

Une aide à la décision pour l'ordonnancement robuste en contexte mono-ressource : un compromis flexibilité / performance

H.-T La, J.-L Santamaria, and C. Briand

LAAS-CNRS, 7 avenue du Colonel Roche, 31077 Toulouse, France
{htla,jlsantam,briand}@laas.fr

Résumé Cet article présente une approche d'aide à la décision pour l'ordonnancement robuste, dans le cadre de problèmes d'ordonnancement mono-ressource avec dates de disponibilité et dates d'échéance. L'objectif est de fournir au décideur des informations riches permettant de le guider pour la recherche d'un compromis flexibilité / performance utile vis-à-vis de la robustesse. La flexibilité considérée est de nature séquentielle ce qui permet de mettre en évidence, pour un problème donné, un ensemble de solutions. Elle est introduite par utilisation d'un ordre partiel dominant dont la structure particulière permet, en évitant l'énumération, d'une part de dénombrer les séquences de l'ensemble et d'autre part, de déterminer les performances au mieux et au pire, en terme de retard algébrique de cet ensemble. Les principes et mécanismes de l'aide à la décision sont présentés et illustrés via l'utilisation d'un outil logiciel d'aide à la décision, baptisé ADOR.

Mots-Clefs. Ordonnancement robuste; Problème à une machine; Aide à la décision.

1 Introduction

Une solution d'ordonnancement ne laisse classiquement aucune flexibilité quant à l'ordre d'exécution des tâches sur chaque ressource. Pourtant, introduire de la flexibilité séquentielle dans une solution d'ordonnancement, afin de disposer d'un ensemble de solutions, peut s'avérer judicieux en particulier dans un contexte opérationnel : il est alors possible de réagir aux aléas survenant lors de la mise en œuvre, sans requérir des efforts de calcul supplémentaires, en passant d'une solution de l'ensemble à une autre de façon opportune.

Cette approche a été largement discutée et utilisée dans de nombreux travaux. Qualifiée de *robuste* dans la littérature, elle est généralement rangée en tant que telle parmi les approches dites *proactives-réactives* [8,17,18]. De telles approches intègrent deux niveaux de décision. Au niveau le plus haut, un ensemble flexible de solutions est déterminé hors ligne, en prenant plus ou moins explicitement en compte l'existence d'incertitudes au sein des données initiales du problème. Le niveau bas, en exploitant l'ensemble de solutions déterminé hors-ligne, réagit en

temps réel aux aléas et tente de maîtriser les éventuelles dégradations de performance. Les approches proactives-réactives se distinguent notamment par la nature des incertitudes prises en compte et le type de flexibilité manipulée : temporelle et/ou séquentielle. La flexibilité temporelle, déjà naturellement présente au sein d'une solution d'ordonnancement, est volontairement augmentée dans le cadre de l'ordonnancement robuste soit par introduction de temps morts entre certaines tâches [16,21], soit par surdimensionnement des durées opératoires [15] ou des marges temporelles des activités [9,19]. Cet article s'intéresse à un autre type de flexibilité, de nature séquentielle, et sauf ambiguïté, le mot flexibilité sera exclusivement réservé pour désigner une flexibilité de ce type.

Afin d'introduire de la flexibilité séquentielle dans une solution d'ordonnancement, un ordre partiel est souvent utilisé. Un ordre partiel, laissant libres certaines décisions de séquençage sur les ressources, permet en effet de caractériser un ensemble de solutions en évitant la complexité d'une énumération. Afin de pouvoir être utilisé au sein d'une procédure d'ordonnancement, il est souhaitable que cet ordre partiel permette également une détermination aisée de la performance *au pire* de l'ensemble de solutions qu'il caractérise. En effet, ces solutions ne possédant pas nécessairement des performances identiques du point de vue de la fonction objectif considérée, savoir déterminer la solution la pire, en un temps de calcul acceptable, est important du point de vue de l'optimisation.

Un ordre partiel souvent manipulé en ordonnancement est celui relatif au concept de *groupe de tâches*. Ce concept permet en effet de caractériser une famille de solutions en associant à chaque ressource une *séquence de groupes* [10,22]. Les groupes sont donc ordonnés mais les tâches de chaque groupe sont totalement permutables (ou partiellement permutables [2,23]), ce qui définit bien un ordre partiel. De plus, la structure particulière de cet ordre partiel rend possible, en temps polynomial, le calcul de la pire des solutions contenue dans l'ensemble, vis-à-vis du retard algébrique L_{max} . Cet ordre partiel a donc été utilisé soit pour calculer en ligne une séquence de groupes [3,20], soit pour introduire de la flexibilité séquentielle au sein d'une solution initiale (éventuellement optimale) [13]. Comme il est également facile de dénombrer les solutions qu'une séquence de groupes caractérise, la recherche d'un compromis flexibilité / performance est aussi envisageable. En effet, dans le cadre d'une procédure d'ordonnancement proactive, et ainsi que le soulignent les auteurs dans [12,13], la détermination d'un ensemble de solutions le plus riche possible induit généralement une détérioration de la performance; inversement, la recherche de l'optimalité tend à réduire la flexibilité de l'ensemble de solutions. Il est donc en pratique intéressant de rechercher un compromis entre une solution peu flexible, mais très performante, et une solution très flexible, mais peu performante.

