

Habilitation à diriger des recherches

préparée au

Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

en vue de l'obtention du

Diplôme de l'Institut National Polytechnique de Toulouse

Approche par contraintes des problèmes d'ordonnancement et d'affectation : structures temporelles et mécanismes de propagation

par

Pierre LOPEZ

Chargé de Recherche au CNRS

Soutenue le 10 décembre 2003 au LAAS-CNRS
en Salle de Conférences

devant le Jury composé de :

Jacques Erschler	Professeur INSA, Toulouse
Gérard Fontan	Professeur INP, Toulouse
Claude Le Pape	Directeur R&D Ilog, Gentilly
Philippe Michelon	Professeur Univ. d'Avignon et des Pays de Vaucluse (rapporteur)
Eric Pinson	Professeur Univ. Catholique de l'Ouest, Angers (rapporteur)
Roman Słowiński	Professeur Univ. de Technologie de Poznan, Pologne (rapporteur)

Résumé

Ces travaux présentent une classe d'approches par contraintes pour aborder les problèmes d'ordonnancement de tâches et d'affectation de ressources. Ils présentent notamment les recherches menées sur la conception de mécanismes généraux de propagation de contraintes. Des rapprochements sont faits avec des techniques issues des problèmes de satisfaction de contraintes. Le cas des problèmes d'ordonnancement purement temporels est rapidement abordé, puis la présentation se consacre à des problèmes dans lesquels les tâches doivent respecter à la fois des contraintes temporelles et des contraintes de partage et d'affectation de ressources. On présente notamment une synthèse des travaux ayant pour but l'élaboration de règles de propagation faisant interagir des contraintes temporelles simples et des contraintes de ressources. Trois types de raisonnement sont étudiés et dans certains cas étendus par rapport à leur formulation originale. Le premier, les opérations locales, nécessite une analyse préalable des conflits entre tâches et infère des conditions de séquençement locales à une ressource. Le deuxième, les opérations globales, agit suivant le principe d'une contrainte posée sur l'ensemble du problème. Le troisième, le raisonnement énergétique, intègre simultanément les contraintes de temps et de ressources et permet de limiter l'intervalle de temps sur lequel peut être réalisée une tâche. L'utilisation et le contrôle des mécanismes de propagation de contraintes sont discutés. Ils peuvent servir à découvrir rapidement une inconsistance globale dans la formulation du problème, ou à simplifier sa résolution. Nous fournissons ainsi des éléments sur la décomposition et la stratégie de résolution des problèmes.

Mots-clés : ordonnancement, affectation, propagation de contraintes, structures temporelles

A Constraint-based Approach for Scheduling and Allocation: Temporal Structures and Propagation Mechanisms

Abstract:

This work presents a class of constraint-based approaches for task scheduling and resource allocation. The research aims at designing general constraint propagation mechanisms. Akin techniques coming from Constraint Satisfaction Problems are reviewed. Throughout the presentation are respectively presented pure temporal scheduling problems, then problems dealing with time and resource sharing constraints, finally problems considering also resource allocation constraints. The focus is done on works of which the objective is to design constraint propagation rules where time and resource constraints interact. Three kinds of methods are particularly studied, and extensions from their original formulation are proposed: local operations (such as "edge-finding" techniques), global operations ("shaving"), and energetic reasoning. The use and the control of the various mechanisms to detect a global inconsistency or simplify the resolution are discussed; thus, elements for problem decomposition as well as strategies for solving the problems under consideration are proposed.

Keywords: Scheduling, Allocation, Constraint Propagation, Temporal Structures

Table des matières

I – Descriptif des activités de recherche	1
Avant-propos	1
1 Introduction	1
2 Les problèmes d’ordonnancement	3
2.1 Définitions liminaires	3
2.2 Notations de base	3
2.3 Résolution	4
2.4 Modélisation adoptée	5
2.4.1 Problèmes de satisfaction de contraintes temporelles	5
2.4.2 Inégalités de potentiels	6
2.4.3 Représentation graphique	7
3 Propagation de contraintes	8
3.1 Généralités	8
3.2 Propagation de contraintes temporelles	9
3.3 Propagation de contraintes de partage de ressources	10
3.3.1 Introduction	10
3.3.2 Opérations locales	11
3.3.3 Opérations globales	14
3.3.4 Raisonnement énergétique	18
3.3.5 Interprétation sur un graphe de contraintes	23
3.4 Propagation de contraintes d’affectation de ressources	25
3.5 Propagation de contraintes et recherche de solutions	27
4 Structures temporelles pour la propagation	29
4.1 Groupes de tâches à rangs inclus	30

4.2	Treillis d'intervalles de tâches	31
5	Orientations de recherche	33
5.1	Bilan	33
5.2	Prospective	34
II	Références bibliographiques	37
III	Annexes	47
A	Curriculum vitæ	47
B	Publications	48
C	Participation à projets	55
D	Actions pour la communauté scientifique	58
D.1	Encadrement doctoral	58
D.2	Rayonnement scientifique	59
D.3	Gestion de la recherche	60
D.4	Enseignements dispensés	60
E	Conditions « non première / non dernière »	61
E.1	Travaux existants	61
E.2	Modification de l'algorithme calcLB2 de Nuijten	61
E.3	Un nouvel algorithme	63
E.4	Comparaison avec l'algorithme Baptiste-Le Pape	65
IV	Copie de quelques publications	67

I – Descriptif des activités de recherche

Avant-propos

Le thème de recherche majeur auquel je me suis intéressé durant les dix dernières années concerne les problèmes d’ordonnancement. Ce thème a été initialement développé, au sein du LAAS, dans le groupe de recherche « Organisation et Conduite de Systèmes Discrets » dont les objectifs sont la modélisation, l’analyse et la conduite des systèmes de production discrète, qu’il s’agisse de biens ou de services.

Je dois ici remercier plusieurs personnes, auprès de qui j’ai directement travaillé ou qui m’ont soutenu dans certaines actions (dont l’écriture de ce document !), notamment les membres du groupe « Modélisation, Optimisation et Gestion Intégrée de Systèmes d’Activités » où mon travail personnel est actuellement poursuivi, et plus particulièrement Marie-José Huguet, Jacques Erschler, Patrick Esquirol et François Roubellat. Je dois également citer Christian Artigues et Jean-Charles Billaut, anciens du LAAS et désormais personnalités actives en ordonnancement. Sans se lancer dans une dithyrambe personnalisée, je voudrais les remercier globalement, mais chaleureusement. Je n’oublie pas ma famille qui, forcément, a un peu pâti de la préparation de ce travail : mille baisers à *Muriel, Armand et Basile*.

Dans le document, les travaux originaux auxquels j’ai contribué sont signalés par une étoile dans la marge (j’ai emprunté ça à Thomas Schiex... qu’il veuille bien me pardonner !). De plus, lors de citations multiples de références, j’indique par une police différente comme celle-ci, celles où je suis intervenu. ★

1 Introduction

Dans le contexte actuel d’évolution des entreprises, il faut disposer de méthodes et d’outils de plus en plus performants pour l’organisation et la conduite de la production. Cette organisation de la production doit être vue au niveau de l’entreprise elle-même, mais aussi au niveau de la chaîne logistique dont elle constitue l’un des maillons. Pour atteindre ces objectifs, l’organisation repose en général sur la mise en œuvre d’un certain nombre de fonctions assurées par des méthodes d’autant plus sophistiquées que les systèmes d’information sont de plus en plus fiables et dynamiques. Parmi ces fonctions, l’ordonnancement joue un rôle essentiel.

La fonction ordonnancement vise à organiser l'utilisation des ressources technologiques et humaines présentes dans les ateliers ou les services de l'entreprise pour satisfaire soit directement les demandes des clients, soit les demandes issues d'un plan de production préparé par la fonction de planification de l'entreprise.

Compte tenu de l'évolution des marchés et de leurs exigences, cette fonction doit organiser l'exécution simultanée de multiples travaux sur des délais de réalisation de plus en plus courts, à l'aide de ressources plus ou moins polyvalentes disponibles en quantités limitées. Ceci constitue un problème complexe à résoudre¹. En cela, apporter des solutions efficaces et performantes aux problèmes d'ordonnancement constitue sûrement un enjeu économique important.

Nous présentons, dans la suite, des éléments de modélisation qui constituent la première étape dans la résolution d'un problème. La démarche suivie dans notre travail pour la résolution des problèmes d'ordonnancement s'inscrit dans la problématique d'une approche par contraintes. Au centre des préoccupations d'une telle approche, on trouve des recherches sur la conception de mécanismes efficaces et généraux de propagation de contraintes; ceci constitue le cœur de ce document. Des résultats spécifiques tirés de l'étude des problèmes d'ordonnancement et des résultats généraux issus de l'étude des problèmes de satisfaction de contraintes ont été regroupés.

On présente tout d'abord le cas des problèmes d'ordonnancement purement temporels. Il existe des algorithmes de propagation complets et de complexité acceptable pour des problèmes temporels simples, dans lesquels chaque contrainte limite la distance entre deux dates à un domaine connexe. Le traitement de contraintes temporelles plus évoluées, basées notamment sur des disjonctions d'intervalles temporels, rend le problème très difficile, même lorsque ces disjonctions sont limitées à deux intervalles.

On se consacre ensuite à des problèmes dans lesquels les tâches doivent respecter à la fois des contraintes temporelles et des contraintes de partage et d'affectation de ressources. On présente notamment une synthèse des travaux ayant pour but l'élaboration de règles de propagation faisant interagir des contraintes temporelles simples et des contraintes de ressources. Trois types de raisonnement sont présentés. Le premier nécessite une analyse préalable des conflits entre tâches et infère des conditions de séquençement locales à une ressource. Le deuxième agit suivant le principe d'une contrainte globale. Le troisième intègre simultanément les contraintes de temps et de ressources; il permet de limiter l'intervalle de temps sur lequel peut être réalisée une tâche, de façon à respecter un bilan énergétique.

