

CIGI 2011

Processus d'accélération de projets sous contraintes de ressources avec modes de chevauchement

LUCAS GREZE¹, ROBERT PELLERIN¹, PATRICE LECLAIRE²

¹ CHAIRE DE RECHERCHE JARISLOWSKY/SNC-LAVALIN EN GESTION DE PROJETS INTERNATIONAUX

Ecole Polytechnique de Montréal
C.P. 6079, succ. Centre-ville
Montréal (QC), Canada H3C3A7
lucas.greze@polymtl.ca
robert.pellerin@polymtl.ca

² INSTITUT SUPÉRIEUR DE MÉCANIQUE DE PARIS –SUPMÉCA

3 rue Fernand Hainaut, 93407 Saint-Ouen, France
patrice.leclaire@supmeca.fr

Résumé –

Le chevauchement d'activités est une pratique courante pour accélérer l'exécution de projets. Cette technique consiste à exécuter en parallèle deux activités, normalement séquentielles, en autorisant l'activité en aval à commencer avec des informations préliminaires. Dans la pratique, le chevauchement est étudié en prenant compte les contraintes de ressources, mais ne prend pas toujours en compte les interactions entre activités, qui peuvent entraîner des retouches et des augmentations de coût. À l'opposé, la littérature scientifique comporte beaucoup d'études sur les interactions qui ne prennent pas en compte les contraintes de ressources. Dans ce papier, nous proposons un processus d'accélération de projet sous contrainte de ressources à l'aide du chevauchement d'activités. Le processus propose l'identification des couples d'activités chevauchables, l'évaluation des paramètres de chevauchement, jusqu'à l'ordonnement du projet. Nous considérons ici deux techniques d'ordonnement, soit un modèle de programmation linéaire et un modèle heuristique. Ces deux approches ont été testées sur neuf instances de projets pour illustrer et quantifier les gains obtenus. Les résultats obtenus illustrent l'efficacité du chevauchement en tant que technique d'accélération de projet. Nous présentons aussi les limites d'une méthode de résolution exacte pour une approche industrielle et l'efficacité des méthodes heuristiques.

Abstract –

Overlapping activities is widely used to accelerate project execution. Overlapping consists in executing in parallel two sequential activities by allowing a downstream activity to start before the end of an upstream activity based on preliminary information. In companies, overlapping is examined in resource constraints scheduling without considering interaction between activities and rework caused by alteration of information exchanged until finalized information is available at the completion of the upstream activity. By contrast, the literature deals with overlapping of couple of activities without considering a whole project with resource constraints. We propose a process of accelerating project using the overlapping of activities. This process describes the identification of overlappable couples of activities, the assessment of overlap parameters and then the scheduling of the project. We propose in this paper two different scheduling approaches, an integer linear programming problem and a heuristic algorithm. We applied this process on nine instances of project to illustrate and quantify the associated gains. Results illustrate the effectiveness of overlapping as a technique for accelerating project execution. We also present the limitations of an exact resolution of the RCPSP problem.

Mots clés - Chevauchement d'activité, ingénierie simultanée, gestion de projet, ordonnancement de projet

Keywords – Activity overlapping, concurrent engineering, project management, project scheduling

1 INTRODUCTION

Les problèmes d'ordonnement de projet avec contraintes de ressources (en anglais, RCPSP, *Resource Constrained Project Scheduling Problems*), constituent un domaine d'étude de prédilection. Parmi les extensions étudiées dans la littérature, différentes pratiques, dont le chevauchement, la compression et la substitution d'activités [Gerk et Quassim, 2008] ont été mises en avant pour réduire le temps d'exécution du projet afin de produire un échancier initial ou pour le modifier en cours d'exécution. Parmi les techniques d'accélération préconisées

lors de projets d'ingénierie, le chevauchement d'activités s'avère être une technique privilégiée pour réduire le temps nécessaire aux phases de développement [Bogus et al., 2005]. En pratique, cette technique consiste à démarrer une activité avant de recevoir l'ensemble des informations requises au préalable. Cela implique toutefois des tâches de retouche et des modifications suite à la réception d'informations complémentaires transmises après le démarrage de l'activité. Cette pratique génère dans bien des cas une charge de travail et un coût d'exécution supplémentaires. Au final, il en résulte une réduction du temps total d'exécution du projet, représentée par

la différence entre la période de chevauchement et le temps supplémentaire du aux retouches, et un coût supplémentaire associé.

Parmi les études portant sur le sujet, on note deux types de modèles de chevauchement développés dans la littérature. Le premier type considère uniquement des couples d'activités sans contrainte de ressources pour établir le meilleur compromis entre chevauchement et retouches. Krishnan *et al.* [Krishnan *et al.*, 1997] ont ainsi introduit les concepts d'évolution de l'information et de sensibilité de l'activité aval pour décrire les interactions entre les activités. Le premier critère décrit l'évolution de l'information utile pour l'activité en aval générée par l'activité en amont. La sensibilité mesure l'impact sur l'activité en aval des changements au sein de l'activité en amont. Plus l'impact est important, plus la sensibilité est grande. A partir de ces concepts, Krishnan a défini différentes stratégies de chevauchement adaptées.

Le second type de modèle de chevauchement considère, non pas un couple d'activités, mais un projet dans sa globalité en supposant que la relation entre la durée de chevauchement et la durée de retouche nécessaire est connue préalablement pour les activités chevauchables [Lin *et al.*, 2009]. La plupart de ces modèles utilisent des matrices de décomposition ou DSM (*Design Structure Matrix*), introduites par Steward [Steward, 1981] pour représenter les dépendances entre les activités, limiter les rétroactions et identifier les opportunités de chevauchement d'activités [Maheswari et Varghese, 2005]. Parmi les modèles développés, Gerk et Quassim ont proposé un modèle linéaire analytique d'accélération de projet utilisant la compression d'activité, le chevauchement et la substitution et en considérant des contraintes de ressources. Wang et Lin [Wang et Lin, 2008] ont aussi développé un modèle de simulation considérant des itérations et des probabilités de retouche.

