

Ordonnancement cyclique sous contraintes¹ énergétiques

Carybe BÉGUÉ

ENS Rennes, Université Rennes 1

Résumé

Les problèmes d'ordonnancement cycliques trouvent de nombreuses applications dans l'industrie afin d'optimiser les coût de production. Nous allons introduire ici une notion de coût énergétique des ordonnancements chercherons à minimiser ce coût pour un temps de cycle fixe puis variable.

Référence : Stage effectué du 20 mai au 4 août 2019 au Laboratoire d'Analyse et d'Architecture Système de Toulouse, sous la direction de Laurent HOUSSIN et Sandra Ulrich NGUEVEU, LAAS, équipe ROC

I. INTRODUCTION

Les problèmes d'ordonnancement cycliques sont des problèmes d'optimisation combinatoire (en nombres entiers) très étudiés qui possède de nombreuses applications dans l'industrie. Ici, nous allons introduire une notion de coût énergétique des ordonnancements. Ce rapport s'intéresse à la mise au point d'un modèle qui trouve l'ordonnancement d'énergie minimale pour un temps de cycle borné, tout d'abord fixe puis variable.

La deuxième partie de ce rapport présente le sujet. La troisième partie propose une première approche théorique du problème. La quatrième présente les méthodes de résolutions mises en œuvre. La cinquième analyse ces modèles. Enfin, la sixième conclut ce rapport.

II. PRÉSENTATION DU PROBLÈME

A. Ordonnancement cyclique

a) *Instance et graphe de précedence:* On dispose d'un ensemble de tâches $1, \dots, n$ de durée (ou longueur) L_1, \dots, L_n qui dépendent les unes des autres par des contraintes de précédences et qui doivent être exécutées à l'infini. Pour cela, on associe à chaque tâche i une fonction $t_i : \mathbb{N} \rightarrow \mathbb{R}$ périodique de période α qui associe à chaque

occurrence de cette tâche une date de début. On se ramène donc à résoudre le système suivant :

$$t_j(k) \geq l_i + t_i(k - H_{ij}) \quad \forall k \geq H_{ij}, i, j \in \llbracket 1, n \rrbracket \quad (1)$$

$$t_i(k) \geq l_i + t_i(k - 1) \quad \forall i \in \llbracket 1, n \rrbracket \quad (2)$$

Les contraintes (1) entre la tâche j et la tâche i de hauteur H_{ij} traduisent le fait que "la $k^{\text{ème}}$ occurrence de la tâche j ne peut commencer qu'une fois que la $(k - H_{ij})^{\text{ème}}$ occurrence de la tâche i est terminée" (contraintes de précédence). Les contraintes (2) traduisent le fait qu'une occurrence d'une tâche doit attendre que l'occurrence précédence soit terminée pour démarrer (contraintes de non-réentrance).

Nous pouvons associer à une instance du problème un graphe de précédence $G = (\mathcal{T}, \mathcal{P})$ où $\mathcal{T} = \{1, \dots, n\}$, ensemble des tâches, et \mathcal{P} arcs entre ces tâches : $e_{ij} \in \mathcal{P} \Leftrightarrow H_{ij}$ définie. Les arcs e_{ij} sont doublement étiquetés par L_i et H_{ij} .

b) *Ordonnancement:* Comme $\forall k \in \mathbb{N}, t_i(k) = t_i(0) + \alpha \times k$, on peut donc se contenter de donner les dates $t_1(0), \dots, t_n(0)$ et la période α de l'ordonnancement. Par la suite, nous représenterons un ordonnancement \mathcal{S} par des valeurs $t_1, \dots, t_n \in \mathbb{R}$, dates de début de la première occurrence de chaque tâche, et $\alpha \in \mathbb{R}$, période de l'ordonnancement.

Avec ces informations, on peut réécrire les contraintes (1) et (2) [2] :

$$t_j - t_i + \alpha \times H_{ij} \geq l_i \quad \forall i, j \in \mathcal{P} \quad (3)$$

$$\alpha \geq l_i \quad \forall i \in \llbracket 1, n \rrbracket \quad (4)$$

Résoudre un problème d'ordonnancement cyclique est de trouver un ordonnancement de ces tâches de période minimale. Cela revient à résoudre le problème d'optimisation $\alpha^* = \min_{\alpha \in \mathbb{R}} \alpha$ sous les contraintes linéaires (3) et (4).

Deux exemples d'ordonnancement sont présentés en annexe VII-A.

c) α garanti entier: On a $\alpha^* = \max_{c \text{ circuit de } G} \frac{L(c)}{H(c)}$ [3] avec $L(c)$ et $H(c)$ respectivement somme des longueurs et des hauteurs des arcs de c (notons qu'il existe un ordonnancement cyclique d'un ensemble de tâches \mathcal{T} si et seulement si $H(c) > 0$ pour tous circuits du graphe de précédence de ces tâches). Les circuits critiques de G sont ceux tels que $\frac{L(c)}{H(c)} = \alpha^*$.

Il est possible d'avoir $\alpha \in \mathbb{Q}$. Dans ce cas, on peut déterminer grâce à l'algorithme d'Howard [1] les circuits critiques et donc multiplier L_1, \dots, L_n par le *ppcm* des hauteurs de ces circuits pour garantir $\alpha \in \mathbb{N}$. Par la suite, nous supposons que c'est le cas.

B. Ajout de contraintes énergétiques

À présent, nous associons à chaque tâche i une puissance P_i nécessaire à son exécution. La puissance dépensée à un instant t est la somme des puissances de chacune des tâches exécutées à cet instant. Comme nous nous intéressons à l'énergie consommée, nous introduisons une échelle des temps Δt qui correspond à la durée en seconde d'une unité de temps. Elle est initialisée à 1 mais si α doit être remis à l'échelle, on divise Δt par le *ppcm* des hauteurs de des circuits critique afin de garantir que chaque tâche consomme la même énergie au total.