L'utilisation et le contrôle des mécanismes de propagation de contraintes sont discutés. Ils peuvent servir à découvrir rapidement une inconsistance globale dans la formulation du problème, ou à simplifier sa résolution. Nous fournissons ainsi des éléments sur la décomposition et la stratégie de résolution des problèmes.

¹Très souvent, le problème de décision associé à un problème d'ordonnancement en présence de ressources limitées appartient à la classe des problèmes NP-complets.

2 Les problèmes d'ordonnancement

2.1 Définitions liminaires

La résolution d'un problème d'ordonnancement consiste à placer dans le temps des activités ou tâches, compte tenu de contraintes temporelles (délais, contraintes d'enchaînement, ...) et de contraintes portant sur l'utilisation et la disponibilité des ressources requises par les tâches [EL99, LR01].

★

Un ordonnancement constitue une solution au problème d'ordonnancement. Il décrit l'exécution des tâches et l'allocation des ressources au cours du temps, et vise à satisfaire un ou plusieurs objectifs. Plus précisément, on parle de problème d'ordonnancement lorsqu'on doit déterminer les dates de début et les dates de fin des tâches, alors qu'on réserve le terme de problème de séquençement au cas où l'on cherche seulement à fixer un ordre relatif entre les tâches qui peuvent être en conflit pour l'utilisation des ressources. Un ordonnancement induit nécessairement un ensemble unique de relations de séquençement. En revanche, la solution purement séquentielle d'un problème d'ordonnancement recouvre une famille d'ordonnements (éventuellement une infinité si le domaine de variation des dates est non borné ou réel).

2.2 Notations de base

Les problèmes d'ordonnancement que l'on considère dans la suite ne concernent dans un premier temps que ceux manipulant des ressources disjonctives (ordonnancement d'atelier par exemple) et pour lesquels le problème d'affectation est résolu. Nous reviendrons dans un deuxième temps, d'une part sur les problèmes mêlant décisions d'ordonnancement et d'affectation et, d'autre part, sur les problèmes à ressources cumulatives (ordonnancement de projet).

Un ensemble \mathcal{T} de n tâches doit être réalisé à l'aide d'un ensemble \mathcal{M} de m machines. Chaque machine ne peut exécuter qu'une seule tâche à la fois. Sur une machine μ , on doit exécuter les n_μ tâches d'un ensemble T_μ . Chaque tâche i est caractérisée par (voir figure 1) sa date de début st_i , sa durée p_i , sa date de fin ft_i et doit être exécutée dans une fenêtre temporelle $[r_i, d_i]$ (r_i est la date de début au plus tôt et d_i la date de fin au plus tard). On a les relations : $st_i \geq r_i$ et $st_i + p_i \leq d_i$. La préemption est interdite : une fois commencée, une tâche ne peut être interrompue. Les contraintes prises en compte sont des contraintes temporelles (dates limites, précédences entre tâches, délais d'exécution) ainsi que des contraintes de partage de ressources. Une relation de précedence entre i et j est notée $i \prec j$.

Une particularité importante des problèmes considérés est que la durée des tâches n'est pas fixée mais bornée : $p_i \in [\underline{p}_i, \bar{p}_i]$. Cela exprime notamment des incertitudes sur la connaissance exacte de la durée d'exécution des tâches qui ne sera réellement connue qu'à l'exécution effective (la durée est ainsi qualifiée de *contingente*). Ce caractère incertain

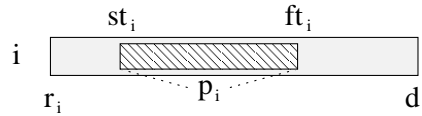


FIG. 1 – Notations de base

de la durée justifie l'utilisation de deux variables temporelles st_i et ft_i pour représenter l'exécution de i .

2.3 Résolution

Lorsqu'on aborde la résolution d'un problème d'ordonnancement, on peut choisir entre deux grands types de stratégies, visant respectivement à l'optimalité des solutions par rapport à un ou plusieurs critères, ou plus simplement à leur admissibilité vis-à-vis des contraintes.

Dans une démarche basée sur un modèle d'optimisation, l'ensemble des connaissances permettant de fournir une solution idéale est intégré dans un modèle programmable dont l'exécution ne demande aucune intervention — ou presque — d'un décideur. De plus, cela suppose que les solutions candidates à un problème puissent être ordonnées de manière rationnelle selon un ou plusieurs critères d'évaluation numériques permettant d'apprécier la qualité des solutions.

Lorsqu'au contraire, certaines connaissances sont difficiles à représenter ou à exploiter (données incertaines, imprécises, critères contradictoires, qualitatifs ou fortement dépendants du contexte), et/ou lorsqu'il est difficile de traduire l'ensemble des objectifs de résolution par un ou plusieurs critères numériques, une approche par satisfaction de contraintes ou basée sur l'aide à la décision peut s'avérer plus réaliste et plus souple. Les techniques de caractérisation des solutions prennent alors tout leur intérêt, notamment celles basées sur la propagation de contraintes qui permettent de restreindre l'espace de recherche des solutions.

La notion d'exactitude est relative : une méthode utilisant un critère d'optimisation est exacte si elle garantit l'optimalité des solutions trouvées ; sinon elle est dite approchée, ou heuristique, lorsqu'on observe empiriquement qu'elle fournit de « bonnes » solutions. Si l'objectif se réduit à l'admissibilité des solutions, une méthode par caractérisation sera dite exacte — on préférera dans ce cas le terme *complète* — si elle fournit toutes les solutions satisfaisant toutes les contraintes du problème, y compris celles que constituent les décisions fournies interactivement par un décideur. Une méthode interactive basée sur la propagation de contraintes peut être considérée comme approchée si ses déductions sont incomplètes.

On désigne par « approches par contraintes » des problèmes d'ordonnancement les travaux centrés autour de la représentation et de l'exploitation des contraintes en ordonnancement. Ces approches confrontent et combinent des méthodes de Recherche Opérationnelle (théorie des graphes, programmation mathématique, méthodes d'optimisation

combinatoire), avec des modèles de représentation et de traitement des contraintes issus de l'Intelligence Artificielle (problèmes de satisfaction de contraintes, algorithmes de propagation de contraintes, langages de programmation par contraintes). L'objectif est de mettre au point des outils facilitant l'interaction entre les modèles et les décideurs, en intégrant des méthodes d'analyse utiles dans un contexte d'aide à la décision (renforcement de la consistance, caractérisation de l'espace des solutions) et des algorithmes efficaces de résolution (génération exhaustive, optimisation).

Les approches par contraintes ont montré leur utilité dans la représentation et la résolution de problèmes d'ordonnancement NP-difficiles tels que les problèmes de Job Shop [ERV76, ERV80, BBD⁺89, DSH90, SC93]. Sans en épouser nécessairement le formalisme, ces approches utilisent des techniques générales des problèmes de satisfaction de contraintes, notamment la propagation de contraintes, mais aussi des stratégies de résolution tirant parti de la spécificité des problèmes d'ordonnancement, portant sur le choix des variables à instancier, le choix de valeurs pour une variable donnée, les techniques de retour arrière, etc. Ces derniers points conditionnent fortement les performances globales de l'algorithme de résolution.

2.4 Modélisation adoptée

2.4.1 Problèmes de satisfaction de contraintes temporelles

Le choix d'une approche par contraintes pour la résolution d'un problème d'ordonnancement rend naturelle l'utilisation des *problèmes de satisfaction de contraintes temporelles* (TCSP) *numériques* pour sa modélisation [DMP91, Hen89, Hen94, Nui94, EL95, ELFS95, Sch98]. Il faut en effet déterminer une affectation des variables représentant la date de début ou de fin de chaque tâche (les deux si la durée des tâches n'est pas connue a priori) à l'intérieur d'un domaine lié à la fenêtre temporelle de la tâche, de telle façon que l'ensemble des contraintes exprimées par des relations numériques soient satisfaites. Par exemple, dans [ELH01], le TCSP numérique associé au problème d'ordonnancement considéré comprend :

- un ensemble de variables temporelles $\mathcal{X} = \{x_1, \dots, x_{2n}\}$ représentant le début et la fin des n tâches ($x_i = st_i$ ou ft_i);
- un ensemble de contraintes unaires $\mathcal{D} = \{D_1, \dots, D_{2n}\}$ représentant les domaines associés à chaque variable. Chaque domaine D_i représente l'ensemble des valeurs pouvant être prises par x_i et est l'union d'un ensemble de n_i intervalles disjoints et ordonnés : $D_i = \{I_i^1, I_i^2, \dots, I_i^{n_i}\} = \{[a_i^1, b_i^1], [a_i^2, b_i^2], \dots, [a_i^{n_i}, b_i^{n_i}]\}$. Ceci impose la disjonction suivante entre doubles inégalités : $(a_i^1 \leq x_i \leq b_i^1) \vee (a_i^2 \leq x_i \leq b_i^2) \vee \dots \vee (a_i^{n_i} \leq x_i \leq b_i^{n_i})$;
- un ensemble de contraintes binaires $\mathcal{C} = \{C_{ij}\}$ limitant les valeurs possibles d'une distance entre deux variables, $x_j - x_i$, par un domaine D_{ij} de n_{ij} intervalles disjoints et ordonnés : $D_{ij} = \{I_{ij}^1, I_{ij}^2, \dots, I_{ij}^{n_{ij}}\}$.

La contrainte universelle correspond à l'intervalle $] -\infty, +\infty[$ qui couvre toutes les valeurs possibles. Les contraintes de domaine peuvent être mises sous la forme de contraintes binaires en introduisant une variable x_0 représentant l'origine. Une incohérence apparaît dans le problème lorsque l'ensemble des intervalles associé au domaine d'une variable ou d'une contrainte est vide.

La précedence entre deux tâches $i \prec j$ est décrite par $st_j - st_i \in \{[p_i, +\infty[\}$. Les contraintes de partage d'une ressource sont modélisées par des contraintes binaires dont le domaine comporte deux intervalles ; par exemple, pour une paire de tâches (l, m) , on a : $st_m - st_l \in \{] -\infty, -p_m], [p_l, +\infty[\}$.