Parmi ces approches, aucune ne considère des possibilités réalistes de chevauchement. En effet, toutes les configurations de chevauchement ne sont pas acceptables dans la réalité. Elles doivent correspondre à un certain mode d'exécution du couple d'activités qui correspond avec un échange d'information. L'objectif premier de ce papier est de proposer un processus d'accélération de projet avec contraintes de ressources à l'aide du chevauchement d'activités, applicable dans un milieu industriel. Nous proposerons donc dans un premier temps une méthode permettant d'obtenir des paramètres de chevauchement acceptables. Puis dans un second temps nous proposerons différentes méthodes de résolution du problème d'ordonnancement dans ce contexte.

La suite du papier est organisée de la façon suivante. La section 2 décrit la méthodologie proposée pour le processus d'accélération de projet sous contraintes de ressources avec mode de chevauchement. La section 3 présentera le modèle de programmation linéaire, les cas d'études ainsi que les résultats associés. La section 4 présentera l'approche heuristique et les résultats obtenus. Nous concluons le papier par une comparaison des différentes approches et une conclusion dans la section 5.

2 MÉTHODOLOGIE PROPOSÉE

La figure 1 illustre la méthodologie que nous proposons. Les premières étapes du processus consistent à identifier les opportunités de chevauchement. Une fois ces opportunités identifiées, il faut choisir des modes de chevauchement pertinents et déterminer les degrés de chevauchement et les

temps de retouches applicables à ces modes. Les modes de chevauchement décrivent des possibilités discrètes de chevauchement. La détermination de ces paramètres est cruciale pour l'application du modèle par la suite.

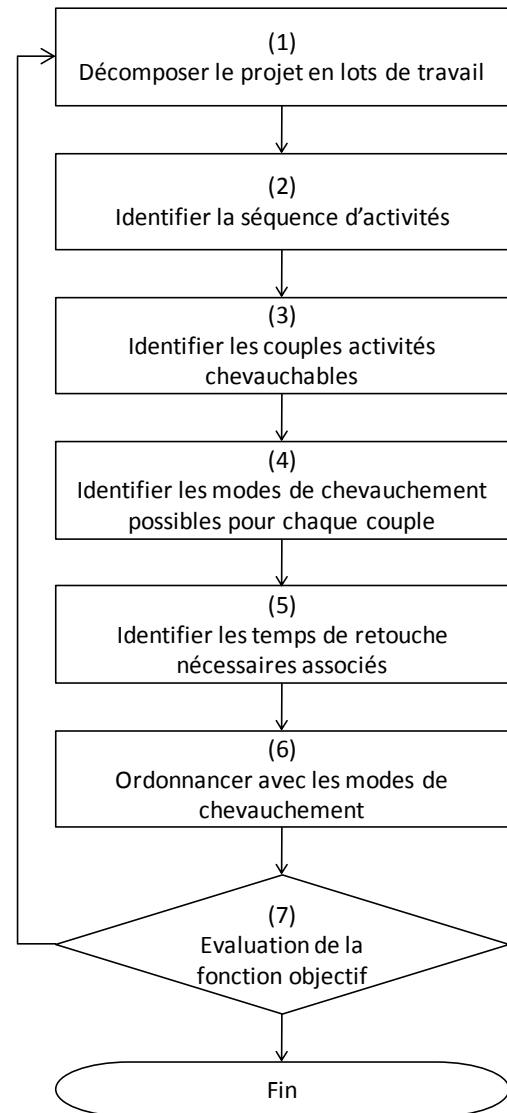


Figure 1 : Méthodologie d'accélération de projet

En premier lieu, il s'agit de décomposer le projet en lots de travail. La taille des lots de travail considérée dépend du niveau de détail exigé par la gestion du projet. Les lots de travail doivent représenter des activités de complexité et de durée comparables. C'est à partir de ces activités que nous allons considérer les possibilités de chevauchement.

Dans la pratique, c'est l'étude des interactions entre les activités qui permet d'identifier les activités chevauchables. La plupart des méthodes de planification de projet décrit les projets sous forme de réseaux, les nœuds représentant les activités (réseau «Activity on Node») ou les relations entre les activités (réseau «Activity on Arc»). Ces réseaux décrivent la séquence d'exécution des activités. Cependant, ces réseaux ne permettent pas de décrire les interactions entre les activités et de modéliser les flux d'information entre les activités.

Les matrices de décomposition (en anglais, *Design Structure Matrix*, DSM) permettent de modéliser ces relations supplémentaires entre les activités [Browning, 2001]. Les

échanges d'information entre activités peuvent avoir lieu à différents moments de l'avancement de l'activité. Il peut s'agir de plans, de données de dimensionnement ou de commandes de matériel, par exemple, nécessaires à l'exécution de l'activité en aval. Une matrice de décomposition est une matrice carrée où les lignes et les colonnes sont toutes deux associées aux activités. Les DSM visent à représenter les flux d'information entre les différentes activités, et permettent ainsi de repérer les retours d'information. Ces retours d'information sont compliqués à gérer au sein d'un projet car ils impliquent des changements dans une activité en cours ou terminée [Wang et Lin, 2008]. Les retours en arrière causent donc des modifications et du travail supplémentaire dans l'activité en amont pour s'accommoder aux changements impliqués par l'activité en aval. De manière à éviter les retours en arrière, on triangularise les DSM de manière à obtenir une séquence unidirectionnelle de flux d'information. En dernier recours, on choisit d'agréger les activités ou de les décomposer de manière à éliminer les retours en arrière. Ce processus nous permet d'obtenir une séquence d'exécution des activités, où les relations de précédence et les flux d'information sont orientés dans le même sens.

Dans la suite du papier, on suppose que des études préliminaires ont été menées de manière à identifier les relations entre les activités et à éliminer les retours en arrière. Les projets considérés sont donc composés uniquement d'activités dépendantes et indépendantes et le flux d'information (entre activités dépendantes) est considéré comme unidirectionnel.