C'est sur ce problème de recherche de compromis flexibilité / performance que cet article se focalise. Nous nous intéressons plus particulièrement au problème mono-ressource où chaque travail à ordonner $j \in T$ est caractérisé par une date de début au plus tôt r_j , une date d'échéance d_j et une durée opératoire p_j . La fonction objectif analysée est la minimisation du plus grand retard algébrique. Ce problème, classiquement noté $1|r_j|L_{max}$, est très fréquem-

ment abordé dans la littérature, notamment parce que les méthodes permettant de le résoudre peuvent être utilisées au sein de méthodes traitant des problèmes d'ordonnancement plus complexes (job shop).

L'approche proposée dans cet article est fondée sur l'utilisation d'un ordre partiel dominant, différent de celui relatif à la notion de groupe de tâches, déterminé par analyse d'une structure d'intervalles associée au problème d'ordonnancement $1|r_j|L_{max}$. La première partie de cet article présente brièvement les notions utilisées pour la détermination de l'ordre partiel, qui est défini dans la deuxième partie. Nous décrivons ensuite les principes et les mécanismes mis en jeu au sein de la procédure d'aide à la décision que nous proposons. Enfin, la dernière partie présente les caractéristiques d'un outil logiciel, baptisé ADOR (Aide à la Décision pour l'Ordonnancement Robuste), mettant en œuvre l'aide à la décision.

2 Structures d'intervalles, sommets, bases et pyramides

Une structure d'intervalles est définie par un couple $\langle I, C \rangle$ où $I = i_1, \dots, i_n$ est un ensemble d'intervalles et C est un ensemble de contraintes sur $I \times I$. Chaque intervalle i_j est défini par un couple de points x_j et y_j tel qu'une relation d'ordre $x_j \prec_R y_j$ quelconque soit vérifiée. Une contrainte entre deux intervalles i_j et i_k peut être exprimée soit en spécifiant un ordre total entre les points définissant ces intervalles soit, de façon équivalente, en utilisant une des relations de l'algèbre de Allen [1], récapitulées sur la figure 1 (où la relation d'ordre \prec_R porte sur le temps).

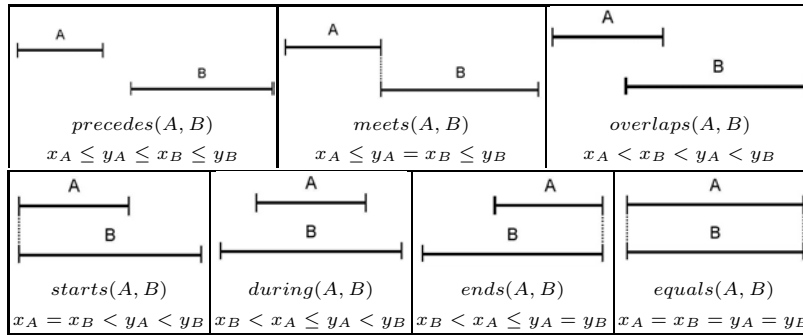


FIG. 1. Algèbre de Allen

Sur la base des relations de Allen, deux types particuliers d'intervalles peuvent être mis en évidence : l'intervalle de type *sommet* et celui de type *base*.

Définition 1 Un intervalle s est dit *sommet* d'une structure d'intervalles $\langle I, C \rangle$ si $\nexists i \in I$ tel que $during(i, s)$.

Définition 2 Un intervalle b est dit base d'une structure d'intervalles $\langle I, C \rangle$ si $\nexists i \in I$ tel que $\text{during}(b, i)$.

En utilisant ces notions de sommet et base, les notions de *sommet-pyramide* (notée *s-pyramide*) et de *base-pyramide* (notée *b-pyramide*) peuvent être définies [14].

Définition 3 Une *s-pyramide* P_s , associée au sommet s d'une structure d'intervalles $\langle I, C \rangle$, est le sous-ensemble d'intervalles $i \in I$ tel que $\text{during}(s, i)$.

Définition 4 Une *b-pyramide* P_b , associée à la base b d'une structure d'intervalles $\langle I, C \rangle$, est le sous-ensemble d'intervalles $i \in I$ tel que $\text{during}(i, b)$.

Les définitions 1 à 4 sont illustrées sur la structure d'intervalles représentée sur la figure 2. On identifie les sommets $\{C, D, E\}$ qui caractérisent respectivement les *s-pyramides* : $P_C = \{A, B, G\}$, $P_D = \{G\}$ et $P_E = \{F, G\}$. De même, on identifie les bases $\{A, B, F, G\}$ et les *b-pyramides* $P_A = \{C\}$, $P_B = \{C\}$, $P_F = \{E\}$, $P_G = \{C, D, E\}$.

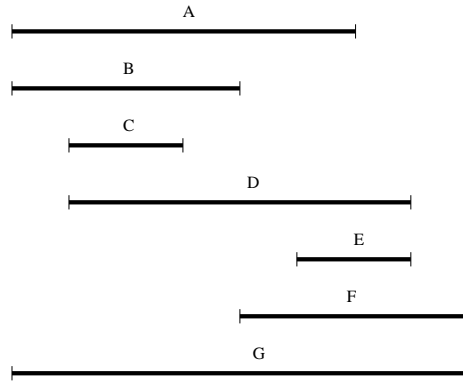


FIG. 2. Exemple d'intervalles

Ce sont les notions de sommet et de *s-pyramide*, appliquées au problème d'ordonnement mono-ressource, que nous utilisons dans cet article afin de caractériser un ordre partiel dominant. Remarquons toutefois que les notions de bases et de *b-pyramides*, bien que non utilisées par la suite, peuvent également être mise à profit pour caractériser des ordres partiels dans le cadre d'autres problèmes d'ordonnement [6].