Dès lors qu'on manipule des contraintes plus compliquées (contraintes de ressources cumulatives, durées des tâches dépendant de la ressource sur laquelle elles sont réalisées, durées appartenant à un intervalle de valeurs, temps de transition dépendants de la séquence, ...), l'expression des contraintes dans le formalisme des TCSP implique des contraintes d'arité 4 (i.e., contraintes impliquant quatre variables). Considérons par exemple deux tâches i et j à effectuer sur une machine et telles qu'il est nécessaire de respecter un temps de transition α entre la fin de i et le début de j , et un temps de transition β entre la fin de j et le début de i . On a : $(st_j - ft_i \in [\alpha, +\infty]) \vee (st_i - ft_j \in [\beta, +\infty])$. Le fait de ne plus être en présence de contraintes binaires bannit l'utilisation future des outils d'analyse disponibles pour les TCSP. Nous nous intéressons alors au modèle des inégalités de potentiels introduit dans les années 60 (du XX^e siècle...) par Roy et Sussman [RS64].

2.4.2 Inégalités de potentiels

Une *inégalité de potentiels* représente une contrainte de distance entre deux variables : $x_j - x_i \geq \delta_{ij}$, où δ_{ij} est la distance minimale entre les variables x_i et x_j . Un problème d'ordonnancement peut être modélisé par :

- un ensemble *conjonctif* (ou une conjonction) d'inégalités de potentiels : $(x_j - x_i \geq \delta_{ij}) \wedge (x_l - x_k \geq \delta_{kl}) \wedge \dots$. L'ensemble est satisfait si toutes les inégalités sont vérifiées ;
- un ensemble *non-conjonctif* (ou une disjonction) d'inégalités de potentiels : $(x_j - x_i \geq \delta_{ij}) \vee (x_l - x_k \geq \delta_{kl}) \vee \dots$. L'ensemble est satisfait si au moins une des inégalités est vérifiée.

Notons qu'en considérant la variable origine x_0 , une contrainte de domaine est soit dans l'ensemble conjonctif si elle est sous la forme $x_i \in [a, b]$ ou dans l'ensemble non-conjonctif si elle est sous la forme $x_i \in \{[a_1, b_1], [a_2, b_2], \dots\}$.

Quelques exemples de modélisation pour une paire de tâches (i, j) :

- contrainte de précedence $i \prec j$:

$$st_j - st_i \geq p_i$$
- contrainte de partage d'une machine avec durées connues :

$$(st_j - st_i \geq p_i) \vee (st_i - st_j \geq p_j)$$
- contrainte de partage d'une machine avec durées bornées :

$$(st_j - ft_i \geq 0) \vee (st_i - ft_j \geq 0)$$

Toute contrainte d'un TCSP peut être traduite par une inégalité de potentiels. En utilisant des propriétés de redondance et de tautologie, on montre aisément qu'une contrainte binaire dans le formalisme des TCSP $x_j - x_i \in \{[a_1, b_1], [a_2, b_2], \dots, [a_n, b_n]\}$ implique le respect de l'inégalité de potentiels $(x_j - x_i \geq a_1) \wedge (x_i - x_j \geq -b_n)$ dans laquelle les « trous » entre intervalles disjoints doivent être éliminés : $\{(x_j - x_i \geq a_2) \vee (x_i - x_j \geq -b_1)\} \wedge \{(x_j - x_i \geq a_3) \vee (x_i - x_j \geq -b_2)\} \wedge \dots \wedge \{(x_j - x_i \geq a_n) \vee (x_i - x_j \geq -b_{n-1})\}$ [HL02].

★

En résumé, l'ensemble de contraintes binaires d'un TCSP se réduit donc à une conjonction de disjonctions d'inégalités de potentiels. La réciproque est fautive comme nous l'avons déjà illustré en fin de § 2.4.1.

Une classe particulière des TCSP est celle des *problèmes temporels simples* (STP) dans lesquels toutes les contraintes C_{ij} n'impliquent qu'un seul intervalle de valeurs de la forme $[a(C_{ij}), b(C_{ij})]$. Dans le cas des STP, l'ensemble des contraintes temporelles peut donc se ramener à un ensemble d'inégalités linéaires entre deux variables. On peut noter que le modèle des inégalités de potentiels est parfois redécouvert et présenté comme une extension d'un STP... [SK00].

2.4.3 Représentation graphique

La représentation graphique d'un TCSP (ou d'un STP) est réalisée par un *graphe de contraintes* dans lequel les sommets sont associés aux variables et les arcs aux contraintes entre deux variables.

Associé au formalisme des inégalités de potentiels, on trouve naturellement les graphes potentiels [Roy70, EFR79]. Devant le besoin, dans notre étude, de différencier les dates de début et de fin des tâches, nous avons proposé un nouvel outil, le *graphe potentiels-bornes*, qui, au prix d'un doublement des sommets, offre un pouvoir de représentation bien supérieur aux graphes potentiels-tâches ou potentiels-étapes, utilisés couramment dans le domaine de l'ordonnancement ou de la gestion de projet [EHL95]. Précisons que le graphe potentiels-bornes est proche du modèle intégrant des *relations de précedence générales* (GPR) proposé dans [EK92]; notre modèle se distingue toutefois par le traitement qui lui est appliqué, à savoir que toute contrainte du graphe peut être propagée, en modifiant ainsi les valeurs associées aux nœuds mais aussi celles associées aux arcs².

★

Malgré les critiques apportées aux TCSP pour la représentation de certaines contraintes (fréquentes) en ordonnancement, on ne peut nier le caractère synthétique d'un graphe de contraintes. Le graphe potentiels-bornes présente, lui, une certaine lourdeur dans sa représentation graphique qui ne facilite pas sa lecture. Aussi, d'un point de vue purement graphique, nous conservons la sémantique d'un graphe potentiels-bornes mais en intégrant la représentation d'une contrainte sous forme d'une appartenance à un intervalle. Nous utilisons donc une représentation graphique telle que celle de la figure 2 (tous les arcs n'y

²Cela explique que la propagation sur un graphe avec GPR est faite par un algorithme du type Bellman-Ford alors que celle sur un graphe potentiels-bornes est réalisée par un algorithme du type Floyd-Warshall.

sont pas représentés pour simplifier), sachant que la modélisation interne utilisée par des algorithmes de propagation (cf. § 3) n'utilise que des inégalités de potentiels.

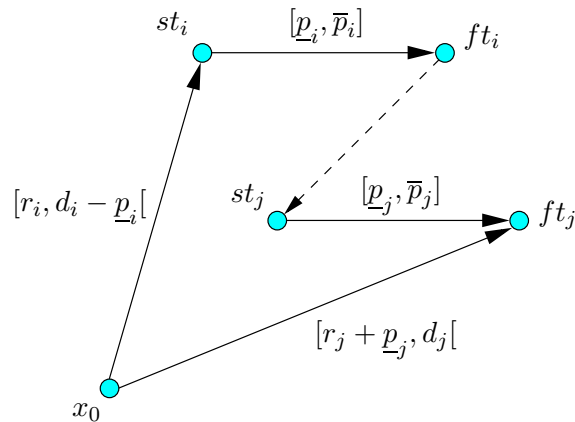


FIG. 2 – Un graphe potentiels-bornes (en pointillés, la contrainte $i < j$)

3 Propagation de contraintes

3.1 Généralités

La propagation de contraintes a pour but de simplifier la résolution d'un problème ou de démontrer l'absence de solution. Elle désigne ainsi, non pas une technique de résolution, mais un ensemble de techniques de réécriture des contraintes qui facilitent l'exploration de l'ensemble des solutions, sans pour autant le modifier. De façon schématique, à chaque étape, le cycle suivant est répété :

1. choix d'un sous-ensemble de contraintes et génération d'une contrainte induite ;
2. mise en évidence d'une contrainte redondante relativement à la contrainte induite ;
3. modification de l'ensemble des contraintes : ajout de la contrainte induite et retrait de la contrainte redondante.

Pratiquement, cela peut consister à retirer du domaine des variables de décision, les valeurs qui n'appartiennent à aucune solution : ce filtrage des domaines évite de nombreuses tentatives de résolution vouées à l'échec. Ces techniques permettent aussi de simplifier l'expression des contraintes, par exemple en éliminant des contraintes redondantes. Dans un contexte d'aide à la décision, cela permet la caractérisation de solutions admissibles en dégageant des degrés de liberté pour la prise de décision effective. La propagation peut enfin dans certains cas apporter la preuve d'une inconsistance globale du problème, et ce avant toute tentative de résolution.

3.2 Propagation de contraintes temporelles

Les contraintes binaires d'un STP (précédences, dates limites, durées fixées) peuvent être représentées sur un graphe potentiels. Le problème minimal³ est alors obtenu de manière polynomiale en appliquant une procédure de recherche de plus longs chemins entre tout couple de sommets, par exemple l'algorithme de Floyd-Warshall de complexité en $O(\mathcal{X}^3)$. Dans le formalisme des TCSP, la propagation correspondante réalise la *consistance de chemins* (PC) [Mon74, Mac77, DMP91]. Celle-ci peut être représentée à partir de la définition de deux opérations entre contraintes : l'*intersection* (notée \oplus) et la *composition* (notée \otimes). Pour toute contrainte C_{ij} , on évalue

$$C_{ij} = C_{ij} \oplus (C_{ik} \otimes C_{kj})$$

qui revient à calculer la quantité

$$C_{ij} = [\max(l(C_{ij}), l(C_{ik}) + l(C_{kj})), \min(u(C_{ij}), u(C_{ik}) + u(C_{kj}))]$$

où les notations $l(\cdot)$ et $u(\cdot)$ représentent respectivement la borne inférieure et la borne supérieure du domaine des contraintes.

Pour des problèmes temporels généraux, PC est sain (les dates supprimées sont bien inconsistantes), mais il est incomplet (il ne garantit pas la détection d'un problème inconsistant). Par ailleurs, son application est limitée à cause d'un phénomène de fragmentation excessive des intervalles associés aux contraintes [SD97].