La troisième étape du processus consiste à identifier les couples d'activités chevauchables. L'étude des échanges d'information entre les activités à l'aide des DSM est des données historiques disponibles permet de différencier les couples d'activités en deux catégories : les couples chevauchables et les couples non chevauchables. Les premiers représentent les cas où l'activité en aval peut commencer avec des informations préliminaires et recevoir durant son exécution les informations finales de l'activité en amont. De ce fait, on peut chevaucher les deux activités, l'activité en aval peut commencer avant que l'activité en amont ne soit finie. Les activités non chevauchables sont connectées par des relations de précédence classiques fin-début, alors que les activités chevauchables sont connectées par des relations de type fin-début plus un retard, le retard étant le temps de chevauchement.

On autorisera les chevauchements lorsque la sensibilité sera faible. On rappelle que la sensibilité définit l'impact plus ou moins important d'une modification apportée par l'activité en amont. Pour les couples d'activités dont la sensibilité est élevée, les changements dans l'information délivrée par l'activité en amont impliqueront de nombreuses retouches sur l'activité en aval. Le gain associé sera donc minime [Krishnan, 1996]. Dans la pratique, cela se résume à autoriser les chevauchements entre des activités d'ingénierie peu sensibles aux changements. A l'inverse, on évitera de chevaucher tous les couples d'activités comprenant des activités très sensibles aux changements.

Dans la suite du papier, nous noterons A et P l'ensemble des couples d'activités chevauchables et non chevauchables respectivement. De même nous noterons $A(i)$ et $P(i)$ l'ensemble des prédécesseurs de l'activité i qui sont

respectivement chevauchables et non chevauchables avec l'activité i .

$$Pred(i) = A(i) \cup P(i) \quad \forall i \in S$$

Les quatrième et cinquième étapes consistent à identifier les modes de chevauchement et les durées de retouche associés. La figure 2 représente le processus de chevauchement. L'activité j commence avec des informations préliminaires de l'activité i . Le taux de chevauchement associé au mode m , α_{ijm} , est exprimé comme une fraction de la durée de l'activité j . Une durée de retouche est par ailleurs ajoutée pour s'accommoder de la mise à jour des informations transmises lors de la réalisation de l'activité j . Le temps d'exécution des deux activités pour le mode de chevauchement m est donc égal à D_{ijm} exprimé comme suit :

$$D_{ijm} = d_i + d_j * (1 - \alpha_{ijm}) + r_{ijm}$$

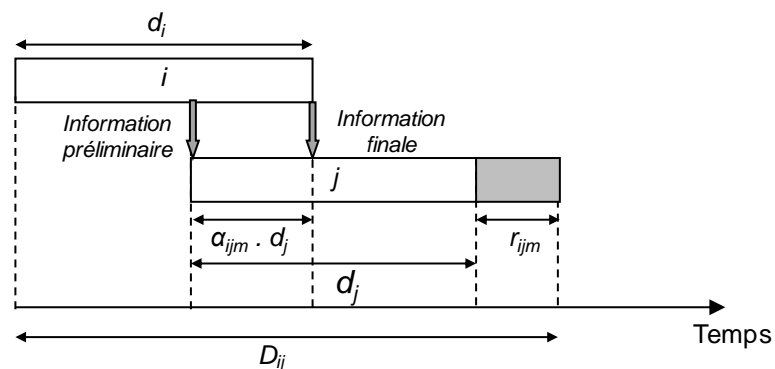


Figure 2 : Processus de chevauchement de deux activités i et j pour le mode m

Les pourcentages de chevauchement et les retouches associées aux modes sont définis de manière discrète. En effet, dans la réalité, l'ordonnancement d'un projet est établi sur la base de périodes discrètes (heures, jours, semaines, etc.). De plus, dans la pratique, l'avancement des activités est mesuré à partir de la réalisation de jalons qui correspondent à des avancements majeurs ou à la réalisation de biens livrables (spécifications, plan, commande de matière première, etc.) définis dans la phase initiale du projet. Les informations préliminaires nécessaires au commencement de l'activité en aval sont issues de la réalisation de ces jalons. Chaque mode de chevauchement est donc caractérisé par un pourcentage de chevauchement et une quantité de retouche supplémentaire à effectuer.

Les valeurs de α_{ijm} sont déterminées afin de correspondre avec la livraison des biens livrables majeurs de l'activité i . Cette méthodologie permet de s'assurer que les informations préliminaires utilisées sont des informations finalisées et approuvées. Le problème de la détermination du temps des retouches par rapport au temps de chevauchement est traité dans de nombreux articles de la littérature [Lin *et al.*, 2009.]. La définition des modes de chevauchement par rapport aux jalons permet de se baser facilement sur des données historiques.

Par la suite, les activités chevauchables peuvent être exécutées de manière conventionnelle sans chevauchement (mode 1), ou elles peuvent être chevauchées (mode > 1), voir figure 3. On

suppose qu'il n'y a pas de restriction concernant le nombre de prédécesseurs chevauchables pour une activité. Dans ce cas, on considérera le total de retouches à effectuer comme la somme de retouches associées aux modes de précédences. Suite à l'évaluation des paramètres de chevauchement, nous pouvons faire l'ordonnancement du projet. Nous avons abordé deux techniques d'ordonnancement, l'une à l'aide d'un modèle de programmation linéaire, l'autre à l'aide de règle heuristique. Nous allons les présenter et comparer leurs résultats dans les sections suivantes.

