jkt} \leq 1 \quad \forall k \in K, \forall t \in T \tag{92}$$

$$\sum_{t \in T} s_{jt}.t \geq \sum_{k \in K} \sum_{l \in [1;m_k]} x_{jkl} r_{jk} \quad \forall j \in J \tag{93}$$

$$T_j \geq \sum_{t \in T} s_{jt}.t + p_j - d_j + \sum_{l \in [1;m_k]} \sum_{k \in K} x_{jkl} r_{jk} \quad \forall j \in J \tag{94}$$

$$1 \leq y_k \leq m_k \quad \forall k \in K \tag{95}$$

$$x_{jkl} \in \{0;1\} \quad \forall l \in L \quad \forall j \in J, \forall k \in K \tag{96}$$

$$s_{jt} \in \{0;1\} \quad \forall j \in J, \forall t \in T \tag{97}$$

$$a_{jkt} \in \{0;1\} \quad \forall j \in J, \forall k \in K, \forall t \in T \tag{98}$$

$$T_j \geq 0 \quad \forall j \in J \tag{99}$$

$$\tag{100}$$

4.2 Consommation d'électricité

4.2.1 Modèle avec périodes de temps

Fonction Objectif:

$$\min \lambda_1 \left(\sum_{k \in K} y_k c_k \right) + \lambda_2 \left(\sum_{j \in J} \sum_{k \in K} x_{jk} e_{jk} \right) + \lambda_3 \left(\sum_{j \in J} T_j \right) + \lambda_4 \left(\sum_{k \in K} \sum_{t \in T} \text{Coût Électricité}_{kt} \right)$$

Contraintes:

$$\sum_{k \in K} x_{jk} = 1 \qquad \forall j \in J \quad (2)$$

$$\sum_{k \in K} y_k = m \qquad (3)$$

$$x_{jk} \leq y_k \qquad \forall j \in J, \forall k \in K \quad (4)$$

$$\sum_{t \in T} sjt = 1 \qquad \forall j \in J \quad (5)$$

$$ajkt \geq \sum_{u \in [t-p_j+1, t]} s_{ju} + x_{jk} - 1 \qquad \forall j \in J, \forall k \in K, \forall t \in T \quad (6)$$

$$\sum_{j \in J} ajkt \leq 1 \qquad \forall k \in K, \forall t \in T \quad (7)$$

$$\sum_{t \in T} sjt \cdot t \geq \sum_{k \in K} x_{jk} r_{jk} \qquad \forall j \in J \quad (8)$$

$$T_j \geq \sum_{t \in T} sjt \cdot t + p_j - d_j + \sum_{k \in K} x_{jk} r_{jk} \qquad \forall j \in J \quad (9)$$

$$\text{Consommation}_{kt} = \sum_{j \in J} a_{jkt} e_j \qquad \forall k \in K, \forall t \in T$$

$$\text{Coût Électricité}_{kt} = c_{kt} \cdot \text{Consommation}_{kt} \cdot (1 + 0.5 \cdot I(\text{Consommation}_{kt} > E_k)) \qquad \forall k \in K, \forall t \in T$$

Variables :

- $y_k \in \{0, 1\}$: Indique si une machine est installée sur le site k.
- $x_{jk} \in \{0, 1\}$: Indique si la tâche j est affectée au site k.
- $sjt \in \{0, 1\}$: Indique si la tâche j commence à la période t.
- $ajkt \in \{0, 1\}$: Indique si la tâche j est en cours d'exécution sur le site k à la période t.
- $T_j \geq 0$: Le retard de la tâche j.
- $\text{Consommation}_{kt} \geq 0$: La consommation d'électricité sur le site k à la période t.