Sur la base du modèle des inégalités de potentiels (et sa représentation par graphe potentiels-bornes), nous avons proposé une procédure de propagation pour la prise en compte des contraintes non conjonctives [EHL95, HL00]; celles-ci sont exprimées par des disjonctions de contraintes temporelles. La procédure met en jeu l'algorithme de Floyd-Warshall sur la partie conjonctive du graphe, ainsi que des techniques de suppression d'arcs non-conjonctifs. Nous procédons par exemple à la suppression d'inégalités de potentiels avérées inconsistantes par rapport à l'ensemble conjonctif de contraintes, et à la suppression d'un ensemble non-conjonctif qui contient une inégalité de potentiels déjà couverte par l'ensemble conjonctif. La caractérisation n'est évidemment pas complète; elle peut néanmoins être très utile pour la simplification, ou même la résolution, de certains problèmes d'ordonnancement. La complexité dans le pire des cas de la méthode est en $O(\mathcal{X}^3 r q)$, où r est le nombre d'ensemble non conjonctifs et q le cardinal maximum d'une non-conjonction. ★

Notons que lorsqu'on ne s'intéresse qu'à l'actualisation des dates limites des tâches, on peut se contenter d'un algorithme qui réalise la 2-consistance dans une complexité moindre (pour des graphes peu denses) que l'algorithme de Floyd-Warshall qui, lui, arrive à la 3-consistance en $O(|\mathcal{X}|^3)$; on peut ainsi appliquer un algorithme du type Bellman-Ford qui s'exécute en $O(|\mathcal{X}| \cdot |\mathcal{C}|)$ et détermine le domaine minimal des contraintes de la forme C_{0i} et C_{i0} .

³Un problème est *minimal* si les domaines associés aux variables et aux contraintes ne comportent aucune valeur inconsistante.

3.3 Propagation de contraintes de partage de ressources

3.3.1 Introduction

Depuis les travaux séminaux de Jacques Carlier et Jacques Erschler au milieu des années 70 du siècle dernier [Car75, Ers76], les techniques de propagation de contraintes de ressources, notamment celles basées sur l'application de règles d'inférence locales à une machine, ont connu un franc succès. Regroupées communément sous l'appellation *opérations locales*, ces techniques ont fait l'objet de nombreuses recherches et ont amené le développement de nouvelles règles et d'algorithmes efficaces pour les propager. Ainsi, les *sélections immédiates* de Carlier et Pinson permettent de déduire des séquencements obligatoires entre tâches et ensembles de tâches [CP89]. Propagées efficacement et utilisées au sein d'une procédure de recherche par séparations et évaluations conçue pour le problème de job-shop, les sélections immédiates ont permis la résolution dès 1986 de la célèbre instance FT10 restée ouverte pendant plus de vingt ans [FT63].

Plus récemment, les *opérations globales* ont permis d'aller encore plus loin dans la propagation, parfois au prix d'un temps de calcul important, mais offrant des perspectives tout à fait intéressantes comme celles de résoudre optimalement le FT10 en ne développant qu'un nœud de l'arborescence de recherche, ou bien encore de réduire ce nombre de nœuds d'un facteur 10000 pour des instances plus difficiles tout en offrant un gain de temps CPU d'un facteur 4 [CP94, Pér96, Pha00].

Ces excellents résultats ont confirmé le potentiel de l'approche par contraintes des problèmes d'ordonnancement et inspiré un effort de recherche important dans cette voie. De nouvelles règles et algorithmes de propagation ont été proposés (voir par exemple [BJS94, CL94, MS96]). De plus, l'extension de ces techniques à d'autres types de contraintes (préemptives [BL96] ou cumulatives [Lop91, Nui94, CL96b, CN00]) renforce la réutilisabilité de ces techniques et leur applicabilité effective dans le cadre de problèmes réels ; les techniques comme le *raisonnement énergétique* [LEE92] ou l'*ordonnancement pseudo-préemptif de Jackson* (JPPS) [CP01] en sont des exemples. Des langages de programmation par contraintes comme CLAIRE [CL96a] ou CHIP [DHS⁺88] et des bibliothèques offrant des primitives de haut niveau fortement paramétrables comme ILOG Schedule [LeP94] permettent une modélisation aisée et une phase de prototypage/évaluation accélérée. ★

Cette combinaison d'abstraction et d'efficacité explique le succès académique et maintenant industriel du paradigme contraintes et son application aux problèmes d'ordonnancement (voir par exemple [DPH00, BLN01, Bap02, Bru02, Lab03]).

Le but des paragraphes suivants (§ 3.3.2 à § 3.3.4, pages 11 à 22) est de décrire respectivement le mécanisme des opérations locales, des opérations globales et du raisonnement énergétique, tout en précisant nos contributions propres. Le paragraphe 3.3.5 refait un lien avec les graphes de contraintes en indiquant comment peuvent y être interprétées et étendues de telles règles.

3.3.2 Opérations locales

La prise en considération des seules contraintes de partage des ressources disjonctives définit un problème de séquençement : il faut ordonner totalement les tâches qui nécessitent les mêmes machines. La prise en compte des contraintes de fenêtres d'exécution réduit l'ensemble des solutions du problème de séquençement ; certaines configurations sont interdites, d'autres deviennent obligatoires. On cherche notamment à démontrer qu'une tâche ne peut pas — ou doit obligatoirement — être placée avant — ou après — toutes les tâches d'un ensemble (cf. les ensembles non postérieurs/non antérieurs dans [Ers76], les règles de « edge-finding » dans [AC91], les sélections immédiates dans [CP94, BJK94] mises en œuvre également dans [Nui94], et les règles « not-first/not-last » exposées dans [BL96, TL00a]). Ces déductions ont comme résultat de resserrer, ou *ajuster*, les fenêtres d'exécution des tâches. ★

Afin de ne pas surcharger le document, nous ne présentons les règles que dans un sens de séquençement, les symétriques pouvant être aisément obtenues. De même, nous continuons à considérer que la durée des tâches est bornée (si la durée est parfaitement connue il suffit de remplacer dans ce qui suit \underline{p}_i ou \bar{p}_i par p_i).

Précédence interdite/obligatoire sur paires Le premier type de règle consiste à découvrir une précédence interdite entre deux tâches (figure 3) :

$$d_j - r_i < \underline{p}_i + \underline{p}_j \implies i \not\prec j \iff j \prec i \quad (1)$$

Preuve. $r_i + \underline{p}_i$ est la fin au plus tôt de i et $d_j - \underline{p}_j$ la date de début au plus tard de j . Ainsi, $i \prec j$ implique que $r_i + \underline{p}_i < d_j - \underline{p}_j$. En renversant cette proposition, on obtient la règle (1). □

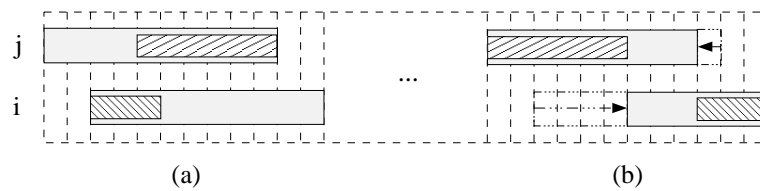


FIG. 3 – Détection d'une relation de précédence simple (a) et ajustements associés (b)

Le déclenchement de la règle (1) provoque les ajustements suivants :

$$\begin{aligned} r_i &\leftarrow \max(r_i, r_j + \underline{p}_j) \\ d_j &\leftarrow \min(d_j, d_i - \underline{p}_i) \end{aligned} \quad (2)$$

Tâche non première La figure 4 illustre une forme de condition plus générale que la règle (1). Les positions interdites d'une tâche i , membre d'une clique de disjonction⁴

⁴Clique de disjonction : ensemble de tâches en disjonction deux à deux.

$S \cup \{i\}$, sont étudiées. Plus précisément, on cherche à montrer que i ne peut être séquencée avant S (i.e., i tâche non première dans $S \cup \{i\}$, cf. paragraphe IV.6 de [Car75], ou S ensemble non postérieur à i , cf. paragraphe IV.3.4 de [Ers76]).

Soit $S \subset \mathcal{T}$, $i \in \mathcal{T} \setminus S$:

$$\max_{s \in S} d_s - r_i < \sum_{s \in S \cup \{i\}} \underline{p}_s \implies i \not\prec S \iff \exists s \in S \text{ tq } s \prec i \quad (3)$$

Preuve. Supposons $\max_{s \in S} d_s - r_i < \sum_{s \in S \cup \{i\}} \underline{p}_s$ et i précède toutes les tâches de S . Soit s_l la dernière tâche séquencée de S ; cela entraîne $st_{s_l} + p_{s_l} \geq r_i + \sum_{s \in S \cup \{i\}} \underline{p}_s$ et suivant (3) on a $r_i + \sum_{s \in S \cup \{i\}} \underline{p}_s > \max_{s \in S} d_s$. Globalement, on arrive à une contradiction. \square

La conclusion de la règle (3) entraîne l'ajustement suivant [LLP99, TL00a] :

★

$$r_i \leftarrow \max[r_i, \min_{s \in S} (r_s + \underline{p}_s)] \quad (4)$$

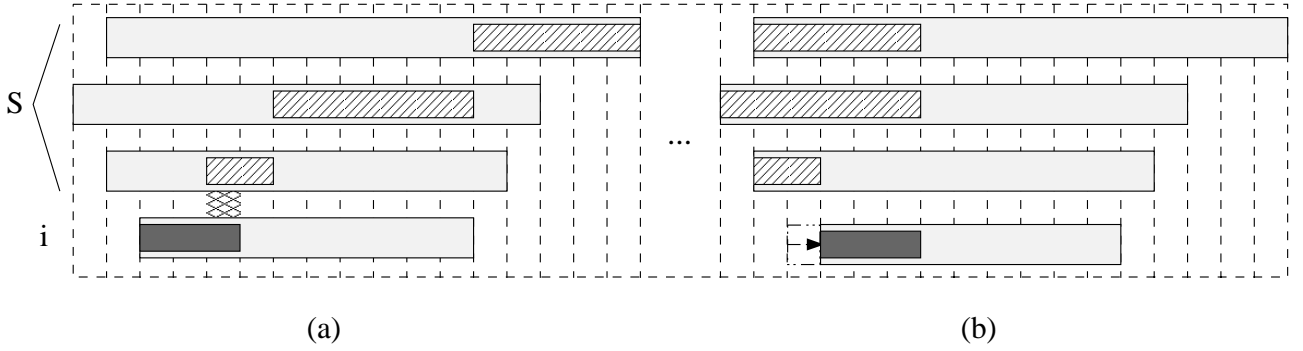


FIG. 4 – Une condition non première (a) et l'ajustement associé (b)

Nous détaillons en annexe E des développements supplémentaires en ce qui concerne la propagation des conditions non-premières/non-dernières. On y présente notamment l'algorithme que nous avons proposé dans [TL00a].