3 Un ordre partiel dominant : le théorème des pyramides

Cet article utilise un ordre partiel, défini par un théorème de dominance démontré dans les années quatre-vingt [11]. La dominance envisagée par les

auteurs est relative soit à l'admissibilité, soit aux critères d'optimisation T_{\max} et L_{\max} . Le corps d'hypothèses considéré prend en compte uniquement l'ordre relatif des dates de début au plus tôt r_j et de fin au plus tard d_j des travaux. Les durées opératoires p_j ainsi que les valeurs explicites des r_j et d_j de chaque travail $j \in T$ ne sont pas considérées. Les résultats suivants restent donc valides quelles que soient les durées opératoires et quelles que soient les valeurs de r_j et d_j dans le respect de l'ordre relatif décrit dans le corps d'hypothèses.

On considère la structure d'intervalles $\langle I, C \rangle$ où un intervalle $i_j \in I$, caractérisé par une date de début r_j et une date de fin d_j , est associé à chaque travail $j \in T$ du problème.

On suppose que les sommets de la structure d'intervalles ainsi obtenue sont indicés par r_j croissant ou par d_j croissant en cas d'égalité. Ceci est équivalent à affirmer que, s_α et s_β étant deux sommets de la structure d'intervalles, $\alpha < \beta$ si et seulement si $(r_{s_\alpha} \leq r_{s_\beta}) \wedge (d_{s_\alpha} \leq d_{s_\beta})$. Si deux sommets possèdent des dates de début au plus tôt et de fin au plus tard identiques, alors ils peuvent être indicés de façon quelconque. La s -pyramide notée P_α désigne la pyramide caractérisée par le sommet s_α . On note $u(i_j)$ (resp. $v(i_j)$) l'indice de la première s -pyramide à laquelle le travail i_j appartient (resp. l'indice de la dernière pyramide à laquelle le travail i_j appartient). Un ordre partiel dominant peut alors être caractérisé grâce au théorème suivant.

Théorème 1 *Un ensemble dominant de séquences peut être constitué par les séquences telles que :*

1. *les sommets sont ordonnés dans l'ordre de leur indice ;*
2. *avant le premier sommet, seuls sont placés des travaux appartenant à la première s -pyramide, rangés dans l'ordre croissant de leur r_j ou, en cas d'égalité, dans un ordre arbitraire ;*
3. *après le dernier sommet, seuls sont placés des travaux appartenant à la dernière s -pyramide, rangés dans l'ordre croissant de leur d_j ou, en cas d'égalité, dans un ordre arbitraire ;*
4. *entre deux sommets s_k et s_{k+1} , sont placés en premier des travaux appartenant à la s -pyramide P_k et n'appartenant pas à P_{k+1} dans l'ordre croissant de leur d_j (dans un ordre arbitraire en cas d'égalité), puis des travaux communs aux deux pyramides P_k et P_{k+1} dans un ordre arbitraire, et enfin des travaux appartenant à la pyramide P_{k+1} et n'appartenant pas à P_k dans l'ordre croissant de leur r_j (dans un ordre arbitraire en cas d'égalité).*

Une propriété intéressante du théorème précédent est qu'il permet d'évaluer la cardinalité de l'ensemble dominant \mathcal{S}_{dom} grâce à la formule $\text{card}(\mathcal{S}_{\text{dom}}) = \prod_{q=1}^N (q+1)^{n_q}$ où n_q désigne le nombre d'intervalles non sommets appartenant exactement à q pyramides et N le nombre total de pyramides. Remarquons que cette formule ne prend pas en compte les possibilités de permutation des tâches placées entre deux sommets s_k et s_{k+1} et appartenant aux s -pyramides P_k et P_{k+1} .

Considérons par exemple, le problème d'ordonnement à quatre travaux du tableau 1. La structure d'intervalles associée à ce problème est représentée sur la figure 3. Elle contient deux sommets : $s_1 = 1$ et $s_2 = 4$. Ces deux sommets caractérisent deux pyramides : $P_1 = \{2, 5\}$ et $P_2 = \{2, 3\}$. Remarquons qu'en cohérence avec la définition de la notion de pyramide, un sommet n'appartient pas à la pyramide qu'il caractérise.

Travaux	1	2	3	4	5
r_i	6	1	21	24	4
d_i	13	37	33	31	17
p_i	4	5	8	6	7

TAB. 1. Un problème à une machine $1|r_j|L_{\max}$

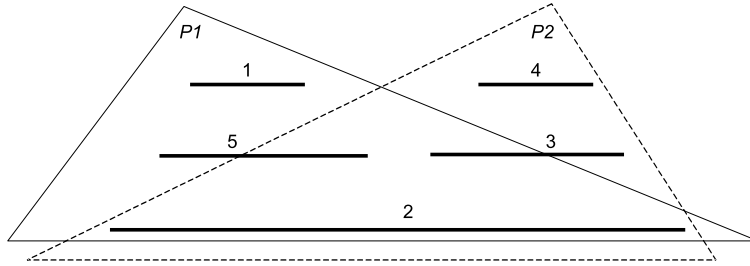


FIG. 3. Structure d'intervalles du problème du tableau 1

L'application du théorème à l'exemple précédent permet de caractériser un ensemble dominant \mathcal{S}_{dom} de cardinalité $\text{card}(\mathcal{S}_{\text{dom}}) = (1 + 1)^2 \cdot (2 + 1)^1 = 12$, constitué des séquences (cf. figure 4) :