Soit à présent une fonction C . Pour une demande donnée x , une source doit fournir une puissance $C(x)$. On introduit de nouvelles contraintes sur t_i et α et les variables δ_i et \mathcal{E} suivantes (avec $\alpha^* \leq \alpha_{min} \leq \alpha_{max}$) :

$$\forall i, t \in \llbracket 1, n \rrbracket, \llbracket 0, \alpha - 1 \rrbracket$$

$$t_i \in \mathbb{N} \quad (5)$$

$$\alpha \in \llbracket \alpha_{min}, \alpha_{max} \rrbracket \quad (6)$$

$$\delta_t^i = 1 \text{ si } \frac{t - t_i}{\alpha} - \left\lfloor \frac{t - t_i}{\alpha} \right\rfloor < \frac{L_i}{\alpha} \quad (7)$$

$$\delta_t^i = 0 \text{ sinon} \quad (8)$$

$$\mathcal{E} = \sum_{t=0}^{\alpha-1} \left[\Delta t \times \sum_{i=1}^n C(P_i \times \delta_t^i) \right] \quad (9)$$

Grâce aux contraintes (7) et (8), $\delta_t^i = 1$ si et seulement si la tâche i est exécutée à l'instant t . En effet, quitte à considérer des occurrences de tâche négatives, on a :

$$\text{tâche } i \text{ à } t \Leftrightarrow \exists k, t_i + \alpha k \leq t < t_i + \alpha k + L_i \quad (10)$$

$$\Leftrightarrow \exists k, k \leq \frac{t - t_i}{\alpha} < k + \frac{L_i}{\alpha} \quad (11)$$

$$\Leftrightarrow \frac{t - t_i}{\alpha} - \left\lfloor \frac{t - t_i}{\alpha} \right\rfloor < \frac{L_i}{\alpha} \quad (12)$$

L'inégalité (12) est valable pour tout instant t en considérant éventuellement des occurrences négatives. Ainsi, indépendamment des valeurs des t_i , il nous suffit de faire varier t entre 0 et $\alpha - 1$ pour avoir la répartition des demandes en énergie au cours d'une période (mais ceci nécessite d'imposer la contrainte (5) et $L_i \in \mathbb{N}$ car le temps est discrétisé).

La contrainte (9) définit l'énergie dépensée lors d'un cycle pour un ordonnancement donné.

La fonction C peut être approchée par une fonction affine par morceaux potentiellement discontinue qui s'annule en 0. Nous supposons dans la suite que c'est le cas.

Notre objectif à présent sera de résoudre le problème d'optimisation $\min \mathcal{E}$ sous les contraintes (3), (5), (6), (7), (8), (9) (α étant plus grand que α^* , les contraintes (4) n'ont plus d'intérêt). Nous appellerons ce problème $Min\mathcal{E}_{\alpha NF}$ (pour α non fixé).

Nous étudierons aussi les variations du coût optimal pour une instance donnée en fonction de α . Pour cela, nous allons minimiser le coût de notre solution à α fixé, nous appellerons ce problème $Min\mathcal{E}_{\alpha F}$.

Notons $\mathcal{E}(\alpha, \mathcal{S})$ le coût d'un ordonnancement \mathcal{S} de période α , $\mathcal{E}^*(\alpha)$ le coût optimal pour α fixé et \mathcal{E}^* le coût optimal pour tous α . De plus, notons $\mathcal{L}_{\mathcal{S}}$ la liste des puissances nécessaires à chaque instant et $\beta(\mathcal{S}) = \#\{p > 0, p \in \mathcal{L}_{\mathcal{S}}\}$, le nombre d'instant où au moins une tâche est exécutée par cycle. Notons que $\sum_{p \in \mathcal{L}_{\mathcal{S}}} p = \sum_{i=1}^n l_i \times P_i$ et donc ne dépend pas de l'ordonnancement \mathcal{S} .

Le problème $Min\mathcal{E}_{\alpha F}$ est NP-complet (démonstration en annexe VII-B). Par réduction immédiate, $Min\mathcal{E}_{\alpha NF}$ l'est aussi.

III. VARIATION DU COÛT OPTIMAL EN FONCTION DE α

A. Cas où C a un morceau

Avec $b \in \mathbb{R}^+$ et $a \in \mathbb{R}^{+*}$, soit C une fonction vérifiant :

$$C : x \in \mathbb{R} \mapsto \begin{cases} 0 & \text{si } x = 0 \\ ax + b & \text{sinon} \end{cases}$$

Ainsi, pour tout ordonnancement \mathcal{S} , on a :

$$\mathcal{E}(\alpha, \mathcal{S}) = \Delta t \times \sum_{\substack{p \in \mathcal{L}_{\mathcal{S}} \\ p > 0}} (a \times p + b) \quad (13a)$$

$$= \Delta t \times [b\beta(\mathcal{S}) + a \sum_{p \in \mathcal{L}_{\mathcal{S}}} p] \quad (13b)$$

Donc \mathcal{E} dépend uniquement de α et du nombre de "trous" dans \mathcal{S} . Le minimum est atteint en $\alpha = \alpha^*$ (démonstration en annexe VII-C).