★

Précédence interdite généralisée Considérons deux sous-ensembles S et $\{i, j\}$. La règle suivante, présentée sous une forme similaire dans [DHP99], offre une généralisation de (1) (on retrouve bien (1) en posant $S = \emptyset$ dans (5)) et permet d'appliquer les ajustements (2) :

$$\max_{s \in S \cup \{j\}} d_s - \min_{s \in S \cup \{i\}} r_s < \sum_{s \in S \cup \{i, j\}} \underline{p}_s \implies i \not\prec j \iff j \prec i \quad (5)$$

Preuve. Pour être admissible, toute séquence composée des tâches de $S \cup \{i, j\}$ satisfaisant la relation $i \prec j$, doit se terminer au plus tard à la date $\max_{s \in S \cup \{j\}} d_s$; de même, ce type de séquence commence au plus tôt à la date $\min_{s \in S \cup \{i\}} r_s$. Ainsi, la durée de réalisation de toute séquence satisfaisant $i \prec j$ est nécessairement inférieure ou égale à $\max_{s \in S \cup \{j\}} d_s - \min_{s \in S \cup \{i\}} r_s$. \square

Tâche non insérable – Tâche obligatoirement dernière La tâche i n'est pas insérable dans S , condition notée $i \nmid S$, s'il n'existe aucun ordonnancement admettant une séquence de la forme $\alpha \prec S' \cup \{i\} \prec \beta$, $\forall \alpha, \beta \in S$, $S' = S \setminus \{\alpha, \beta\}$. De tels ordonnancements s'inscrivent nécessairement dans l'intervalle $[\min_{s \in S} r_s, \max_{s \in S} d_s]$; il en découle :

$$\max_{s \in S} d_s - \min_{s \in S} r_s < \sum_{s \in S \cup \{i\}} \underline{p}_s \implies i \nmid S \quad (6)$$

La tâche i doit donc être exécutée soit avant, soit après toutes les tâches de S (i.e., $i \prec S \vee S \prec i$). Si, de plus, la règle (3) se déclenche, la conclusion est très forte : i est obligatoirement en dernière position par rapport aux tâches de S . On parvient ainsi aux *sélections immédiates* [CP90, CP94]. L'ensemble S est dit *ascendant* pour i (figure 5).

Soit $S \subset \mathcal{T}$, $i \in \mathcal{T} \setminus S$:

$$\max_{s \in S} d_s - \min_{s \in S \cup \{i\}} r_s < \sum_{s \in S \cup \{i\}} \underline{p}_s \implies S \prec i \quad (7)$$

Preuve. Cf. [CP90].

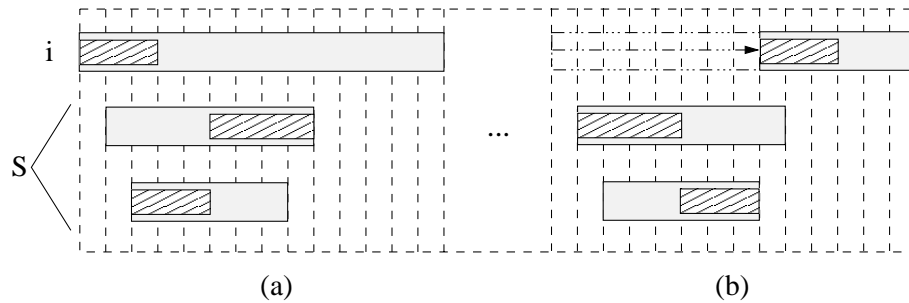


FIG. 5 – Un ensemble ascendant S pour i (a) et l'ajustement associé (b)

A cet ensemble ascendant est associé l'ajustement :

$$r_i \leftarrow \max\left[r_i, \max_{S' \subseteq S} \left(\min_{s \in S'} r_s + \sum_{s \in S'} \underline{p}_s\right)\right] \quad (8)$$

Règles EFF et LSL Dans le cas particulier de la figure 6, il est évident que i doit être séquencée après toutes les tâches de $S = \{j, k\}$. La règle (7) ne détecte pas d'ensemble ascendant pour i . De plus, bien que les séquencements $j \prec i$ et $k \prec i$ soient déduits par la règle des paires disjonctives (1), l'ajustement (8) lié à la précédence $S \prec i$ n'est pas inféré.

Les conditions, appelées LSL pour « Latest Starting time of Last » dans [CL94], permettent de séquencer avant une tâche i toutes les tâches en conflit de ressource dont la date de début au plus tard est strictement inférieure à la date de fin au plus tôt de i .

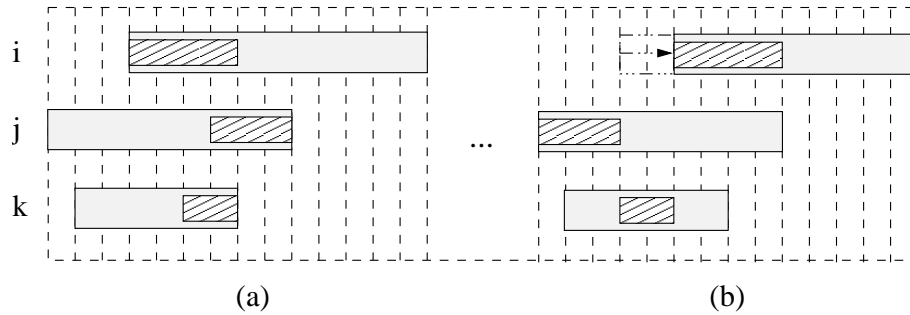


FIG. 6 – Séquencement $\{j, k\} \prec i$ détecté par la règle LSL (a) et ajustement associé (b)

Soit $S \subset \mathcal{T}$, $i \in \mathcal{T} \setminus S$:

$$\max_{s \in S' \subseteq S} (d_s - \underline{p}_s) < r_i + \underline{p}_i \implies S' \prec i \quad (9)$$

La condition de séquencement obtenue définit S' comme un ensemble ascendant pour i et permet donc d'ajuster r_i selon (8).

Dominances entre règles

- Il n'y a pas de relation de dominance entre les paires disjonctives et les sélections immédiates. Pour illustrer cela, on peut constater que la règle (7) n'est jamais déclenchée sur la figure 3. Inversement, l'ajustement de la figure 5 n'est pas déduit par la règle (1).
- Les règles EFF (« Earliest Finishing time of First ») et LSL (9) recouvrent les ajustements obtenus par le raisonnement sur les paires disjonctives (règle (1)).

Preuve. Si l'on considère S' réduit à un singleton $\{j\}$, la règle (9) se réduit à la règle (1). \square

Comme constaté sur la figure 6, la réciproque est fausse.

- La figure 6 montre que la règle (7) ne domine pas la règle (9).
- La figure 5 montre que la règle (9) ne domine pas la règle (7).

3.3.3 Opérations globales

Nous proposons ici de donner une réflexion sur un mécanisme de propagation appelé *opérations globales* ou techniques de « *shaving* » [CP94, MS96, Pér96]. Nous débutons cette réflexion en établissant un parallèle avec certaines techniques de renforcement de la consistance dans les réseaux de contraintes, basées sur un principe de réfutation. Nous proposons ensuite des extensions pour le traitement de problèmes de job shop.

Le principe général des techniques de renforcement de la consistance basées sur un mécanisme de réfutation est le suivant. Une variable est instanciée ; le résultat est propagé ou utilisé dans un arbre de recherche ; si une inconsistance apparaît, la valeur impliquée

dans l'instanciation est retirée du domaine de la variable. Ce principe fait directement référence à une classe particulière de techniques de renforcement de la consistance basée sur la réfutation d'une décision par simple propagation (algorithme SAC, singleton consistance d'arc [DB97]) ou durant une recherche arborescente (algorithme MAC, maintien de la consistance d'arc [SF94]).

Le squelette d'un algorithme de type SAC suit l'algorithme 1 (« $AC(\mathcal{P})$ » désigne la propagation par consistance d'arc sur un CSP \mathcal{P}) :

Algorithm 1 SAC

```

1:  $\mathcal{P} \leftarrow AC(\mathcal{P})$ 
2: pour  $x \in \mathcal{X}$  et  $v \in D_x$  faire
3:    $AC(\mathcal{X}, \mathcal{D}, \mathcal{C} \cup \{x \leftarrow v\})$ 
4:   si inconsistance alors
5:      $\mathcal{C} \leftarrow \mathcal{C} \cup \{\neg(x \leftarrow v)\}$ 
6:    $AC(\mathcal{P})$ 
7:   fin si
8: fin pour

```

Pour une certaine classe de CSP générés aléatoirement (par exemple des réseaux de contraintes denses), SAC offre un bon ratio (temps CPU) / (réduction de domaine), notamment par rapport à la consistance de chemin forte inapte pour le traitement de réseaux de contraintes de grande taille. D'autre part, la maintenance de la consistance d'arc durant la recherche (MAC) permet d'obtenir de meilleures performances que l'algorithme classique de Forward Checking⁵ sur certains problèmes difficiles.

Les techniques de *shaving*⁶ [MS96] ou *opérations globales* [CP94, Pér96] sont basées sur la nature spécifique des contraintes disjonctives : une contrainte locale est posée et les ajustements correspondants sont propagés sur le problème global. La détection d'une inconsistance entraîne la pose de la contrainte inverse.