- $2 \prec 5 \prec 1 \prec 3 \prec 4,$ $2 \prec 5 \prec 1 \prec 4 \prec 3,$ $5 \prec 1 \prec 2 \prec 3 \prec 4,$ $5 \prec 1 \prec 2 \prec 4 \prec 3,$
 $5 \prec 1 \prec 3 \prec 4 \prec 2,$ $5 \prec 1 \prec 4 \prec 3 \prec 2,$ $2 \prec 1 \prec 5 \prec 3 \prec 4,$ $2 \prec 1 \prec 5 \prec 4 \prec 3,$
 $1 \prec 5 \prec 2 \prec 3 \prec 4,$ $1 \prec 5 \prec 2 \prec 4 \prec 3,$ $1 \prec 5 \prec 3 \prec 4 \prec 2,$ $1 \prec 5 \prec 4 \prec 3 \prec 2.$

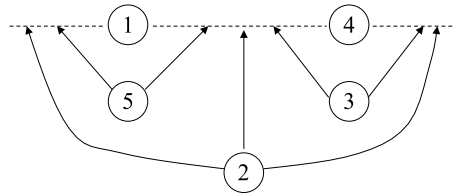


FIG. 4. Ensemble dominant de séquences associé à l'exemple

Conformément à la propriété de dominance, toute séquence $s \notin \mathcal{S}_{\text{dom}}$ possède une qualité inférieure ou égale à au moins une séquence de \mathcal{S}_{dom} . Afin d'évaluer plus précisément la qualité de l'ensemble dominant, il est intéressant de déterminer pour chaque travail j les retards algébriques *au mieux* (noté L_j^{\min}) et *au pire* (noté L_j^{\max}) parmi toutes les séquences contenues dans l'ensemble. Pour cela, les valeurs explicites de r_j , d_j et p_j doivent être prises en compte. Le calcul de L_j^{\min} ou de L_j^{\max} peut être réalisé en temps polynomial ($O(n \log n)$ où n est le nombre de travaux), comme indiqué dans [4,5], sans énumération de l'ensemble des séquences (ce qui est une propriété intéressante de ce type de structure pyramidale). La détermination des valeurs de L_j^{\min} et L_j^{\max} de chaque travail permet de construire un *diagramme de retards*, comme celui illustré sur la figure 5 pour l'exemple précédent.

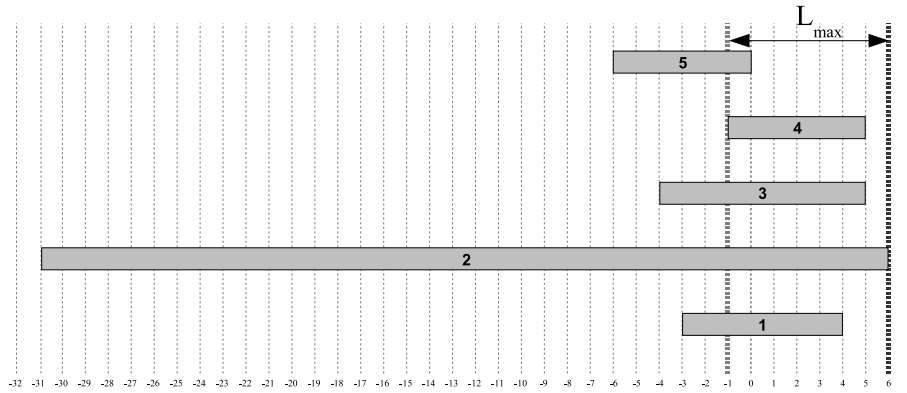


FIG. 5. Diagramme de retards associé à l'exemple

Une information essentielle donnée par le diagramme de retards est un encadrement de la performance vis-à-vis du critère L_{\max} : $\max_{j \in T}(L_j^{\min}) \leq L_{\max} \leq \max_{j \in T}(L_j^{\max})$. Cette information a été exploitée au sein d'une procédure de séparation et évaluation progressive [4] qui permet d'éliminer de l'ensemble initial de séquences dominantes toutes les séquences non optimales. Toutefois, bien que cette méthode ait un intérêt certain du point de vue théorique, il peut être plus intéressant en pratique de déterminer un ensemble de solutions de cardinalité la plus grande possible possédant une qualité acceptable, pas nécessairement optimale. Il s'agit donc de rechercher un compromis entre le nombre de séquences caractérisées et la valeur au pire du retard algébrique.

4 Principes d'une aide à la décision

L'aide à la décision a pour but de guider le décideur pour l'exploration de l'ensemble dominant initial associé au problème, de sorte à déterminer un compromis flexibilité/performance acceptable. Le diagramme de retards a été choisi

comme objet d'interaction. En effet, il permet de juger de l'acceptabilité du compromis flexibilité / performance puisqu'il indique pour chaque travail les retards L_j^{\min} et L_j^{\max} , et qu'un indicateur de flexibilité, précisant le nombre de séquences contenues dans l'ensemble dominant, peut être calculé.