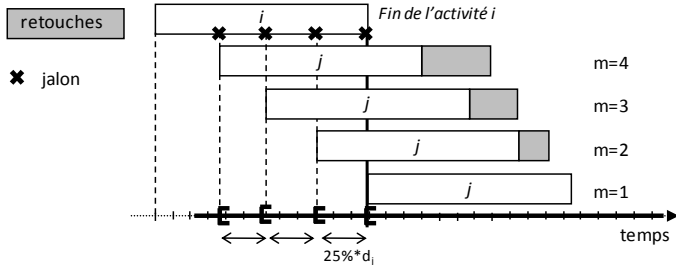


Figure 3 : Différents modes d'exécution possibles pour le couple (i,j)

Une fois l'ordonnancement terminé, on compare l'échéancier obtenu avec l'échéancier objectif, en termes de coût et de durée d'exécution. Si le temps d'exécution du projet ainsi que son coût associé sont acceptables, alors l'algorithme s'arrête ici. Le cas échéant, on recommence le processus à l'étape initiale, en décomposant plus finement les activités pour permettre de nouvelles opportunités de chevauchement qui permettront d'obtenir un échéancier plus compressé. On recommence le processus jusqu'à obtenir une solution acceptable.

3 RESOLUTION PAR PROGRAMMATION LINEAIRE

3.1 Notations

Un projet est défini par une liste d'activités S , comprenant deux activités fictives 0 et $n+1$ avec une durée d'exécution nulle, qui correspondent respectivement au début et à la fin du projet. On note d_j le temps d'exécution de l'activité j en considérant que toute l'information nécessaire des activités précédentes est disponible au début de l'activité j . Toutes les notations sont présentées dans le tableau 1. Les dates de fin au plus tôt et au plus tard des activités correspondent en pratique aux dates désirées a priori pour l'ordonnancement et ne correspondent pas forcément aux dates de fin au plus tôt et au plus tard calculées par une méthode d'ordonnancement. Ces dates sont déterminées par les praticiens et permettent avant tout de réduire le nombre de variables utilisées dans le modèle.

3.1 Le modèle de minimisation du temps d'exécution du projet avec prise en compte du coût induit par le chevauchement

Le coût induit par le chevauchement est considéré comme l'addition des coûts supplémentaires de coordination induits par le chevauchement des activités, et des coûts supplémentaires de retouche. Dans notre papier, nous considérons la durée et le coût de coordination respectivement comme instantanée et négligeable. Le coût des retouches est à l'inverse égal au coût d'utilisation des ressources pour le

travail supplémentaire. Nous supposons que les ressources impliquées dans les retouches sont les mêmes, en quantité et en qualification, que celles impliquées dans la réalisation de l'activité. Le coût total induit par la stratégie de chevauchement sera donc égal à :

$$C = \sum_{i \in A} \sum_{t=EF_i}^{LF_i} \sum_{m=1}^{m_i} \sum_{k \in R} X_{itm} * r_{im} * R_{ik} * \Theta_k \quad (1)$$

Tableau 1 : Définition des symboles

Symboles	Définition
S	Ensemble des activités
n	Nombre d'activités réelles
d_j	Durée d'exécution de l'activité j
A	Ensemble des couples d'activités chevauchables
P	Ensemble des couples d'activités non chevauchables
$A(j)$	Ensemble des prédécesseurs de j qui sont chevauchables avec l'activité j
$P(j)$	Ensemble des prédécesseurs de j qui ne sont pas chevauchables avec l'activité j
$Pred(j) = A(j) \cup P(j) \forall j \in S$	Ensemble des prédécesseurs de j
R	Ensemble des ressources
R_k	Quantité disponible constante pour la ressource k
R_{jk}	Nombre d'unités ressource k nécessaire par période pour l'exécution normale de l'activité j
Θ_k	Coût horaire de la ressource k
m_j	Nombre de mode d'exécution possible pour l'activité j
α_{ijm}	Pourcentage de la durée de chevauchement entre les activités i et j dans le mode d'exécution m , exprimé comme une fraction de d_j
r_{jm}	Durée des retouches associées au mode d'exécution m pour l'activité j
T	Durée maximale d'exécution du projet
$t = 0, \dots, T$	Périodes
EF_j	Date de fin au plus tôt de l'activité j
LF_j	Date de fin au plus tard de l'activité j

La variable de décision du modèle présenté est la variable binaire X_{jtm} . Elle est égale à 1 au temps t où l'activité j , exécutée dans le mode m , se termine. La variable est égale à 0 sinon.

Le modèle de programmation linéaire est présenté ci-dessous :

$$\text{Minimiser } \sum_{t=EF_{n+1}}^{LF_{n+1}} t \cdot X_{n+1,t,1} + \delta * C \quad (2)$$

Contraintes :

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} (t - d_j \cdot (1 - \alpha_{ijm}) - r_{jm}) \cdot X_{jtm} \quad (3)$$

$$\forall j \in S, \forall i \in Pred(j) \quad (3)$$

$$\sum_{j=2}^n \left[R_{jk} \cdot \left(\sum_{m=1}^{m_j} \sum_{b=t}^{t+d_j-1+r_{jm}} X_{jbm} \right) \right] \leq R_k, \forall k \in R, \forall t \in [0, T] \quad (4)$$

$$\sum_{m=1}^{m_i} \sum_{t=EF_i}^{LF_i} t \cdot X_{itm} \leq \sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} t \cdot X_{jtm}, \forall i \in A(j) \quad (5)$$

$$\sum_{m=1}^{m_j} \sum_{t=EF_j}^{LF_j} X_{jtm} = 1, \forall j \in S \quad (6)$$

$$0 \leq \delta * C < 1 \quad (7)$$

$$X_{jtm} \in \{0,1\}, \forall j \in S, \forall t \in [0, T], \quad (8)$$

L'objectif (2) minimise la durée totale du projet tout en assurant une solution à coût minimal. Le facteur δ correspond au facteur de priorisation de l'objectif. La contrainte (3) prend en compte les différents modes de précédence applicables. La contrainte (4) définit les contraintes de ressources. La contrainte (5) permet de s'assurer que l'activité en aval ne puisse se terminer avant son prédécesseur, même en cas de chevauchement. En effet on suppose que l'activité en aval a besoin des informations finales du prédécesseur pour se terminer. La contrainte (6) s'assure qu'un seul mode de précédence est appliqué pour chaque activité et que chaque activité est terminée à la fin du projet. La contrainte (7) permet de choisir δ afin de prioriser la fonction objectif : on minimisera d'abord le temps d'exécution du projet, pour lequel on recherchera ensuite la solution à coût minimal. Enfin, la contrainte (8) définit la variable de décision binaire.