En considérant qu'un problème d'ordonnancement disjonctif peut être formulé comme un CSP n-aire, nous avons facilement pu faire un parallèle entre SAC et un algorithme de shaving [LT02]. Pour cela, notons :

- $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$ un problème de satisfaction de contraintes n-aire ;
- R une base de règles liées à des opérations locales et la propagation des déductions sur le graphe disjonctif ;
- $R(\mathcal{P})$ le point fixe unique induit par l'application de R sur \mathcal{P} .

La convergence de la propagation se fait vers un état stable unique ; cela sous-entend :

- la monotonie de la propagation R :

$$\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C}) \rightarrow \mathcal{P}' = (\mathcal{X}, \mathcal{D}', \mathcal{C}') \text{ avec } \mathcal{D}' \subseteq \mathcal{D}, \mathcal{C}' \supseteq \mathcal{C}$$

- la complétude des raisonnements.

⁵Un algorithme de *Forward Checking* consiste à propager la valeur de l'instanciation venant d'être effectuée sur les variables non encore instanciées et reliées à la variable courante par une contrainte.

⁶Ce terme fait référence à des régions « rasées » de l'espace admissible.

Le squelette d'un algorithme de shaving est ainsi donné par l'algorithme 2.

Algorithm 2 Shaving

```

1:  $\mathcal{P} \leftarrow R(\mathcal{P})$ 
2: soit  $c$  une contrainte arbitraire tq  $c \notin \mathcal{C}$ 
3:  $R(\mathcal{X}, \mathcal{D}, \mathcal{C} \cup \{c\})$ 
4: si inconsistance alors
5:    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\neg c\}$ 
6:    $R(\mathcal{P})$ 
7: fin si

```

Réduction des fenêtres temporelles On contraint de débiter une tâche dans une fenêtre temporelle réduite, par exemple $st_i \in [r_i, \alpha]$, $\alpha < d_i - p_i$; on effectue la propagation; si une inconsistance est détectée, la tâche devra impérativement débiter dans la sous-fenêtre $[\alpha + 1, d_i - p_i]$ [CP94]. Ce mécanisme a été efficacement implémenté dans [MS96] (on opère à une bissection de la fenêtre initiale); un « double » shaving, itérant le principe de base de Carlier-Pinson sur un niveau de récursion, a permis de résoudre l'instance ABZ7, job shop 20×15 , au bout tout de même de deux jours de calcul...

Ce principe a encore été approfondi dans [Pér96] où des conditions de dominance et un ordre topologique du graphe potentiels sont proposés pour restreindre la recherche dichotomique. Il est aussi proposé d'exploiter les déductions obtenues par propagation sur les autres tâches que celle sur laquelle le shaving est étudié, déductions ayant la particularité d'être toujours vérifiées quelle que soit la restriction faite sur l'intervalle d'étude. Enfin, un raisonnement basé sur un partitionnement complexe des fenêtres temporelles est introduit et permet à l'auteur d'obtenir de remarquables résultats, comme prouver l'optimalité du FT10 sans le moindre backtrack.

Enfin, toujours basé sur ce principe, des résultats assez spectaculaires ont été obtenus dans [Pha00]. Par exemple, sur les instances FT10, ABZ5-6, LA16-20, ORB01-10, le shaving entraîne une réduction du nombre de nœuds par un facteur 100 à 200 au prix d'une augmentation du temps CPU jusqu'à un facteur 6. Pour des instances plus difficiles comme FT20, LA21-30, LA36-40, le shaving amène des gains en temps jusqu'à un facteur 50 en résolution; en preuve d'optimalité, il permet un gain de temps CPU jusqu'à un facteur 4 en diminuant le nombre de nœuds jusqu'à un facteur 10000... On assiste également à une amélioration des meilleures bornes inférieures connues sur de très grandes instances (SWV, YN). Pour des problèmes d'open shop, toutes les instances 10×10 de Taillard et toutes les instances de Hurink [BHJW97], sauf une, sont résolues.

Arbitrage de disjonctions Cette technique, introduite dans [CP94], consiste à caractériser des sélections immédiates. Pour chaque paire non séquencée (i, j) , une séquence est posée, e.g., $i \prec j$, elle est propagée, et si une inconsistance apparaît, il devient nécessaire de respecter la séquence $i \succ j$. Il a été prouvé dans [Pér96] que cette technique est dominée par celle réduisant les fenêtres temporelles.

Propositions d’extensions des techniques de shaving Dans [TL00b], nous avons proposé des extensions des techniques de shaving ; ces extensions sont résumées ci-dessous. L’idée de base est de généraliser le shaving sur des paires à un shaving sur des k -cliques de disjonction avec $k > 2$. ★

Shaving sur triplets La première étape est de considérer des triplets de tâches en conflit. L’idée de dériver des déductions sur une analyse locale des triplets de tâches est apparue dans [Dew92] et reprise dans [Far94]. Six conditions locales sont ainsi testées et si au moins une est vérifiée, une disjonction au moins du triplet est résolue.

Nous avons testé l’efficacité de cette technique utilisée comme shaving. Etant donné une machine sur laquelle n tâches doivent être exécutées, il existe C_n^3 triplets à considérer et à séquencer suivant $3! = 6$ permutations possibles. Chaque permutation est successivement ajoutée en tant que contrainte et propagée sur le problème global. Si (i, j) est une paire d’un triplet, i doit être séquencée avant j si toutes les permutations du triplet où j est séquencée avant i sont prouvées inconsistantes par propagation.

Shaving sur k -cliques ($k > 3$) Il est possible de généraliser la technique précédente. Dans [BJK94], les « *r-set conditions* » sont des opérations locales basées sur la généralisation des sélections immédiates de Carlier-Pinson (cas $r = 2$) à des sous-ensembles de taille > 2 .

Nous avons évalué l’opportunité d’utiliser cette idée comme une technique d’opération globale. Le principe est exactement le même que le shaving sur les triplets. Etant donné une machine sur laquelle n tâches doivent être exécutées, une k -clique est choisie. Les $k!$ permutations possibles sont successivement propagées. Le but est là de déduire des séquences invariantes.

Deux inconvénients surviennent : d’un côté, le coût induit par les $k!$ propagations nécessitées pour chaque k -uplet, d’autre les C_n^k k -uplets à considérer. Le nombre de permutations à considérer pour chaque k -uplet implique une limitation sur la valeur de k . Selon la taille des problèmes considérés, des valeurs de 7 (5040 propagations) à 8 semblent raisonnables, sachant que la propagation complète par un ensemble des règles les plus courantes sur des instances 10×10 de job shops est réalisée en environ un centième de seconde sur un (vieux) Pentium II 350 MHz.

D’un point de vue de l’implémentation, nous cherchons à stopper l’analyse le plus tôt possible, sans nécessairement attendre la fin des $k!$ propagations. Ainsi, chaque permutation est testée successivement avec son inverse ; en effet, si les deux séquences sont prouvées inconsistantes, cela signifie que l’analyse du k -uplet étudié peut être abandonnée. Dans le même ordre d’idée, nous n’appliquons cette technique que sur le k -uplet le plus critique. Nous nous appuyons pour cela sur la notion d’intervalle de tâches critique définie dans [CL94] et présentée dans le § 4.2.

Premiers résultats Nous avons obtenu des premiers résultats sur cette technique raisonnant sur les k -uplets avec $k = 7$ pour calculer des bornes inférieures sur 55 instances de job shop tirées de la littérature. Le nombre d’opérations varie de 100 à 400. Pour résumer

les résultats, on peut dire de manière générale que, d'une part, le shaving sur les triplets domine légèrement le shaving par paires, et, d'autre part, que le shaving sur k -uplets améliore toutes les bornes inférieures (lorsque, bien sûr, celles-ci ne sont pas optimales), sauf deux, obtenues par le shaving déduisant des réductions de fenêtres. La limitation de l'étude à l'intervalle de tâches critique permet d'obtenir davantage d'information que l'étude des C_n^3 triplets de chaque machine pour une augmentation du temps CPU d'un facteur 2 pour les grandes instances à 6 pour les petites.

Perspectives d'amélioration et d'utilisation Les résultats obtenus dans [TL00b] ont montré que le shaving sur les k -cliques était prometteur. De nettes améliorations pourraient toutefois être apportées. Il serait notamment très intéressant de maintenir incrémentalement les ajustements minimaux sur les tâches n'appartenant pas à la k -clique considérée obtenus après propagation (à l'heure actuelle seule l'inconsistance d'une permutation de la k -clique est conservée). Ces ajustements constituent en effet des bornes valides sur ces tâches. ★

Une autre amélioration pourrait être apportée en reprenant le principe du double shaving de Martin et Shmoys [MS96]. L'inconvénient de la récursivité, même sur un seul niveau, est d'augmenter le temps de résolution d'un facteur 100 pour un problème de taille 10×10 , à plusieurs milliers pour des instances plus grandes. La voie pourrait être alors d'étudier des propriétés de dominance qui permettraient de déduire la taille des ensembles de tâches sur lesquels serait effectuée la récursion.

Enfin une autre voie prometteuse concernant ces techniques de shaving serait de les utiliser au sein de procédures de recherche par voisinage. Appliquer des techniques aussi fortes que le shaving pourrait en effet permettre d'éliminer des disjonctions et donc des choix pour des procédures basées sur l'échange (« *swapping* ») de tâches sur un chemin critique, et augmenter ainsi les chances de sortir rapidement d'un optimum local. Cependant, compte tenu du coût engendré par les techniques de shaving, nous pouvons prédire qu'une telle hybridation de méthodes ne pourrait être intéressante que pour de grandes instances, c'est-à-dire pour des tailles à partir de 15×10 .