B. Cas où C a deux morceaux

C supposée continue sur \mathbb{R}^{+*} vérifiant :

Avec $b \in \mathbb{R}^+$ et $a, c, d \in \mathbb{R}^{+*}$, on a :

$$C : x \in \mathbb{R} \mapsto \begin{cases} 0 & \text{si } x = 0 \\ xa + b & \text{si } x < d \\ xc + b + d(a - c) & \text{sinon} \end{cases}$$

Notons $\mathcal{N}_{\mathcal{S}} = \{p \in \mathcal{L}_{\mathcal{S}}, p > d\}$. On montre aisément :

$$\mathcal{E}(\alpha, \mathcal{S}) = \Delta t \times \sum_{p \in \mathcal{L}_{\mathcal{S}}} ap \quad (14a)$$

$$+ \Delta t \times b\beta(\mathcal{S}) \quad (14b)$$

$$+ \Delta t \times \sum_{p \in \mathcal{N}_{\mathcal{S}}} (c - a)(p - d) \quad (14c)$$

Voyons ce que signifie chaque terme :

- (14a) correspond au coût d'exécution des tâches si la fonction C n'avait qu'un seul morceau de pente a ;
- (14b) correspond au coût fixe d'activation des machines durant la période ;
- (14c) correspond à l'ajout d'un second morceau de pente c à partir de l'abscisse d .

Ainsi, à α fixé, seules les parties (14c) et (14b) dépendent de l'ordonnancement : $\mathcal{E}(\alpha, \mathcal{S})$ est minimale si et seulement si (14c)+(14b) l'est.

1) *Cas C concave continue*: Dans ce cas, si $e_1, e_2 \in \mathbb{R}^+$,

$$\begin{cases} e_1, e_2 \leq a \\ e_1 + e_2 > a \end{cases} \Rightarrow C(e_1 + e_2) < C(e_1) + C(e_2)$$

Intuitivement, l'énergie est minimale lorsque la solution est la plus "compacte" car on vise la zone de plus forte pente. Il semble logique que la solution optimale soit atteinte en α^* mais je n'ai pas réussi à le démontrer formellement.

2) *Quelques hypothèses*: Lors de tests, nous avons pu établir quelques hypothèses sur les variations de $\mathcal{E}(\alpha)$ en fonction de α .

L'exemple 1 de l'annexe VII-A nous prouve que $\mathcal{E}(\alpha)$ n'est pas nécessairement monotone. Cependant, empiriquement, il semblerait que $\mathcal{E}(\alpha)$ ne puisse pas croître plus d'une fois. De même, si $\mathcal{E}(\alpha)$ croît pour un α donné, elle ne pourra pas décroître pour un temps de cycle plus élevé.

Intuitivement, les ordonnancements obtenus pour α assez grand devant α^* ont bien souvent un instant où aucune tâche n'est exécutée. Dans ce cas, il est possible de d'étirer l'ordonnancement pour obtenir un autre ordonnancement d'un temps de cycle supérieur de même énergie, comme dans la figure 1.

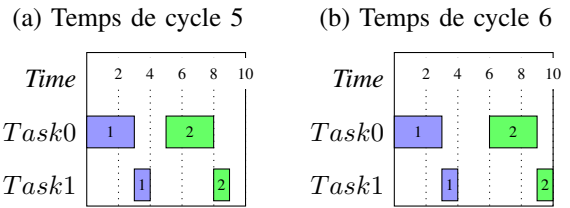


FIGURE 1: Étirement d'un ordonnancement

IV. MÉTHODE DE RÉOLUTION

A. Solveur

Pour résoudre les problèmes $Min\mathcal{E}_{\alpha F}$ et $Min\mathcal{E}_{\alpha NF}$, j'ai utilisé le solveur Cplex [5]. Le modèle associé à la détermination de α^* étant linéaire non entier, le problème est polynomial et se résout avec l'algorithme du simplexe.

Dans le cas des MILP (Mixed Integer Linear Program), programme linéaire contenant des variables entières, le solveur explore l'arbre des solutions possibles. A chaque étape, il choisit une variable entière ou binaire x non encore fixée, il divise son domaine de définition en deux. Deux sous-problèmes correspondant à chaque partie du domaine de définition sont ainsi créé. Le solveur évalue ensuite une borne inférieure sur le sous-problème obtenu. Si cette borne est inférieure à la meilleure solution connue, il continue d'explorer cette branche avec le domaine de définition x restreint, sinon, il coupe la branche. Ce procédé est appelé *Branch-and-Bound* [4]. La borne inférieure du nœud est obtenue grâce à une relaxation continue. Toutes les variables discrètes sont transformées en variables continues et le problème obtenu est résolu grâce à l'algorithme du simplexe. La solution obtenue est une borne inférieure du problème initial.

B. Énergie minimale à α fixé

Intéressons-nous au problème $Min\mathcal{E}_{\alpha F}$ ($\min \mathcal{E}$ sous les contraintes (SC.) (3), (5), (7), (8), (9)).

Comme Cplex n'accepte pas les contraintes d'inégalités strictes, nous introduisons des constantes de précisions $\epsilon_f > 0$ et $\epsilon_\delta > 0$ (respectivement pour (16) et (8)), fixés empiriquement à 10^{-4} .

Pour le calcul de $\lfloor \frac{t-t_i}{\alpha} \rfloor$, nous devons introduire des variables entières $f_t^i \in \mathbb{N}$ et donc les contraintes :

$$f_t^i \leq \frac{t-t_i}{\alpha} \quad (15)$$

$$\frac{t-t_i}{\alpha} \geq f_t^i + 1 - \epsilon_f \quad (16)$$

L'implémentation des variables δ_t^i est faite par des contraintes indicatrices de Cplex, qui traduisent les contraintes (7) et (8) :

$$\delta_t^i = 1 \Rightarrow \frac{t-t_i}{\alpha} - f_t^i \leq \frac{L_i}{\alpha} - \epsilon_\delta \quad (17)$$

$$\delta_t^i = 0 \Rightarrow \frac{t-t_i}{\alpha} - f_t^i \geq \frac{L_i}{\alpha} \quad (18)$$

$Min\mathcal{E}_{\alpha F}$ peut être exprimé comme $\min \mathcal{E}$ SC. (3), (5), (17), (18), (15), (16) (9). Il s'agit donc bien d'un MILP.