3.2 Cas d'étude

Nous avons utilisé le générateur de projet PROGEN développé par Kolish [Kolish et Sprecher, 1996] pour générer les projets. Les projets générés sont des projets de 30 activités plus deux activités fictives représentant le début et la fin du projet. Les relations de précédence ainsi que les durées d'activités sont générées par PROGEN. Les paramètres de chevauchement ($A, m_j, \alpha_{ijm}, r_{jm}$) sont générés ensuite. Nous rappelons que, pour les activités non chevauchables, un seul mode est réalisable et les taux de chevauchement et les quantités de retouches associées sont nuls. Les modes de chevauchement générés sont illustrés dans la figure 3. On génère pour chaque couple d'activités chevauchables, 4 modes de précédence, correspondant à des fractions de 0%, 25%, 50% et 75% de la durée d'exécution de l'activité en aval. Ce découpage est arbitraire et dans la pratique, les modes de précédences doivent correspondre aux différents jalons délivrés par l'activité en amont comme expliqué précédemment.

Pour notre cas d'étude, nous considérons que la quantité de retouche nécessaire est égale à 40% de la durée de chevauchement.

Afin de mettre en lumière l'importance de la prise en compte des contraintes de ressources nous avons, pour chaque projet étudié, généré trois degrés de sévérité des contraintes de ressources grâce au paramètre RS défini par Kolisch *et al.* de la façon suivante :

$$R_k = Q_{k,\min} + Round(RS * (Q_{k,\max} - Q_{k,\min}))$$

$Q_{k,\min}$ est la quantité minimum acceptable pour la ressource k , égale à la demande maximale parmi les activités pour la ressource k . $Q_{k,\max}$ est égale à la demande maximale observée

pour la ressource k lorsque l'on exécute le projet sans considérer les contraintes de ressources.

RS représente finalement la force des contraintes de ressource, plus il est petit, plus le projet sera contraint. Les cas envisagés sont des contraintes sévères de ressources ($RS=0,5$), des contraintes moyennes ($RS=0,75$) et enfin l'absence de contrainte de ressources lorsque le projet est exécuté sans chevauchement ($RS=1$), où les ressources disponibles sont égales à la demande de ressource lors de l'exécution du projet sans chevauchement.

3.3 Résultats

Le modèle de programmation linéaire est résolu à l'aide d'AMPL Studio en utilisant le solveur CPLEX. La figure 4 illustre les différents gains obtenus suite à l'optimisation du planning de projet en prenant en compte les modes de chevauchement. On peut observer que plus les contraintes de ressources sont sévères, moins l'impact du chevauchement sur le temps d'exécution du projet est important. Pour le projet 3 par exemple, le gain varie de 15% pour des contraintes de ressources faibles à 4% pour des contraintes de ressources fortes. La figure illustre aussi l'augmentation du coût d'exécution du projet, associé au chevauchement d'activités. On observe que l'augmentation du coût n'est pas linéaire vis-à-vis de la réduction du temps d'exécution du projet, ce qui était prévisible du fait de l'aspect combinatoire du problème d'ordonnancement avec contrainte de ressources. Dans le tableau 2, on observe que les temps de résolution dépendent en grande partie de la sévérité des contraintes de ressources et du réseau d'activités considéré. Les temps de calculs observés sont trop importants et trop aléatoires pour que l'on puisse envisager une application du modèle à des projets industriels comprenant des centaines d'activités. En effet, le problème RCPSP est un problème NP difficile et on peut s'attendre à une augmentation significative des temps de calculs avec l'augmentation de la taille des problèmes. De plus, dans un contexte de re-planification, le temps de calcul se doit d'être limité au maximum afin de permettre une prise de décision rapide.

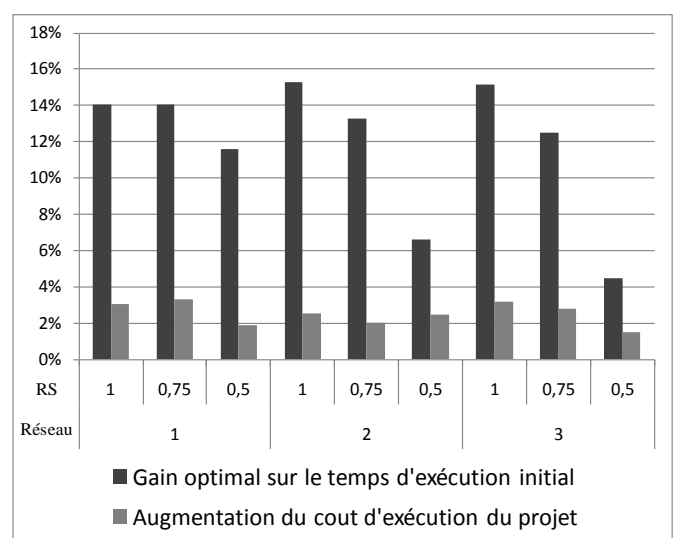


Figure 4 : Gain et coût associés à la résolution du modèle de programmation linéaire en nombre entier

Tableau 2 : Temps de calculs associés

1		2		3	
RS	Tps CPU (s)	RS	Tps CPU (s)	RS	Tps CPU (s)
1	9	1	5	1	3
0,75	28	0,75	13	0,75	7
0,5	93	0,5	>3600	0,5	21

A partir de ce constat, nous avons développé une approche heuristique qui vise à obtenir une bonne solution dans un temps de calcul raisonnable.