4.2.2 Modèle Sans Périodes de Temps

Fonction Objectif:

Minimisez:

$$\lambda_1 \left(\sum_{k \in K} y_k c_k \right) + \lambda_2 \left(\sum_{j \in J} \sum_{k \in K} x_{jk} e_{jk} \right) + \lambda_3 \left(\sum_{j \in J} T_j \right)$$

Contraintes:

$$\sum_{k \in K} x_{jk} = 1 \qquad \forall j \in J \quad (17)$$

$$\sum_{k \in K} y_k = m \qquad (18)$$

$$x_{jk} \leq y_k \qquad \forall j \in J, \forall k \in K \quad (19)$$

$$zij + zji \geq x_{ik} + x_{jk} - 1 \qquad \forall i, j \in J, i \neq j, \forall k \in K \quad (20)$$

$$sj \geq \sum_{k \in K} x_{jk} r_{jk} \qquad \forall j \in J \quad (21)$$

$$si + pi \leq sj + M(1 - zij) \qquad \forall i, j \in J, i \neq j \quad (22)$$

$$T_j \geq sj + pj - dj + \sum_{k \in K} x_{jk} r_{jk} \qquad \forall j \in J \quad (23)$$

Variables :

- $y_k \in \{0, 1\}$: Indique si une machine est installée sur le site k.
- $x_{jk} \in \{0, 1\}$: Indique si la tâche j est affectée au site k.
- $z_{ij} \in \{0, 1\}$: Indique si la tâche i est planifiée avant la tâche j sur le même site.
- $sj \geq 0$: La date de début de la tâche j.
- $T_j \geq 0$: Le retard de la tâche j.

5 Résolution heuristique

Pour notre méthode de résolution heuristique on trie les tâches par ordre croissant de $p_j + d_j$.

On définit un recherche de la manière suivante:

On choisit m sites au hasard à ouvrir, ces sites sont $\bar{K}$.
Pour chaque site $k \in \bar{K}$ on fixe un temps courant $t_k = 0$.
Pour chaque tâche $j \in J$ on affecte j au site $k \in \bar{K}$ étant à la plus petite distance r_{jk} puis on fixe la date début de la tâche j à $s_j = \max\{r_{jk}, t_k\}$ et $t_k = s_j + p_j$
C'est la fin de la recherche, les affectations et les dates de début sont fixées.

L'algorithme complet prend en paramètre un temps maximal et effectue des recherches jusqu'à que le temps soit écoulé et renvoie la meilleure solution.

L'objectif de cet algorithme est de pouvoir faire des recherches en très peu de temps pour pouvoir essayer pleins de $\bar{K}$ différents.
Un faiblesse est que l'affectation des tâches une fois que $\bar{K}$ est fixé est toujours la même, introduire de l'aléatoire dans cette étape pourrait y remédier.
Une amélioration possible serait d'améliorer les solutions produites en faisant des recherches locales ou simplement en résolvant les problèmes d'ordonnancement de chaque machine.

6 Expérimentations numériques

Les tests ont été réalisés sur un intel-core i5 3.40GHZ.
On cherche à évaluer la capacité des méthodes de résolution à trouver la meilleure solution possible pour un temps $<$ 5min.

6.1 Capacité à trouver une solution

Le modèle PLNE avec périodes de temps prend plus de 30 minustes pour trouver une solution réalisable pour la plus petite instance (A 2 2 0).

Le modèle PLNE sans périodes de temps trouve une solution réalisable en 5 minutes pour plus de la moitié des instances A et l'optimum pour quelques instances.

L'heuristique trouve une solution réalisable en moins d'une seconde pour n'importe quelle instance, même les plus grandes de B.

6.2 Qualité des solutions

On compare ici les valeurs des solutions du modèle PLNE sans périodes de temps et de l'heuristique sur les instances où le modèle PLNE a pu trouver une solution réalisable. Le modèle PLNE a eu un temps de résolution maximal de 5 minutes, l'heuristique a eu 10 secondes pour faire ses recherches.

En moyenne l'heuristique trouve une solution de valeur $4,9\%$ plus petite que le modèle PLNE. Sur les petites instances où le modèle trouve l'optimum, l'heuristique trouve une solution de valeur 2% plus grande en moyenne.

6.3 Conclusion

L'heuristique surpasse le modèle PLNE sur n'importe quelle instance qui n'est pas suffisamment petite pour trouver l'optimum.

L'heuristique est notamment meilleure sur les très grandes instances où elle est capable de rapidement fournir une solution, contrairement au modèle PLNE qui est incapable d'en fournir même avec beaucoup de temps. Cependant on ne connaît pas la qualité de ces solutions données par l'heuristique (il faudrait calculer une borne inférieure).

On suspecte aussi que le modèle PLNE est capable de fournir de bien meilleurs résultats en ajoutant des contraintes qui améliorent la relaxation et en se servant d'abord de l'heuristique pour fixer des meilleurs big M.