C. Énergie minimale avec α non fixé

Une première méthode naïve pour résoudre le problème $Min\mathcal{E}_{\alpha NF}$ est de résoudre $Min\mathcal{E}_{\alpha F}$ pour chaque $\alpha \in \llbracket \alpha_{min}, \alpha_{max} \rrbracket$. Nous avons par la suite mis au point un modèle permettant de résoudre directement notre problème avec α variable. Les efficacités des deux approches est comparées dans la partie V-B.

1) *Solveur à α variable*: On souhaite à présent résoudre $Min\mathcal{E}_{\alpha NF}$ SC.(3), (5), (6), (7), (8), (9) (le modèle final est décrit intégralement en annexe VII-D).

La période α étant à présent une variable du problème, les contraintes (17), (18), (15) et (16) ne sont plus linéaires. Pour y remédier, nous posons le changement de variables : $\tau = \frac{1}{\alpha}$, $u_i = \frac{t_i}{\alpha} \forall i \in \llbracket 1, n \rrbracket$ proposé par Claire HANEN dans [3]. On obtient les nouvelles contraintes (27), (32), (33), (34) et (35), explicitées en annexe VII-D.

Pour calculer l'énergie d'une période, comme α est à présent une variable du problème, il faut introduire de nouvelles variables binaires $\rho_t = 1 \Leftrightarrow t < \alpha$. L'énergie d'une période devient :

$$\mathcal{E} = \sum_{t=0}^{\alpha_{max}-1} \left[\Delta t \times \rho_t \sum_{i=1}^n C(P_i \times \delta_t^i) \right] \quad (19a)$$

$$= \sum_{t=0}^{\alpha_{max}-1} \left[\Delta t \times \sum_{i=1}^n C(P_i \times \delta_t^i \rho_t) \right] \quad (19b)$$

Le produit $\delta_t^i \times \rho_t$ nous impose d'introduire une nouvelle variable binaire $\lambda_t^i = \delta_t^i \times \rho_t$ afin de linéariser (19b). Les contraintes (37), (38) et (39) de l'annexe (VII-D) calculent λ_t^i de manière linéaire.

Pour des raisons d'implémentation, nous ajoutons de plus des variables intermédiaires p_t , la puissance demandé à l'instant t , et $p_t^{cost} = C(p_t)$ le coût de la puissance nécessaire à t .

Comme l'énergie peut être minimale pour plusieurs temps de cycle, nous cherchons le temps de cycle minimal.

Pour cela, il est possible de résoudre un système multi-objectif, c'est-à-dire où l'on cherche à optimiser deux variables. Comme la version de Cplex utilisée (12.8) ne permet pas d'implémenter de tels systèmes, on résout tout d'abord en minimisant l'énergie, puis en fixant l'énergie à cette valeur et en maximisant τ .

Voici une représentation des différentes étapes de l'algorithme en figure 1, avec τ_{exp} la valeur de τ de la solution optimale trouvée par la minimisation de \mathcal{E} .

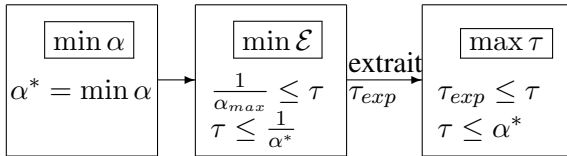


FIGURE 2: Schéma résolution à α non fixé

2) *Cas fonction de coût convexe continue partout*: On se place à présent dans le cas où C est convexe et continue (en particulier $C(0^+) = 0$).

Dans ce cas précis, on peut montrer que $\mathcal{E}(\alpha)$ est décroissant en fonction de α . Le minimum est donc atteint pour $\alpha_{max} = \sum_{i=0}^n L_i$. Reste à trouver $\min\{\alpha \mid \mathcal{E}(\alpha) = \mathcal{E}(\alpha_{max})\}$. On peut procéder par dichotomie, pour une complexité de $\mathcal{O}(\log(\alpha_{max}) \times C(n))$, avec $C(n)$ la complexité de la recherche d'énergie minimale à α fixé.

V. ANALYSE DU MODÈLE

A. Comparaisons de différentes versions

Pour représenter les contraintes de la forme $bin = 0 \Rightarrow x \leq 0$ (*), il est possible d'utiliser des contraintes indicatrices ou d'utiliser des contraintes de type big-M. (*) devient $x - M * bin \leq 0$ avec M constante positive assez grande (ici, $M \geq \sup x$). Afin d'éviter les erreurs d'approximations et faciliter le traitement de ces contraintes par Cplex, il faut que M soit le plus petit possible (ici, on prendra $M = \sup x$). On

traduit de manière similaire les autres types de contraintes indicatrices.

La question est de savoir lesquelles de ces contraintes, indicatrices ou big-M, sont les plus efficaces en terme de temps de calcul. Pour cela, nous avons fait des tests sur différentes instances générées aléatoirement en relevant le nombre de nœuds explorés, le nombre d'itérations du simplexe, le gap final, la dernière borne, le nœud au quel on a trouvé la meilleure solution et le temps de calcul (à titre indicatif simplement).

Nous avons également supprimé les variables ρ_t en les remplaçant par des contraintes sur λ_t^i . De même, nous avons exécuté des tests sur les mêmes instances pour savoir quelles combinaisons fonctionnent le mieux.

Les résultats sont résumés dans l'annexe VII-E1. Pour chacune des fonctions de coût testées (convexe et concave à deux morceaux), l'utilisation des variables δ_t^i et des contraintes indicatrices semble en moyenne respectivement 2 et 3 fois efficace que l'utilisation de contraintes big-M et l'absence de variables δ_t^i . Nous allons donc utiliser cette version du modèle pour la suite.

B. Comparaisons énumération et modèle à α non fixé

Comme évoqué précédemment, nous avons deux algorithmes pour résoudre $Min\mathcal{E}_{\alpha NF}$ et trouver le temps de cycle minimal d'énergie minimale :

- résoudre pour chaque α le problème $Min\mathcal{E}_{\alpha F}$;
- utiliser le modèle développé dans la partie IV-C1 qui résout notre problème à l'aide de deux MILP.