4 RÉSOLUTION PAR APPROCHE HEURISTIQUE

4.1 Présentation de l'algorithme

La figure 5 présente l'algorithme inspiré des méthodes de compression d'activité. La première étape consiste à ordonnancer le projet sans chevauchement d'activités afin d'obtenir un plan de référence. Ensuite on chevauche de manière incrémentale les activités selon un ordre de priorité. A chaque étape, on prend une décision par rapport au temps d'exécution du projet actif : si celui-ci est strictement inférieur au plan de référence, alors le plan actif devient le nouveau plan de référence. Sinon, on supprime l'activité de la liste des activités chevauchables. L'algorithme s'arrête lorsque qu'il n'y a plus d'opportunité de chevauchement dans la liste

Les activités chevauchables sont priorisées selon trois règles hiérarchisées. Tout d'abord, les activités sont classées en fonction de leur marge totale ; celles qui ont les marges totales les plus petites sont placées en premier. La marge totale est la différence des dates de fin d'une activité entre la planification au plus tard et la planification au plus tôt des activités. Les activités dont la marge totale est nulle sont les activités critiques du projet, elles ont une influence directe sur le temps d'exécution du projet. On ne peut cependant pas identifier de chemin critique du fait de la prise en compte des contraintes de ressources. En cas d'égalité de la marge totale entre plusieurs activités, on priorise les activités dont la quantité de ressource nécessaire à leur réalisation est la plus faible. Les activités mobilisant le moins de ressources sont en effet celles qui auront le moins d'impact sur les contraintes de ressources. Enfin, en cas d'égalité, on classe les activités en fonction de leur nombre de successeurs, plus celui-ci est important, plus les activités seront mise en avant. La liste de priorité est actualisée et mise à jour à chaque itération.

On rappelle que la fonction d'évaluation de l'algorithme est la comparaison entre le temps d'exécution du plan actif et le temps d'exécution du plan de référence. Le plan actif devient le plan de référence si le temps d'exécution actif est strictement inférieur au temps d'exécution de référence.

Nous avons implémenté l'algorithme dans un progiciel de gestion de projet, ici MS Project 2007. MS Project est un

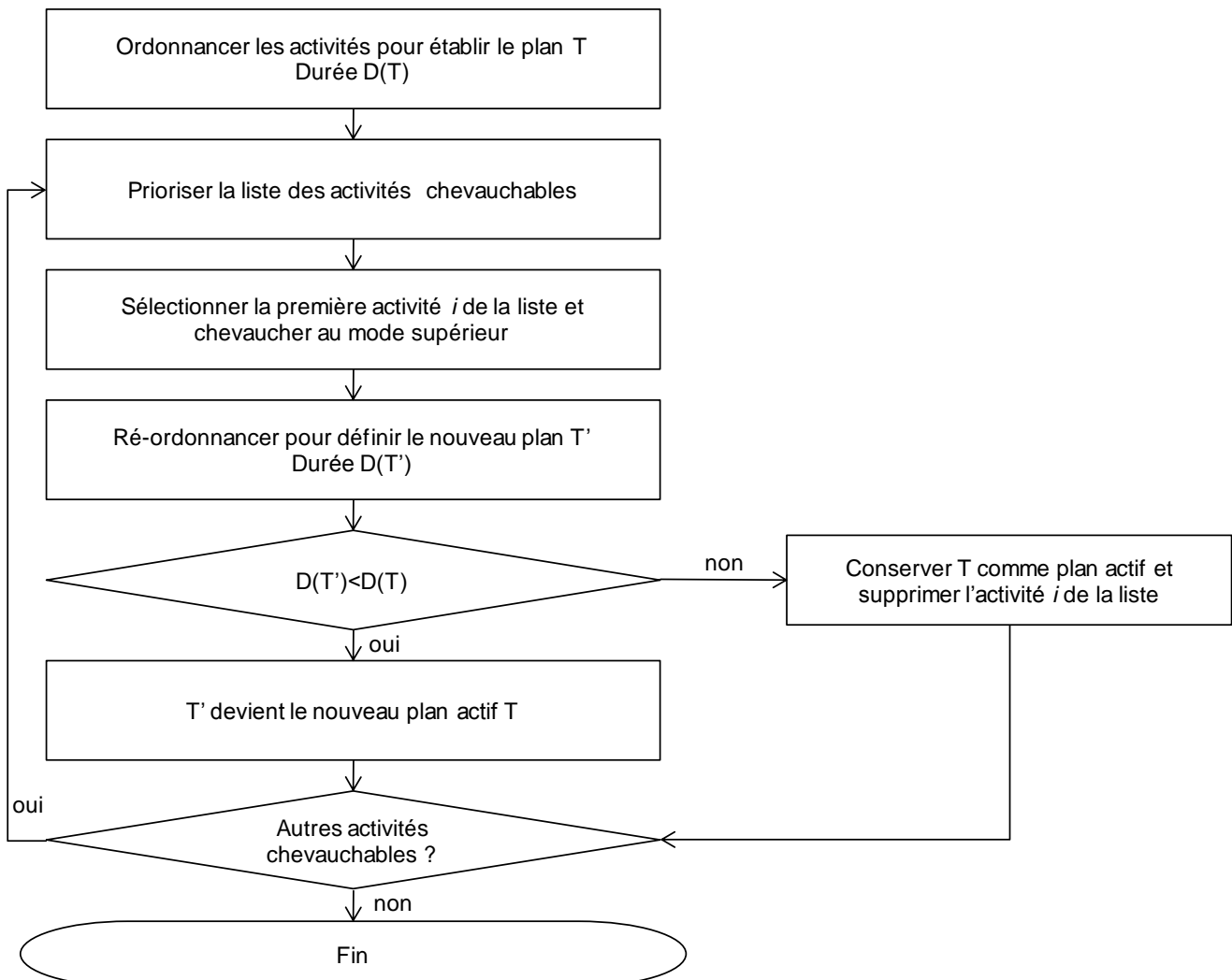


Figure 5 : Algorithme de résolution

progiciel de gestion de projet très répandu dans l'industrie. Les performances de son outil d'ordonnancement sont limitées [Kolisch, 1999] mais nous permettent de tester notre heuristique sur un outil utilisé dans la pratique. L'algorithme de résolution et les règles heuristiques ont été implantés à l'aide d'une macro Visual Basic liant un fichier de calcul Excel avec le logiciel de planification MS Project. Les résultats obtenus et leurs comparaisons avec les résultats optimaux seront présentés dans la section suivante.