L'aide à la décision consiste, dans le cas où le compromis ne satisfait pas le décideur, à lui donner la possibilité de sélectionner une borne, L_j^{\min} ou L_j^{\max} d'un travail, pour indiquer qu'il désire la réduire ou l'augmenter, en précisant la nouvelle valeur souhaitée (appelée *consigne* dans la suite du texte). Le principe de l'aide à la décision consiste à coupler une telle action à une modification de la structure d'intervalles, comme indiqué dans les paragraphes suivants.

La réduction ou l'augmentation du retard L_j^{\max} est respectivement réalisée en décrémentant ou en incrémentant progressivement la valeur de $v(j)$, en vérifiant toujours l'inégalité $v_0(j) \geq v(j) \geq u(j)$. La décrémentant de $v(j)$ conduit à la mise à jour de la valeur de d_j de sorte que $d_j \leftarrow \max(d_{s_{v(j)-1}}, d_{s_{u(j)}})$. L'incrémentant de $v(j)$ est elle aussi réalisée en mettant à jour la valeur de d_j de sorte que $d_j \leftarrow \min(d_{s_{v(j)+1}}, d_j^0)$, où d_j^0 est la valeur initiale de d_j .

Similairement, la réduction ou l'augmentation du retard L_j^{\min} est respectivement réalisée en décrémentant ou en incrémentant progressivement la valeur de $u(j)$, en vérifiant l'inégalité $u_0(j) \leq u(j) \leq v(j)$. L'incrémentant de $u(j)$ conduit à la mise à jour de la valeur de r_j de sorte que $r_j \leftarrow \min(r_{s_{u(j)+1}}, r_{s_{v(j)}})$. La décrémentant de $u(j)$ est elle aussi réalisée en mettant à jour la valeur de r_j de sorte que $r_j \leftarrow \max(r_{s_{u(j)-1}}, r_j^0)$, où r_j^0 est la valeur initiale de r_j .

L'incrémentant ou la décrémentant des valeurs de $u(j)$ ou $v(j)$, est automatiquement réitérée jusqu'à ce que la valeur de L_j^{\min} ou L_j^{\max} soit la plus proche possible de la consigne donnée par le décideur. Remarquons que, quelle que soit l'action envisagée, elle se traduit toujours par une augmentation ou une diminution des bornes r_j ou d_j associées au travail j considéré. Une nouvelle structure d'intervalles est donc obtenue telle que l'ensemble dominant de séquences correspondant est inclus dans l'ensemble dominant initial (du fait de la façon d'actualiser les valeurs de r_j et d_j). Notons également que l'augmentation ou la réduction d'un retard au pire ou au mieux d'un travail j peut avoir des conséquences sur les retards des autres travaux du fait de la méthode de calcul des L_j^{\min} et L_j^{\max} . De façon générale, le resserrement de l'intervalle de retard d'un travail j réduit la flexibilité et peut entraîner le resserrement d'intervalles de retard d'autres travaux. Inversement, l'élargissement d'un intervalle de retard augmente la flexibilité et peut entraîner l'élargissement d'intervalles de retard d'autres travaux.

5 Un outil d'aide à la décision : ADOR

Afin de valider les principes de l'aide à la décision, une interface logicielle, baptisée ADOR (Aide à la Décision pour l'Ordonnancement Robuste), a été élaborée en Ada. Pour l'aspect graphique, la bibliothèque gratuite GtkAda a naturellement été adoptée.

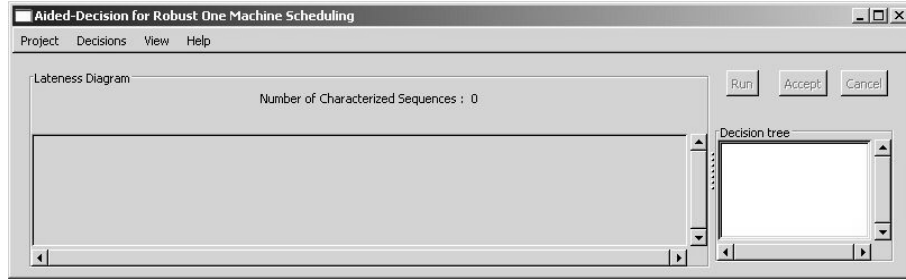


FIG. 6. Fenêtre principale d'accueil

Au lancement, la fenêtre d'accueil, représentée sur la figure 6, apparaît. Elle est munie de deux régions intitulées *Lateness diagram* et *Decision tree* dont les rôles sont explicités dans la suite du texte. Pour commencer, il est nécessaire soit de saisir un nouveau problème, ce qui crée un projet, soit d'ouvrir un projet existant (utilisation du menu *project*). La figure 7 représente l'interface graphique associée à la saisie du problème du tableau 1.