Pour cela, nous avons résolu 36 instances de $Min\mathcal{E}_{\alpha NF}$ en utilisant les deux méthodes. Les moyennes et médianes des temps de calcul, le nombre de nœuds explorés et d'itérations du simplexe sont indiqués dans les tableaux I, II et III. La méthode itérative explore 3,8 fois moins de nœuds et appelle 2,6 fois moins de fois l'algorithme du simplexe que la méthode de la partie IV-C1. De ce point de vue là, la méthode itérative semble plus efficace. Notons qu'elle a nécessité 10 fois plus de temps que l'autre mais cette

différence peut être expliqué par le fait que les deux tests n'ont pas été réalisés au même moment.

TABLE I: Méthode 1 : α fixe

	Temps(s)	Nœuds	Itérations
Moyenne	5185,9	70444,5	652761,8
Médiane	4472	57922	523332,5

TABLE II: Méthode 2 : α non fixe

	Temps(s)	Nœuds	Itérations
Moyenne	706,9	267672,8	1715712,9
Médiane	692	141318	1034214,5

TABLE III: Différence entre les deux méthodes

	Temps(s)	Nœuds	Itérations
Moyenne	-4478,5	197228,3	1062951,2
Médiane	-3550	94292,5	472848,5

VI. CONCLUSION

En conclusion, nous avons mis au point un premier algorithme qui permet de résoudre $Min\mathcal{E}_{\alpha F}$.

Par la suite, nous avons créé deux modèles pour résoudre $Min\mathcal{E}_{\alpha NF}$: un qui se base sur le précédent en énumérant les temps de cycles admissibles, l'autre qui résout directement notre problème avec α variable.

Nous avons par la suite testé ces deux modèles. A priori, la méthode énumérative est la plus efficace au vu du nombre de nœuds explorés. Des tests plus poussés que nous n'avons pas pu réaliser faute de temps sont nécessaires afin de conclure.

Enfin, nous avons tenté une approche plus théorique afin de déterminer des propriétés sur les variations de $\mathcal{E}(\alpha)$ en fonction de α . Nous avons pu émettre quelques hypothèses sans réussir à les affirmer ou infirmer.

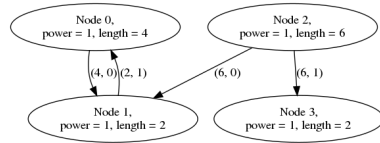
Par la suite, il serait nécessaire d'effectuer plus de tests afin de perfectionner le modèle et pouvoir résoudre des instances plus importantes. Nous pourrions également approfondir l'approche théorique afin de résoudre plus facilement certains types d'instances.

RÉFÉRENCES

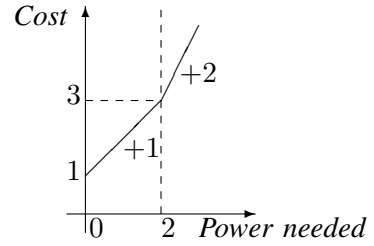
- [1] Stéphane GAUBERT. Howard's multichain policy iteration algorithm for max-plus linear maps. <http://www.cmap.polytechnique.fr/~gaubert/HOWARD2.html>, Jul. 2008.
- [2] Idir HAMAZ. *Méthodes d'optimisation robuste pour les problèmes d'ordonnement cyclique*. PhD thesis, Université de Toulouse 3 Paul Sabatier, 2018.
- [3] Claire HANEN. Study of a np-hard cyclic scheduling problem : The recurrent job-shop. *European Journal of operational research*, 72(1) :82–101, 1994.
- [4] Michel MINOUX. *Programmation mathématique, Théorie et algorithmes*, chapter 7 : Programmation en nombres entiers. Lavoisier, 2^{ème} édition, 2008.
- [5] Cplex python api reference manual. <http://perso.ensta-paristech.fr/~diam/ro/online/cplex/cplex1271/refpythoncplex/html/cplex-module.html>. ©copyright IBM Corp. 1987, 2017.