4.2 Résultats

Les temps de références, correspondant aux temps d'exécution du projet sans chevauchement d'activité, donné par MS Project ne sont pas les temps d'exécution optimaux. Pour chaque projet nous allons donc évaluer le gain par rapport au temps de référence donné par MS Project et non pas par rapport à l'optimal. La figure 6 illustre les différents gains et coûts associés aux projets suite à l'exécution de l'algorithme précédemment décrit.

On observe sur la figure 6 que les gains obtenus sur le temps d'exécution du projet sont moins importants en moyenne (10% au lieu de 12%) et plus aléatoires que les gains obtenus par la méthode précédente alors que les coûts sont comparables. Le nombre d'itérations nécessaires à l'obtention des résultats est inférieur à 150 pour l'ensemble des instances considérées ; chaque itération impliquant le calcul d'un ordonnancement du projet.

Cette première constatation nous permet de confirmer l'efficacité de l'algorithme pour accélérer l'exécution du projet et cela en un temps de calcul raisonnable.

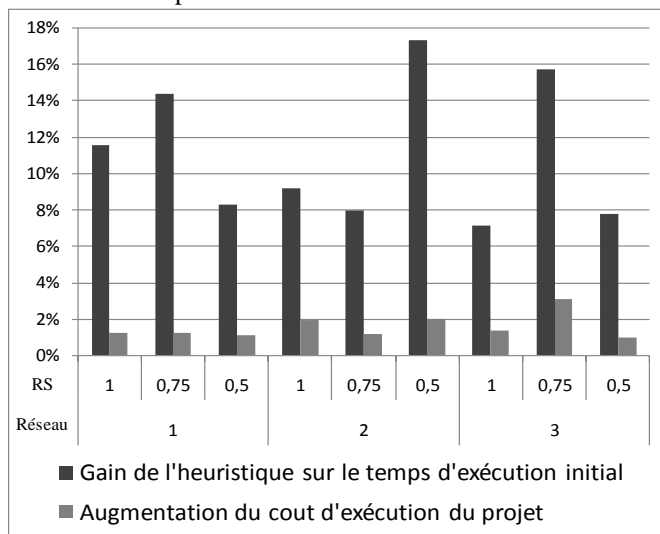


Figure 6 : Gain et coût associés à l'exécution de l'algorithme

Cependant la figure 6 ne nous permet pas d'évaluer la qualité des solutions obtenues. En effet, pour certaines instances on remarque que le gain sur le temps d'exécution peut être plus important avec l'heuristique, par exemple pour l'instance 2 avec RS=0,5, le gain observé est de 17% contre 7% à l'optimal. Cela s'explique par la valeur du temps de référence donnée par MS Project. Les gains observés précédemment ne sont pas comparables aux gains de la première méthode car ils ne se basent pas sur le même temps de référence.

Dans la figure 7, nous allons donc observer la qualité des solutions en les comparant aux solutions optimales. Une valeur de 0% signifie que la valeur trouvée par l'heuristique est égale à la valeur optimale. Une valeur supérieure à 0% illustre la marge d'erreur entre la valeur optimale et la valeur obtenue à la suite de l'exécution de l'algorithme.

La figure 7 illustre les marges d'erreur des temps de référence et des solutions données par l'algorithme. On observe tout d'abord que pour les temps de référence, l'erreur est d'autant plus importante que les contraintes de ressources sont sévères. Cela s'explique par la faible performance de l'outil d'ordonnancement utilisé (MS Project). La moyenne d'erreur pour les temps de référence est de 5% par rapport à l'optimal. Cette valeur est comparable avec les résultats donnés dans des études précédentes [Kolisch, 1999]. L'erreur observée s'explique par l'utilisation par MS Project de règles heuristiques simples permettant l'élaboration rapide d'un échéancier de projet quelles que soit les contraintes imposées. La qualité observée des solutions suite à l'application de l'algorithme est variable, et la marge d'erreur moyenne par rapport à l'optimal est de 6%. En comparaison avec les temps de référence, il n'y a donc qu'un pourcent d'erreur supplémentaire induit par l'utilisation de l'algorithme pour l'obtention d'un échéancier accéléré prenant en compte les modes de chevauchement et les temps de retouche associés.

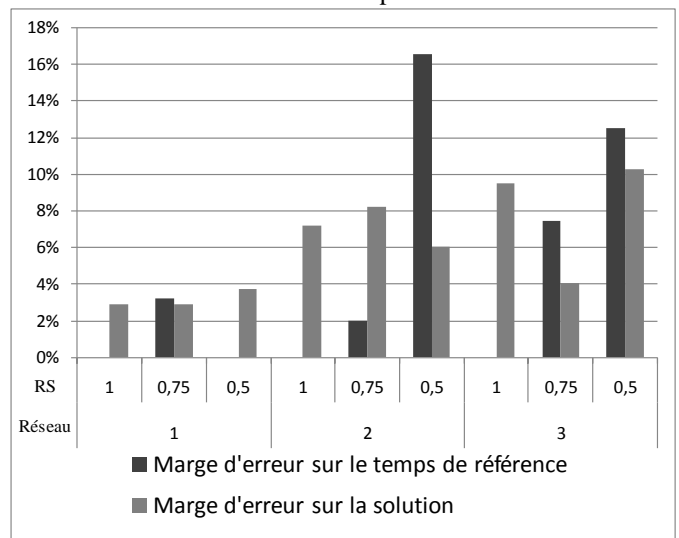


Figure 7 : Marges d'erreur associées à l'heuristique

On remarque aussi que l'utilisation de l'algorithme permet dans certains cas de compenser la faible performance de l'outil d'ordonnancement de projet, comme par exemple pour l'instance 2 avec RS=0,5. La marge d'erreur observée passe de 16% pour le temps de référence à 6% pour la solution. Aussi, dans nos résultats, l'utilisation de l'algorithme induit une erreur comprise entre 3% et 10% par rapport à l'optimal.