3.3.4 Raisonnement énergétique

Nous avons proposé un nouveau type de raisonnement lié au concept d'*énergie* ; celui-ci permet d'effectuer des raisonnements quantitatifs intégrant les contraintes de temps et de ressources [ELT91, Lop91, LEE92, LE96]⁷. Comme les opérations locales, ce raisonnement peut produire des conditions de séquençement, sans toutefois nécessiter une analyse du problème en termes d'*ensembles critiques en conflit*⁸. Mais il permet aussi d'interdire la ★

⁷En cela, le raisonnement énergétique est plutôt à rapprocher des opérations locales, mais sa grande généralité nous invite à le présenter à part de ces dernières.

⁸Les ensembles critiques de tâches en conflit sont les plus petits sous-ensembles de tâches qui ne peuvent s'exécuter simultanément compte tenu de la disponibilité insuffisante des ressources (appelés également « *conflict sets* » dans [BEN82] ou « *forbidden sets* » dans [BMR88]). Dans ce document, nous ne détaillons pas les algorithmes de recherche des ensembles critiques en conflit ; ceux-ci sont présentés

localisation d'une tâche sur certains intervalles de temps qui engendrerait un bilan énergétique déficitaire. Ce raisonnement prend tout son intérêt lorsque certaines caractéristiques de réalisation ne sont pas entièrement connues, par exemple lorsque la durée des tâches dépend des ressources utilisées.

Les déductions liées au raisonnement énergétique font appel à des bilans d'utilisation des ressources sur certains intervalles temporels, ce qui entraîne l'identification et le calcul de différentes énergies. Sur un intervalle de temps, l'énergie peut en effet être produite par une ressource ou consommée par une tâche. Dans ce dernier cas, on distinguera sa consommation minimale de sa consommation maximale sur l'intervalle.

Etant donné la généralité du raisonnement énergétique, la présentation des résultats est faite pour des problèmes cumulatifs, le cas disjonctif n'étant présenté, si le besoin s'en fait sentir, que comme un cas particulier. A cet égard, nous introduisons les notations supplémentaires suivantes : A_r est la capacité d'une ressource r (la quantité maximale instantanée disponible de la ressource) ; a_{ir} est l'intensité d'une tâche i (la quantité nécessaire requise à chaque instant pour l'exécution de la tâche). Pour alléger la notation, nous faisons l'hypothèse qu'une tâche i ne requiert qu'une ressource et son intensité sera notée a_i .

Expression de l'énergie produite/consommée L'énergie maximale disponible qu'une ressource μ produit sur un intervalle $\Delta = [t_1, t_2]$ correspond au produit de la longueur de l'intervalle par la capacité (constante) A_μ de la ressource :

$$W_\mu^\Delta = A_\mu \times (t_2 - t_1) \quad (10)$$

Pour une tâche i localisée dans le temps (st_i fixé), l'énergie requise — ou *consommation* — par une tâche i sur un intervalle $\Delta = [t_1, t_2]$, notée w_i^Δ , est donnée par le produit de son intensité par l'intersection de Δ avec l'exécution de i :

$$w_i^\Delta = a_i \times \max[0, \min(ft_i, t_2) - \max(st_i, t_1)]$$

soit, sachant que $ft_i - st_i = p_i$

$$w_i^\Delta = a_i \times \max[0, \min(p_i, t_2 - t_1, ft_i - t_1, t_2 - st_i)] \quad (11)$$

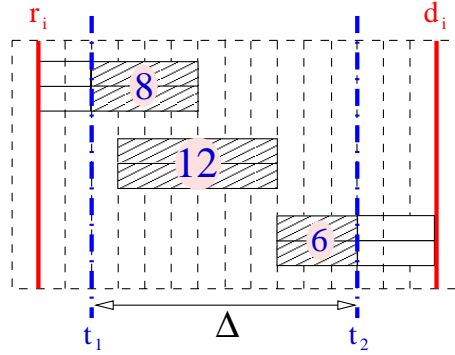
La formule (11) est illustrée par la figure 7 où les zones hachurées représentent la consommation d'une tâche entre t_1 et t_2 pour différentes exécutions.

Dans l'exploitation qui est faite de cette consommation pour la détermination de conditions nécessaires d'admissibilité, nous avons besoin d'exprimer ses bornes inférieure et supérieure, appelées respectivement *consommation obligatoire* et *consommation maximale*.

La consommation obligatoire, notée \underline{w}_i^Δ , est obtenue en considérant les positions de la tâche pour lesquelles son intersection avec Δ est minimale :

$$\underline{w}_i^\Delta = a_i \times \max[0, \min(\underline{p}_i, t_2 - t_1, r_i + \underline{p}_i - t_1, t_2 - d_i + \underline{p}_i)] \quad (12)$$

dans [Esq87, ELH01].

FIG. 7 – Consommation d’une tâche pour trois de ses positionnements ($a_i = 2$)

De manière duale, on obtient la formule de l’énergie maximale, notée \bar{w}_k^Δ :

$$\bar{w}_i^\Delta = a_i \times \max[0, \min(\bar{p}_i, t_2 - t_1, d_i - t_1, t_2 - r_i)] \quad (13)$$

Exemple. Sur la figure 7, la consommation obligatoire de i , qui a ici une durée fixe, est obtenue pour son calage à droite, soit $\underline{w}_i^\Delta = 6$; la consommation maximale est obtenue lorsque i est exécutée totalement entre t_1 et t_2 : $\bar{w}_i^\Delta = 12$.

Règles de déduction Nous formulons à présent les règles basées sur des bilans énergétiques que nous avons établies. Celles-ci permettent des déductions sur l’admissibilité d’un ordonnancement, le séquençement entre tâches ou l’ajustement de fenêtres d’exécution.

Condition suffisante d’inconsistance globale

$$\exists \Delta \text{ tq } W_\mu^\Delta < \sum_{i \in T_\mu} \underline{w}_i^\Delta \implies \text{inconsistance} \quad (14)$$

Preuve. Triviale, compte tenu du caractère minimal de la consommation obligatoire. \square

Ajustements Pour une tâche i et un intervalle Δ donnés, soit $S_i^\Delta = W_k^\Delta - \sum_{j \in T_k \setminus \{i\}} \underline{w}_j^\Delta$ l’énergie maximale disponible pour exécuter i sur Δ compte tenu de la consommation obligatoire des autres tâches de T_k :

$$\exists i, \Delta \text{ tq } S_i^\Delta < \bar{w}_i^\Delta \implies \text{il existe des dates interdites pour } i \quad (15)$$

Les dates conduisant à une consommation plus grande que S_i^Δ sont inconsistantes et doivent être supprimées. Sachant que l’énergie \bar{w}_i^Δ admet $a_i p_i$ comme borne supérieure, que i est non interruptible et que son intensité est constante en cours d’exécution, si $p_i > \frac{S_i^\Delta}{a_i}$, une partie de la tâche, de durée $p_i - \frac{S_i^\Delta}{a_i}$, doit être maintenue à l’extérieur de Δ . Cela entraîne un intervalle de valeurs interdites pour st_i : $[t_1 + \frac{S_i^\Delta}{a_i} - p_i + 1, t_2 - \frac{S_i^\Delta}{a_i} - 1]$ (dans

la figure 8, ces dates interdites sont symbolisées par \times). La suppression de ces valeurs dans le domaine $[r_i, d_i - p_i]$ initialement permis, peut créer un ajustement de r_i et/ou de d_i (avec éventuellement création d'un « trou » dans le domaine de st_i).

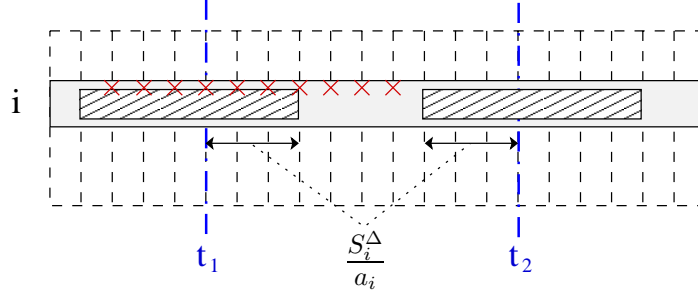


FIG. 8 – Dates interdites pour le début d’une tâche (« trou »)

La procédure d’actualisation pouvant être effectuée en $O(1)$ et étant donné $O(n^2)$ intervalles à considérer pour n tâches à actualiser, l’ensemble des actualisations découlant de l’application de la règle (15) peut être réalisé en $O(n^3)$.

Relations de séquençement Pour une paire de tâches (i, j) et un intervalle Δ donnés, soit $S_{(i,j)}^\Delta = W_k^\Delta - \sum_{l \in T_k \setminus \{i,j\}} \underline{w}_l^\Delta$ l’énergie maximale disponible pour exécuter i et j sur Δ , compte tenu de la consommation obligatoire des autres tâches de T_k . On a la règle suivante :

$$\exists i, j, \Delta \text{ tq } S_{(i,j)}^\Delta < \bar{w}_i^\Delta + \bar{w}_j^\Delta \implies \text{il existe des couples de dates interdites pour } i \text{ et } j \quad (16)$$

En d’autres termes, si i et j consomment toutes deux sur Δ , elles doivent être éloignées d’une distance minimale. On ne peut cependant pas préciser dans quelle position relative se trouvent i et j : cette règle ne permet pas une interprétation directe en termes d’ajustements de fenêtres. Dans ce but, il faut considérer des hypothèses supplémentaires.

Tout d’abord, on peut se limiter aux problèmes disjonctifs et choisir des intervalles Δ tels que les consommations des deux tâches lorsque $i \prec j$ soient maximales, par exemple $\Delta = [r_i, d_j]$; on peut alors conclure à une nouvelle relation de précedence interdite (cf. figure 9) :

$$d_j - r_i < \underline{p}_i + \underline{p}_j + \sum_{l \in T_j \setminus \{i,j\}} \underline{w}_l^\Delta \implies i \not\prec j \iff j \prec i \quad (17)$$

Preuve. Dérivée de celle de la règle (1) associée au caractère minimal de la consommation obligatoire. \square

La règle (17) domine la règle (1) : elle peut se déclencher dans des cas où (1) ne conclurait pas ; en revanche, si (1) se déclenche, (17) aussi.