VII. ANNEXES

A. Exemples d'ordonnancement

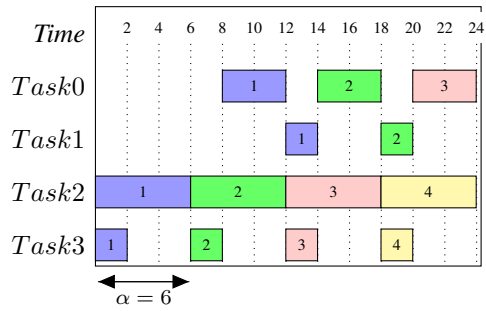


(a) Graphe

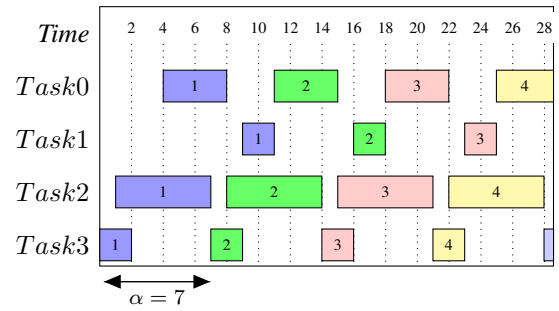


(b) Fonction de coût

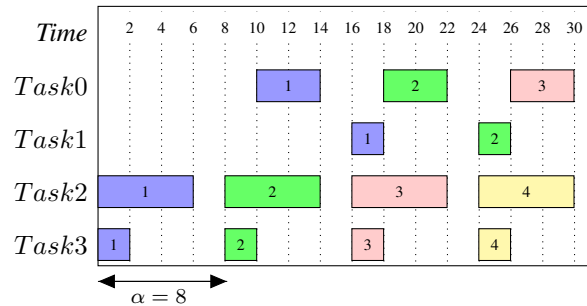
FIGURE 3: Instance exemple 1



(a) Exemple 1, $\alpha = 6$



(b) Exemple 1, $\alpha = 7$



(c) Exemple 1, $\alpha = 8$

FIGURE 4: Ordonnancement exemple 1

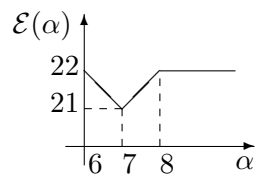
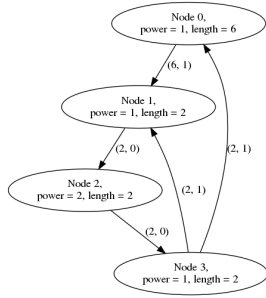
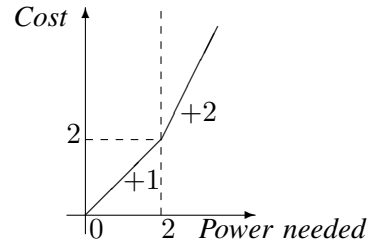


FIGURE 5: Allure de $\mathcal{E}(\alpha)$

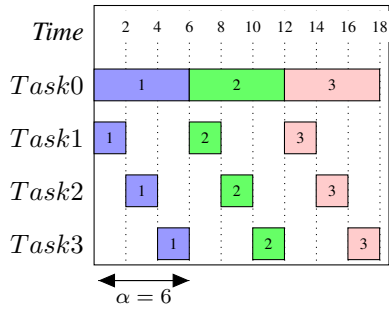


(a) Graphe

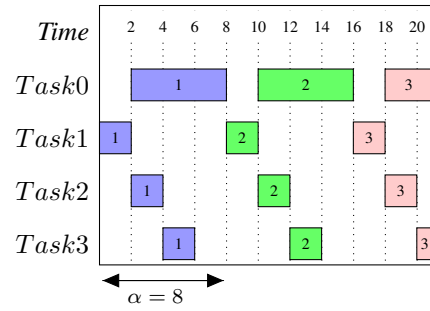


(b) Fonction de coût

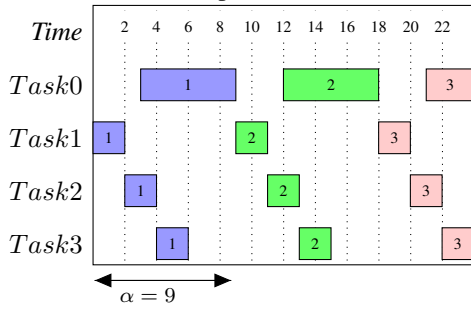
FIGURE 6: Instance exemple 2



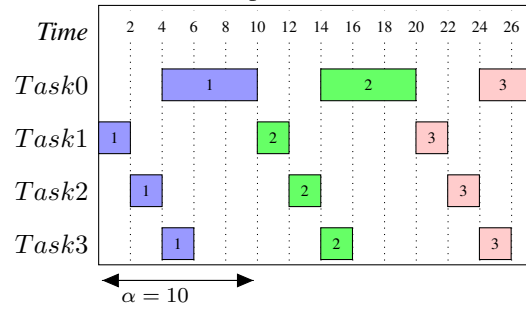
(a) Exemple 2, $\alpha = 6$



(b) Exemple 2, $\alpha = 8$



(c) Exemple 2, $\alpha = 9$



(d) Exemple 2, $\alpha = 10$

FIGURE 7: Ordonnancement exemple 2

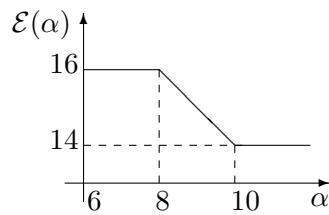


FIGURE 8: Allure de $\mathcal{E}(\alpha)$

B. $Min\mathcal{E}_{\alpha F}$ NP-complet

Nous allons démontrer que le problème $Min\mathcal{E}_{\alpha F}$ est NP-complet pour des fonctions de coût convexe ou concave discontinue. Notons $Min\mathcal{E}_{dec}$ la version décisionnelle de ce problème qui prend

en entrée $G = (\mathcal{T}, \mathcal{P})$ un graphe de précédence, α le temps de cycle fixé, C une fonction de coût et $k \in \mathbb{N}$ et qui renvoie *Oui* si et seulement si il existe un ordonnancement \mathcal{S} de temps de cycle α tel que $\mathcal{E}(\alpha, \mathcal{S}) \leq k$

$Min\mathcal{E}_{dec} \in NP$: Un certificat de $Min\mathcal{E}_{dec}$ est un ordonnancement quelconque \mathcal{S} . On vérifie que cet ordonnancement vérifie les contraintes de précédences de G en $\mathcal{O}(|\mathcal{P}|)$. Le calcul de $\mathcal{E}(\alpha, \mathcal{S})$ ce fait en $\mathcal{O}(\alpha|\mathcal{T}|)$. On a donc un vérificateur en temps polynomial $\mathcal{O}(\mathcal{P} + \alpha|\mathcal{T}|)$.

$Min\mathcal{E}_{dec}$ NP-dur: Pour montrer cela, nous allons faire une réduction du problème de *bin packing* décisionnel.

bin packing: Nous disposons de $k \in \mathbb{N}$ boîtes de même capacité $C \in \mathbb{R}$ et de $n \in \mathbb{N}$ objets de volume $\{c_1, \dots, c_n\} \in \mathbb{R}^n$. Le problème de *bin packing* décisionnel renvoie *Oui* si et seulement si les n tâches peuvent être rangées dans les k boîtes sans que le volume total des tâches contenues dans chaque boîte de ne dépasse C .