Le tableau 3 illustre le nombre d'itérations nécessaires pour atteindre la meilleure solution à l'aide de l'algorithme. Chaque itération correspond à un ordonnancement de projet. On observe que le nombre d'itération est assez faible et aléatoire. Aussi, l'application d'une heuristique nous permet de s'affranchir de la complexité du problème RCPSP multimode, le temps de résolution ne sera pas directement lié avec la taille de problème. L'application de l'heuristique nous permet donc d'obtenir une solution de bonne qualité dans un temps raisonnable.

Tableau 3 : Nombre d'itérations nécessaires à l'obtention de la solution

	Réseau	Nombre d'itération		
		1	2	3
RS	1	13	67	94
	0,75	14	5	135
	0,5	17	22	3

Nous avons aussi envisagé une autre règle de priorisation des activités. On classe les activités en fonction de leurs marges totales, de leurs nombres de successeurs et enfin de la quantité de ressources nécessaire à l'exécution normale de l'activité. Les résultats obtenus sont similaires aux résultats obtenus précédemment, seule une instance donne un résultat supérieur au résultat précédent, les autres étant égales. Cependant cette seconde configuration a une influence sur le temps de résolution, augmentant en général le nombre d'itération nécessaires à l'obtention la solution.

Cette remarque nous montre l'importance du choix des règles de priorité, et l'importance d'envisager plusieurs règles dans des travaux futurs afin d'obtenir des résultats de meilleure qualité.

5 CONCLUSION

Le chevauchement d'activités est une des stratégies d'accélération de projet les plus utilisées, que ce soit lors de la planification initiale, ou lors d'une re-planification suite à des retards durant l'exécution du projet. Le chevauchement comprend des risques car l'activité en aval ne commence pas avec des informations confirmées et finalisées. Cependant, en pratique, les retouches nécessaires pour s'accommoder au changement dans l'information transmise par l'activité en amont ne sont pas toujours prises en compte. La littérature a beaucoup étudié la relation entre la durée de chevauchement et la durée de retouche nécessaire en fonction des caractéristiques d'un couple d'activités mais peu d'articles ont incorporé le chevauchement au problème d'ordonnancement. Cependant la résolution d'un problème de type RCPSP nécessite un temps de calcul conséquent pour des problèmes complexe, ce papier propose donc d'utiliser une méthode heuristique pour obtenir une solution de qualité acceptable en un temps réduit.

Nous avons joint le chevauchement d'activités à l'ordonnancement de projet avec les hypothèses suivantes : (1) les activités chevauchables sont identifiées préalablement, (2) les échanges d'information sont unidirectionnels, (3) les échanges d'information sont gratuits et instantanés, (4) les durées des chevauchements possibles sont définies par des modes (de manière discrète) et (5) les retouches sont estimées au préalable pour chaque mode de chevauchement. La contribution principale de ce papier est de présenter un processus d'accélération de projet et une heuristique permettant d'obtenir une solution de qualité acceptable en un temps réduit. L'heuristique développée s'inspire des méthodes de compression d'activités et a été implantée dans un progiciel de gestion de projet, MS Project. Nous avons aussi développé un modèle de programmation linéaire qui nous permet de comparer et d'évaluer la qualité des solutions obtenues.

Nos résultats montrent les limites d'un modèle de résolution linéaire pour une application industrielle et illustrent la

pertinence d'une approche heuristique. Il serait intéressant par la suite de tester plusieurs règles de priorité pour l'algorithme afin de sélectionner la meilleure solution. Une implantation sur des projets réels de grande envergure permettrait d'évaluer avec plus de certitude l'applicabilité de notre processus et l'efficacité de l'heuristique proposée. Enfin, le développement de métaheuristiques permettrait de s'affranchir des problèmes de minimum local et d'obtenir ainsi des solutions de meilleure qualité.

6 RÉFÉRENCES

- Bogus S. M., Molenaar K. R., Diekmann J. E., (2005) Concurrent engineering approach to reducing design delivery time, *Journal of Construction Engineering and Management-Asce*, 131(11), pp. 1179-1185.
- Browning, T.R., (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions, *IEEE Transactions on engineering management*, 48(3), pp. 292-306.
- Gerk J. E. V., Qassim R. Y., (2008) Project Acceleration via Activity Crashing, Overlapping, and Substitution, *Ieee Transactions on Engineering Management*, 55(4), pp. 590-601.
- Kolisch, R., Sprecher, A., (1996) PSPLIB – A project scheduling problem library, *European Journal of Operational Research*, 96(1), pp. 205-216.
- Kolisch, R., (1999) Resource allocation capabilities of commercial project management software packages, *Interfaces*, 29(4), pp. 19-31.
- Krishnan V., Eppinger S. D., Whitney D. E., (1997) A model-based framework to overlap product development activities, *Management Science*, 43(4), pp. 437-451.
- Krishnan, V., (1996) Managing the simultaneous execution of coupled phases in concurrent product development, *IEEE Transactions on Engineering Management*, 43(2), pp. 210-217.
- Lin J., Chai K. H., Brombacher A. C., Wong Y. S., (2009) Optimal overlapping and functional interaction in product development, *European Journal of Operational Research*, 196(3), pp. 1158-1169.
- Maheswari J. U., Varghese K., (2005) Project Scheduling using Dependency Structure Matrix, *International Journal of Project Management*, 23(3), pp. 223-230.
- Steward D. V., (1981) The Design Structure System: A Method for Managing the Design of Complex Systems, *Ieee Transactions on Engineering Management*, EM-28(3), pp. 71-74.
- Wang, J. T., Lin, Y. I., (2009) An overlapping process model to assess schedule risk for new product development, *Computers & Industrial Engineering*, 57(2), pp. 460-474.