	Release Date	Due Date	Processing Time
Job 1	6	13	4
Job 2	1	37	5
Job 3	21	33	8
Job 4	24	31	6
Job 5	4	17	7

FIG. 7. Saisie des données d'un problème

Après validation de la saisie, le diagramme de retards initial est automatiquement calculé et affiché ainsi qu'illustré sur la figure 8. Le nombre de séquences contenues dans l'ensemble dominant initial est également indiqué au dessus du diagramme de retards). Pour représenter les retards, une règle graduée, munie de deux curseurs positionnés à ces extrémités, est associée à chaque travail. Les curseurs gauche et droit d'un travail j matérialisent respectivement les valeurs de L_j^{\min} et L_j^{\max} . Une région ombrée, commune à l'ensemble des règles, indique l'intervalle dans lequel est situé le retard optimal ($L^* \in [\max(L_j^{\min}), \max(L_j^{\max})]$), ici l'intervalle $[-1, 6]$. Le sous-menu *search optimal L^** du menu *Decisions* détermine, si l'utilisateur le souhaite, la valeur exacte du retard optimal (utilisation de la procédure de Carlier [7]). Dans ce cas, la région est réduite à une seule valeur, 4 dans notre cas (cf. figure 9).

Afin de permettre l'interaction, les curseurs de chaque règle peuvent être déplacés à l'aide de la souris. Cette interaction, permet au décideur d'indiquer

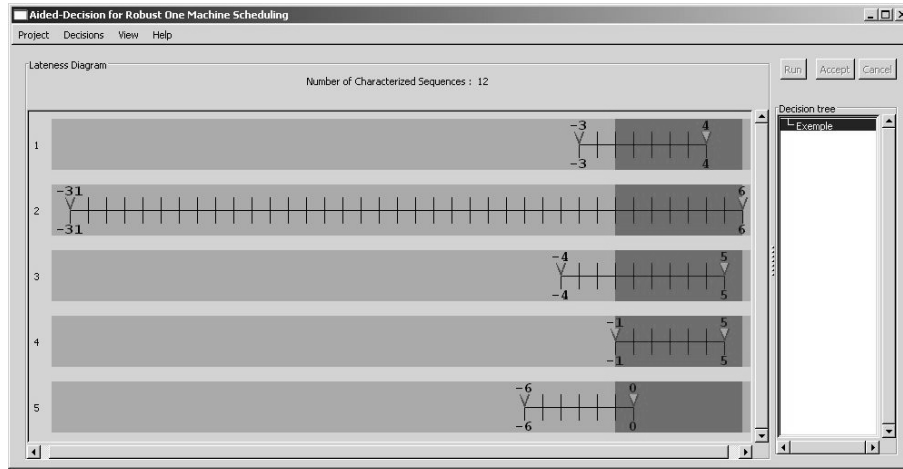


FIG. 8. Affichage du diagramme des retards initial

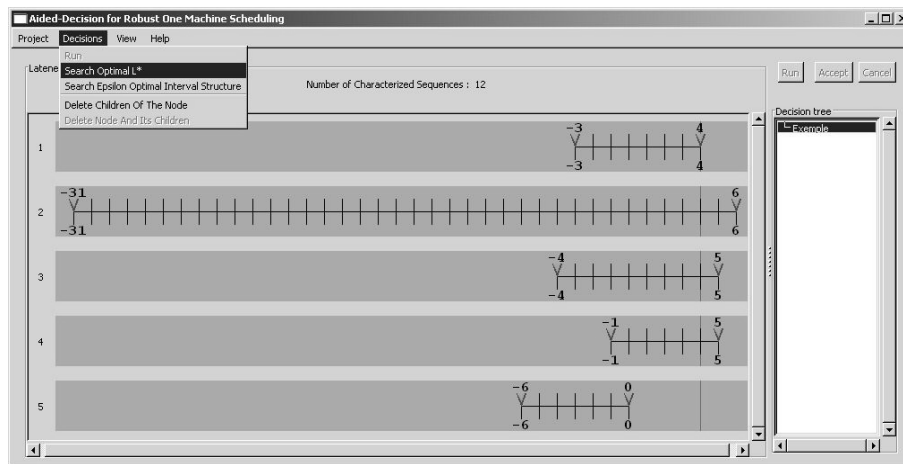


FIG. 9. Détermination du retard optimal

qu'il souhaite modifier un retard au mieux ou au pire associé à un travail. Supposons par exemple que le décideur désire modifier le retard au mieux du travail 2 de sorte qu'il soit le plus proche possible de la valeur 12. Dans ce cas, il doit sélectionner le curseur gauche de la règle associée à ce travail, puis le faire glisser à la valeur 12, et enfin cliquer sur le bouton *run*. La fenêtre est alors actualisée (cf. figure 10) et affiche pour chaque travail les anciennes et nouvelles valeurs des retards (un curseur grisé indique la valeur initiale d'un retard lorsqu'elle a été modifiée). On constate que, conformément à la procédure de mise à jour de $u(2)$ (voir section précédente), la valeur la plus proche de la consigne 12 (représentée par un marqueur à cette abscisse) est 3 (nouvelle position du curseur) et il reste 4 séquences dans l'ensemble dominant sur les 12 initiales. De plus, les retards au pire des travaux 1 et 4 ont été réduits.