Soit à présent $I = \{C, \{c_1, \dots, c_n\}, k\}$ une instance de *bin packing* décisionnel. Posons :

$$— C : d \in \mathbb{R} \mapsto \begin{cases} 0 & \text{si } d \leq C \\ d - C & \text{sinon} \end{cases} ;$$

$$— \alpha = k ;$$

— G composé de n tâches d'énergie $\{c_1, \dots, c_n\}$ de durée unitaire et sans contraintes de précédences entre elles (les seuls arcs de G sont les arcs de non-réentrance) ;

$I' = \{G, \alpha, C, 0\}$ est une instance de $Min\mathcal{E}_{dec}$, obtenue en appliquant une transformation polynomiale à I .

Si I est une instance positive de *bin packing*, alors en numérotant les boîtes, on peut poser $t_i = j$ avec t_i le début de la tâche i et j le numéro de la boîte contenant l'objet i dans le rangement de *bin packing*. Comme il n'y a pas de contraintes de précédences, l'ordonnancement obtenu est admissible pour G . De plus, comme $\forall j \in \llbracket 1, k \rrbracket, \sum_{\substack{i=1 \\ c_i \in b_j}}^n c_i \leq C$, avec b_j la boîte j , on a bien $\mathcal{E}(\alpha, \mathcal{S}) = 0$ et I' est

une instance positive de $Min\mathcal{E}_{dec}$.

Réciproquement, si I' est une instance positive de $Min\mathcal{E}_{dec}$, alors considérons $\mathcal{S} = \{t_1, \dots, t_n\}$ un ordonnancement d'énergie nulle. En numérotant les k boîtes, on met l'objet i dans la boîte j si et seulement si $j = \lfloor t_i \rfloor$.

Ainsi, $\sum_{\substack{i=1 \\ c_i \in b_j}}^n c_i = \max_{t \in [j, j+1[} e(t)$ et comme l'énergie de l'ordonnancement est nulle, la demande à

chaque instant est inférieure à C et donc I est une instance positive de *bin packing* décisionnel.

Ainsi, dans le cas où la fonction de coût est convexe, notre problème est bien NP-complet.

Dans le cas des fonctions concaves discontinues en $a > 0$, si le saut de discontinuité est assez grand, strictement supérieur à $\sum_{i=1}^n (P_i \times p + c)$, alors on montre que le problème est aussi NP-dur en posant :

$$C : d \in \mathbb{R} \mapsto \begin{cases} d & \text{si } d \leq C \\ \frac{d}{2} + kC + 1 & \text{sinon} \end{cases}$$

C. $Min\mathcal{E}_{\alpha NF}$ polynomial si C a un morceau

Nous allons démontrer que le problème $Min\mathcal{E}_{\alpha F}$ dans le cas où C ne possède qu'un morceau est polynomial. En particulier, nous allons montrer qu'un ordonnancement d'énergie minimale est de temps de cycle α^* .

On rappelle que l'énergie d'un ordonnancement dans le cas où $C(x) = ax + b$ si $x > 0$, 0 sinon est :

$$\mathcal{E}(\alpha, \mathcal{S}) = \Delta t \times [b\beta(\mathcal{S}) + a \sum_{p \in \mathcal{L}_{\mathcal{S}}} p]$$

Ainsi, $\mathcal{E}(\alpha) = \mathcal{E}^*$ si et seulement si $\beta(\mathcal{S})$ est minimal. Supposons par l'absurde :

$$\exists \hat{\alpha} > \alpha^*, \mathcal{E}(\hat{\alpha}) < \mathcal{E}(\alpha^*)$$

. Dans ce cas, $\beta(\hat{\alpha}) < \beta(\alpha^*)$.

Hors, $\beta(\alpha^*) = \alpha^*$ car par définition de α^* , il existe un circuit critique c de temps de cycle $\frac{L(c)}{H(c)} = \alpha^*$.

Dans ordonnancement optimal associé à $\hat{\alpha}$, il y a $\hat{\alpha} - \beta(\hat{\alpha})$ instants où aucune tâche n'est exécutée. Ainsi, on peut supprimer ces instants de l'ordonnancement et obtenir un nouvel ordonnancement de même énergie qui vérifie bien les contraintes de précédences de période $\beta(\hat{\alpha}) < \alpha^*$, ce qui contredit la minimalité de α^* .

On a bien l'ordonnancement d'énergie minimale atteint en α^* dans le cas où C ne possède qu'un seul morceau. De plus, comme $\mathcal{E}(\alpha^*, \mathcal{S})$ ne dépend que de $\beta(\mathcal{S}) = \alpha^*$, tout ordonnancement de période α^* est d'énergie minimale, et le problème est bien polynomial.

D. Modèle à α non fixé

a) *Minimisation de l'énergie:* Avec Δt échelle de temps, α_{min} et α_{max} bornes positives de α choisies par l'utilisateur, t_i dates de début de la première occurrence de chacune des n tâches et en posant $\tau = \frac{1}{\alpha}$ et $u_i = \frac{t_i}{\alpha}$:

$$\min \mathcal{E}$$

$$\mathcal{E}, \tau \in \mathbb{R}^+ \quad (20)$$

$$u_i \in \mathbb{R}^+ \quad \forall i \in \llbracket 0, n-1 \rrbracket \quad (21)$$

$$f_t^i \in \mathbb{N} \quad \forall t, i \in \llbracket 0, \alpha_{max} - 1 \rrbracket, \llbracket 0, n-1 \rrbracket \quad (22)$$

$$\delta_t^i, \lambda_t^i \in \{0, 1\} \quad \forall t, i \in \llbracket 0, \alpha_{max} - 1 \rrbracket, \llbracket 0, n-1 \rrbracket \quad (23)$$

$$\rho_t \in \{0, 1\} \quad \forall t \in \llbracket \alpha_{min}, \alpha_{max} - 1 \rrbracket \quad (24)$$