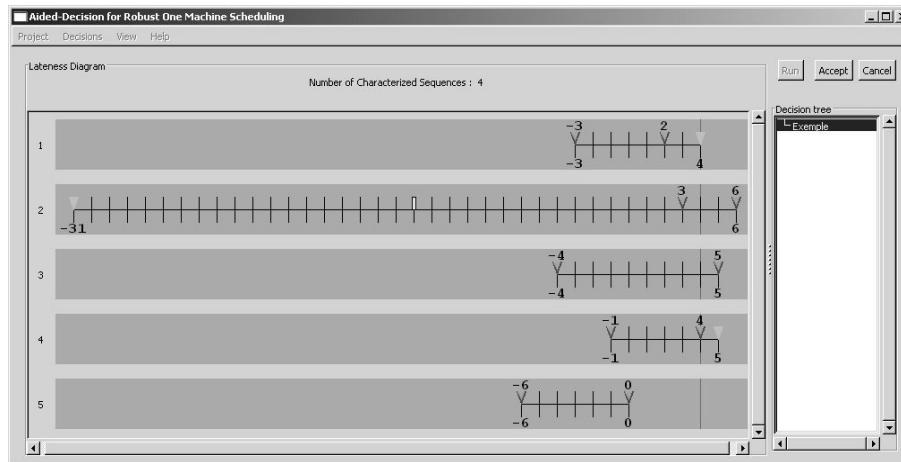


FIG. 10. Exemple d'interaction

L'utilisateur doit alors indiquer s'il souhaite ou non mémoriser la structure d'intervalles correspondant à ce nouveau diagramme de retards, en cliquant sur les boutons *Cancel* ou *Accept*. Ainsi qu'illustré sur la figure 11, l'appui sur le bouton *Accept* provoque d'une part la création d'un nouveau nœud dans l'arbre de décision, et d'autre part la validation des nouvelles positions des curseurs.

En se plaçant sur un nœud quelconque de l'arbre et en modifiant un retard, l'utilisateur peut répéter l'opération, et affiner ainsi sa stratégie de recherche, en développant soit en séquence, soit en concurrence un ensemble de décisions (cf. figure 12). Le diagramme des retards affiché correspond à celui associé au nœud sélectionné dans l'arbre de décision. L'arborescence de l'arbre de décision peut être gérée (suppression de nœud ou de branches) en utilisant les sous-menus du menu *Decisions*.

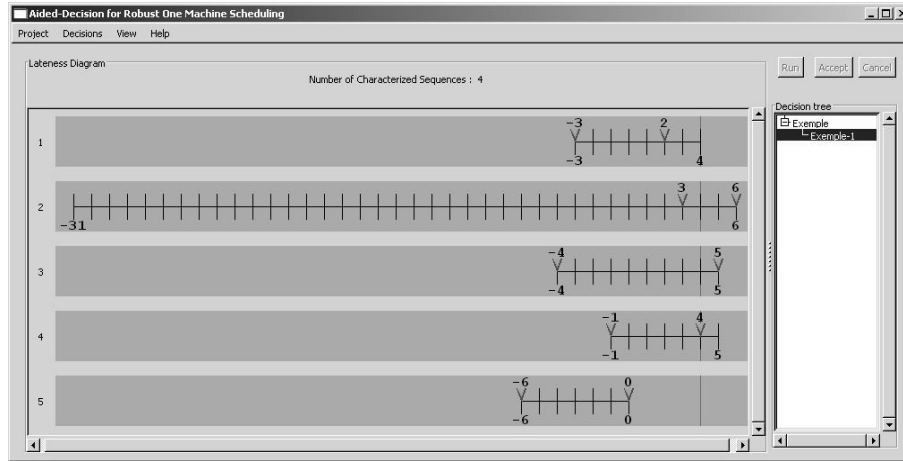


FIG. 11. Validation d'une décision

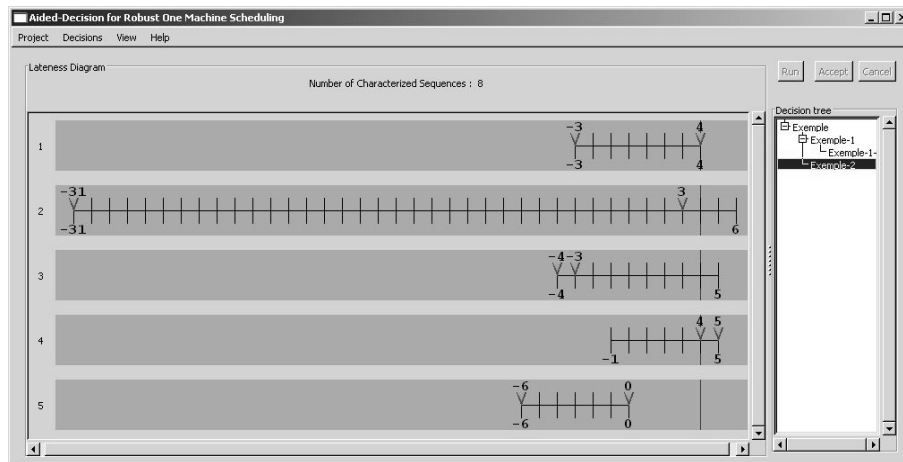


FIG. 12. Exemple de décisions concurrentes

6 Conclusion

Dans cet article, une aide à la décision pour un compromis flexibilité / performance a été proposée dans le cadre du problème mono-ressource. Cette aide à la décision est fondée sur l'utilisation d'un ordre partiel dominant permettant de caractériser une famille de solutions. L'intérêt de cet ordre partiel réside dans le fait qu'il permet de déterminer, sans énumération des solutions, à la fois un indicateur de flexibilité (nombre de solutions) et un indicateur de performance (retard algébrique au mieux et au pire). Les vœux du décideur relatif à la performance pour chaque travail sont pris en compte en agissant sur la structure d'intervalles associée au problème de façon à explorer l'ensemble dominant initial. À chaque itération, un diagramme de retards est produit et un nouvel indicateur de flexibilité est calculé afin de permettre au décideur d'adapter le compromis flexibilité / performance à sa guise.

Au delà de cette contribution, ce type d'approche montre l'intérêt d'ordres partiels dominants pour l'ordonnancement robuste, en particulier dans le cadre d'une aide à la décision. Dans notre cas, cet ordre est construit à partir d'une information relativement pauvre : l'ordre relatif des dates de disponibilité et des dates d'échéance des travaux. Il est donc relativement insensible aux variations des données du problème, dans la mesure où l'ordre relatif reste inchangé. Néanmoins, si cette dernière propriété est intéressante du point de vue de la robustesse, elle constitue aussi un obstacle pour pouvoir produire une aide à la décision plus riche. Dans la pratique, il serait en effet plus approprié de permettre à un décideur de dimensionner ses incertitudes, en associant par exemple des fenêtres de valeurs aux données r_j , d_j et p_j , et en lui montrant l'effet produit par une augmentation ou une diminution de ces fenêtres sur le compromis flexibilité / performance, et vice-versa.

Références

1. Allen, J.F. : Time and time again : The many ways to represent time. *International Journal of Intelligent Systems*, 6 :341-355 (1991)
2. Aloulou, M.A, Portmann, M-C. et Vignier, A. : Predictive-Reactive Scheduling for the Single Machine Problem. *Eighth International Workshop on Project Management and Scheduling (PMS'2002)*, 39-42, Valencia (Spain) (2002)
3. Billaut, J-C, et Roubellat, F. : A new method for workshop real time scheduling. *International Journal of Production Research* 34(6), 1555-1579 (1996)
4. Briand, C. et La, H.T. : Une procédure par séparation et évaluation progressive pour l'ordonnancement robuste de problème à une machine. *Actes de la Conférence Recherche Informatique Vietnam et Francophonie (RIVF'2003)*, 11-16, Hanoi (Vietnam) (2003)
5. Briand, C., La, H.T. et Erschler, J. : Une approche pour l'ordonnancement robuste de tâches sur une machine. *Actes de la 4ème Conférence Francophone de MOdélisation et SIMulation (MOSIM'03)*, 205-211, Toulouse (France)(2003)

6. Briand, C., La, H.T. et Erschler, J. : Robust scheduling for the two-machine flowshop problem. Proceedings of the 9th International Workshop on Project Management and Scheduling (PMS'2004), 371-374, Nancy (France) (2004)
7. Carlier, J. : The one-machine sequencing problem. European Journal of Operational Research, 11 :42-47 (1982)
8. Davenport, A.J. et Beck, J.C. : A survey of techniques for scheduling with uncertainty. Manuscrit non publié, disponible à l'adresse <http://www.eil.utoronto.ca/chris/chris.papers.html> (2000)
9. Davenport, A.J., Gefflot, C. et Beck, J.C. : Slack-based Techniques for Robust Schedules. Proceedings of the Sixth European Conference on Planning (ECP-2001), Toledo (Spain) (2001)
10. Demmou R. : Etude de familles remarquables d'ordonnements en vue d'une aide à la décision Thèse de doctorat, Université Paul Sabatier, Toulouse (France) (1977)
11. Erschler, J., Fontan, G., Merce, C. et Roubellat, F. : A New Dominance Concept in Scheduling n Jobs on a Single Machine with Ready Times and Due Dates. Operations Research, 31(1) :114-127 (1983)
12. Esswein, C., et Billaut, J.-C : Trade-off between flexibility and maximum completion time in the two-machine flowshop scheduling problem, International Symposium on Combinatorial Optimisation, Paris (France) (2002)
13. Esswein, C. : Un apport de flexibilité séquentielle pour l'ordonnement robuste. Thèse de doctorat. Université François Rabelais, Tours (France)(2003)
14. Esquirol, P. et Lopez, P. : Structures temporelles pour le problème d'ordonnement à une machine. Rapport LAAS (2002)
15. Gao, H., Fox, M.S, Chiang, W-Y. et Hikita, S. : Building Robust Schedules - An Empirical Study of Single Machine Scheduling with Uncertainty. Soumis à Management Science (1995)
16. Goldratt, E. : Critical Chain. Great Barrington : The North River Press (1997)
17. GOTha. : Flexibilité et Robustesse en Ordonnement. Bulletin de la ROADEF, 8 :10-12 (2002)
18. Herroelen, W. et Leus, R. : Robust and reactive project scheduling : a review and classification of procedures. International Journal of Production Research, 42(8) :1599-1620 (2004)
19. Herroelen, W. et Leus, R. : The construction of stable project baseline schedules. European Journal of Operational Research, 156 :550-565 (2004)
20. Le Gall, .A. : Un système interactif d'aide à la décision pour l'ordonnement et le pilotage en temps réel d'atelier. Thèse de doctorat. Université Paul Sabatier, Toulouse (France) (1989)
21. Mehta, S.V. et Uzsoy, R.H. : Predictable Scheduling of a Job Shop Subject to Breakdowns. IEEE Transactions on Robotics and Automation, 14(3) :365-378 (1998)
22. Thomas, V. : Aide à la décision pour l'ordonnement d'ateliers en temps réel. Thèse de doctorat, Université Paul Sabatier, Toulouse, France (1980)
23. Wu, S.D., Byeon, E-S. et Storer, R.H. : A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Operations Research, 47(1) :113-124 (1999)