$$p_t, p_t^{cost} \in \mathbb{R}^+ \quad \forall t \in \llbracket 0, \alpha_{max} - 1 \rrbracket \quad (25)$$

$$\frac{1}{\alpha_{max}} \leq \tau \leq \frac{1}{\alpha_{min}} \quad (26)$$

$$u_j - u_i + H_{ij} \geq L_i \times \tau \quad \forall e_{ij} \in \mathcal{P}, i \neq j \quad (27)$$

$$\forall t \in \llbracket 0, \alpha_{max} - 1 \rrbracket$$

$$(\rho_t = 0) \Rightarrow \left(\tau \geq \frac{1}{t}\right) \quad (28)$$

$$(\rho_t = 1) \Rightarrow \left(\tau \leq \frac{1}{t+1}\right) \quad (29)$$

$$p_t = \sum_{i=0}^{n-1} (\lambda_t^i \times P_i) \quad (30)$$

$$p_t^{cost} = C(p_t) \quad (31)$$

$$\forall t, i \in \llbracket 0, \alpha_{max} - 1 \rrbracket, \llbracket 0, n-1 \rrbracket$$

$$\tau \times t - u_i - f_t^i \geq 0 \quad (32)$$

$$\tau \times t - u_i - f_t^i \leq 1 - \epsilon_{floor} \quad (33)$$

$$(\delta_t^i = 0) \Rightarrow (\tau \times (t - L_i) - u_i - f_t^i \geq 0) \quad (34)$$

$$(\delta_t^i = 1) \Rightarrow (\tau \times (t - L_i) - u_i - f_t^i \leq \epsilon_{cond}) \quad (35)$$

$$\forall t, i \in \llbracket 0, \alpha_{min} - 1 \rrbracket, \llbracket 0, n-1 \rrbracket$$

$$\lambda_t^i = \delta_t^i \quad (36)$$

$$\forall t, i \in \llbracket \alpha_{min}, \alpha_{max} - 1 \rrbracket, \llbracket 0, n-1 \rrbracket$$

$$\rho_t \geq \lambda_t^i \quad (37)$$

$$\delta_t^i \geq \lambda_t^i \quad (38)$$

$$\lambda_t^i \geq \rho_t + \delta_t^i - 1 \quad (39)$$

$$\mathcal{E} = \sum_{t=0}^{\alpha_{max}-1} (\Delta t \times p_t^{cost}) \quad (40)$$

b) *Minimisation du temps de cycle*: Le MILP est presque le même que pour la minimisation de l'énergie. L'objectif devient :

$$\max \tau$$

Avec e_{min} et τ_{min} l'énergie minimale et la valeur de τ renvoyées par le MILP précédent, on impose de plus :

$$\mathcal{E} \leq e_{min} + \epsilon_{\mathcal{E}}$$

$$\tau \geq \tau_{min}$$

E. Résultats tests

1) *Choix de la version du modèle de la partie IV-C1*: On a réalisé un premier test avec 100 graphes composés de 5 tâches avec une densité d'arcs de 0,3 générés aléatoirement. Pour les fonctions de coût, nous avons pris deux fonctions à 2 morceaux chacune, une convexe et l'autre concave. Voici les moyennes et médianes des temps de calcul (donnés à titre purement indicatif), nombre de nœuds explorés et itérations du simplexe effectués pour la première fonction convexe :

a) *Calcul de \mathcal{E}* : Moyenne :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération du simplexe
δ_t^i et contraintes indicatrices	1 225 (20min)	1 156 231	7 245 440
δ_t^i et contraintes big-M	1 924 (32min)	2 720 792	14 606 715
sans δ_t^i et avec contraintes indicatrices	2 246 (37min)	3 578 097	23 642 188

Médiane :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération du simplexe
δ_t^i et contraintes indicatrices	895 (15min)	535 367	3 592 070
δ_t^i et contraintes big-M	1 018 (17min)	833 188	4 585 704
sans δ_t^i et avec contraintes indicatrices	1 451 (24min)	1 249 766	7 729 271

b) *Calcul de τ* : Moyenne :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération du simplexe
δ_t^i et contraintes indicatrices	2 076 (35min)	2 691 101	19 293 447
δ_t^i et contraintes big-M	3 905 (1h5min)	7 221 087	45 882 750
sans δ_t^i et avec contraintes indicatrices	2 754 (46min)	4 659 180	34 639 126

Médiane :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération simplexe du
δ_t^i et contraintes indicatrices	965 (16min)	584 410	3 924 197
δ_t^i et contraintes big-M	1 194 (20min)	1 333 617	9 799 990
sans δ_t^i et avec contraintes indicatrices	1 398 (23min)	1 203 388	8 585 828

Pour le cas de la fonction concave, la maximisation de τ était presque immédiate car la solution optimale renvoyée après la première étape était bien souvent déjà de temps de cycle minimale égale à α^* . Nous ne nous intéresserons donc qu'à la première étape, c'est-à-dire la minimisation de \mathcal{E} .

c) Calcul de \mathcal{E} : Moyenne :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération simplexe du
δ_t^i et contraintes indicatrices	857 (14min)	465 778	3 705 972
δ_t^i et contraintes big-M	998 (17min)	883 021	5 975 689
sans δ_t^i et avec contraintes indicatrices	1 185 (20min)	1 135 100	9 729 580

Médiane :

Test	Temps de calcul (en s)	Nombres de nœuds explorés	Nombres d'itération simplexe du
δ_t^i et contraintes indicatrices	750 (12min30)	115 544	1 232 853
δ_t^i et contraintes big-M	804 (13min30)	206 938	1 750 417
sans δ_t^i et avec contraintes indicatrices	786 (13min)	326 635	3 043 629

Les moyennes sont plus de 2 fois plus élevées que les médianes des mêmes données. Cela nous indique que certaines instances sont beaucoup plus longues et complexes à résoudre.