D’autre part, en choisissant des intervalles de la forme $\Delta = [\min_{J_i} r_j, \max_{J_i \cup \{i\}} d_j]$ avec $J_i = \{j = 1, \dots, n / j \neq i, r_i < r_j < d_i\}$, une autre déclinaison de la règle (16)

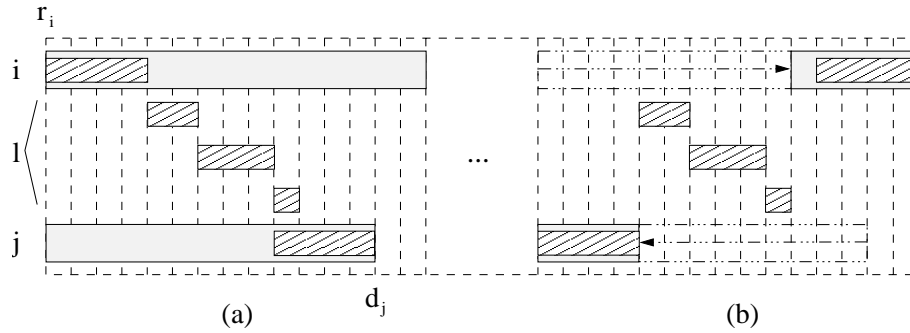


FIG. 9 – Précédence interdite étendue $i \not\prec j$ tenant compte des tâches l (a) et ajustement associé (b)

permet de retrouver l'ajustement produit par la règle (8) liée aux ensembles ascendants (séquencement de i avant toutes les tâches de J_i).

Ainsi, le raisonnement énergétique fournit des règles qui recouvrent des règles performantes et bien connues de propagation (paires de disjonction, sélections immédiates, tâches non premières/non dernières). Il est d'une grande généralité, puisqu'il permet aussi de fournir des règles déduisant des dates de début intermédiaires interdites (dans son expression la plus générale, la règle (15) conduit à la création de « trous » dans le domaine de st_i).

Intervalles pertinents pour l'analyse de l'admissibilité Il est possible d'énumérer en $O(n^2)$ tous les intervalles pertinents pour l'analyse de l'admissibilité. Ceux-ci se résument à trois ensembles de dates ([Lop91] pages 74, 77 et 90, [Bap98] page 68) :

$$\left. \begin{aligned} O_1 &= \{r_i\} \cup \{d_i - p_i\} \cup \{r_i + p_i\} \\ O_2 &= \{d_i\} \cup \{r_i + p_i\} \cup \{d_i - p_i\} \\ O(t) &= \{r_i + d_i - t\} \end{aligned} \right\} i = 1, \dots, n$$

Ces trois ensembles correspondent à des discontinuités de pente des courbes de consommation énergétique lorsqu'on fait varier les bornes des intervalles d'étude [Lop91]. L'analyse de l'admissibilité d'un ordonnancement se fait alors en étudiant les bilans énergétiques sur des intervalles $[t_1, t_2]$ dont les bornes sont dans l'un des trois produits cartésiens : $O_1 \times O_2$, $O_1 \times O(t)$, $t \in O_1$ et $O(t) \times O_2$, $t \in O_2$.

Remarque. Dans la recherche d'une solution d'un problème cumulatif, une étude expérimentale des conditions nécessaires d'admissibilité et des actualisations correspondantes [Bap98] montre que le raisonnement énergétique est trop gourmand en temps de calcul, comparativement aux procédures existantes (par exemple [DH92] ou [BKST98]). On peut ainsi décider de limiter les intervalles d'étude au produit cartésien de $O'_1 = \{r_i\} \cup \{d_i - p_i\}$ par $O'_2 = \{d_i\} \cup \{r_i + p_i\}$, $i = 1, \dots, n$, ce qui conduit en revanche à perdre certaines déductions.

3.3.5 Interprétation sur un graphe de contraintes

Le but de ce paragraphe est d'expliciter comment certaines des règles présentées précédemment peuvent être interprétées et étendues sur un graphe de contraintes. Les résultats présentés ci-dessous et proposés dans [HLV02] ne concernent que des règles utilisées dans un cadre d'ordonnancement disjonctif. ★

L'application de la règle (1) ($d_j - r_i < \underline{p}_i + \underline{p}_j \implies j \prec i$) réalise une propagation de type *consistance d'arc aux bornes* [Lho93, Far94], c'est-à-dire une propagation des contraintes disjonctives sous forme d'ajustements des bornes des fenêtres, sans création de trous dans les domaines des dates de début.

Dans cette règle, la quantité $-(d_j - r_i) + \underline{p}_i + \underline{p}_j$ correspond à la longueur du chemin de st_j à ft_i via ft_j , x_0 et st_i sur le graphe de contraintes conjonctif (figure 10). Si ce chemin produit une inconsistance — i.e., un circuit de longueur positive — avec l'arc $ft_i \rightarrow st_j$ correspondant à la séquence $i \prec j$, une telle séquence n'est pas admissible.

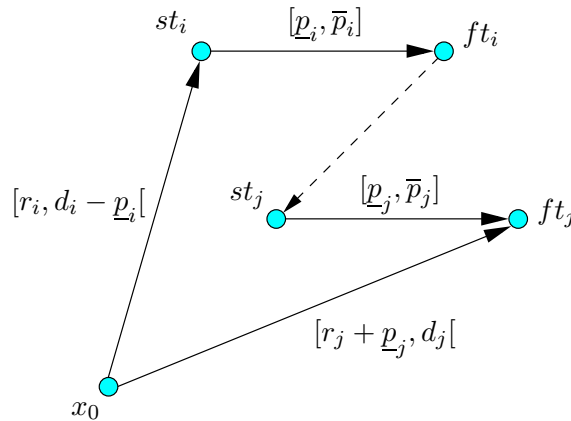


FIG. 10 – Un graphe de contraintes

La règle (1) peut être généralisée en considérant le graphe minimal complet⁹ obtenu à l'issue d'une propagation par un algorithme de type PC sur l'ensemble conjonctif des contraintes temporelles (figure 11). Une borne inférieure de la longueur du chemin de st_j à ft_i est maintenant égale à a' ; en particulier $a' \geq \underline{p}_j - d_j + r_i + \underline{p}_i$. Cela amène la règle suivante qui couvre la règle (1) :

$$a' > 0 \implies i \not\prec j \iff j \prec i \quad (18)$$

Preuve. Dans le graphe minimal complet obtenu après application de PC, tous les chemins de st_j à ft_i ont été parcourus. L'arc $st_j \rightarrow ft_i$ est donc valué par $[a', b']$, où a' est un majorant de la borne inférieure de l'étiquette de l'arc. La condition $a' > 0$ signifie ainsi $i \not\prec j$. \square

⁹minimal \implies toute valeur inconsistante a été détectée
complet \implies il existe un arc entre tout couple de sommet

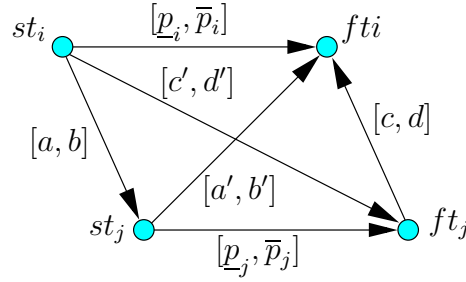
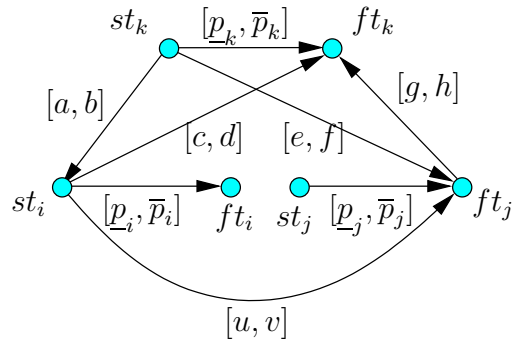


FIG. 11 – Un graphe minimal complet

Appliquons à présent le même raisonnement en considérant la consommation obligatoire d'une tâche k entre st_i et ft_j (figure 12).

FIG. 12 – Consommation obligatoire d'une tâche k dans un graphe de contraintes

L'intervalle de référence est $\Delta = [st_i, ft_j] = [u, v]$. La règle (17) devient alors :

$$v < \underline{p}_i + \underline{p}_j + \sum_{k \neq i, j} \underline{w}_k^{[u, v]} \implies i \not\prec j \iff j \prec i \quad (19)$$

où la formulation générale de la consommation obligatoire de k sur cet intervalle est :

$$\underline{w}_k^{[u, v]} = \max[0, \min(\underline{p}_k, u, c, e)] \quad (20)$$

Preuve. Voir [HLV02].

L'exemple suivant (figure 13) illustre que la règle (19) domine la formulation classique de la règle (17).

La formulation originale de la consommation obligatoire sur l'intervalle $\Delta = [r_i, d_j]$ est $\underline{w}_k^\Delta = \max[0, \min(\underline{p}_k, d_j - r_i, r_k + \underline{p}_k - r_i, d_j - d_k + \underline{p}_k)]$ qui permet d'obtenir $\underline{w}_{k_1}^\Delta = \max[0, \min(1, 9-1, 0+1-1, 9-4+1)] = 0$ et $\underline{w}_{k_2}^\Delta = \max[0, \min(1, 9-1, 2+1-1, 9-9+1)] = 1$. Sur l'intervalle $\Delta = [st_i, ft_j]$, avec la nouvelle formulation $\underline{w}_k^\Delta = \max[0, \min(\underline{p}_k, u, c, e)]$, on obtient $\underline{w}_{k_1}^\Delta = \max[0, \min(1, 4, 1, 4)] = 1$ et $\underline{w}_{k_2}^\Delta = \max[0, \min(1, 4, 2, 0)] = 0$. Supposons que $\underline{p}_i = 4$ et $\underline{p}_j = 3$; avec la proposition (17) le test $8 < 4 + 3 + 0 + 1$ ne permet aucune déduction, tandis que la règle (19) déclenche le test $7 < 4 + 3 + 1 + 0$ qui implique $i \not\prec j$.

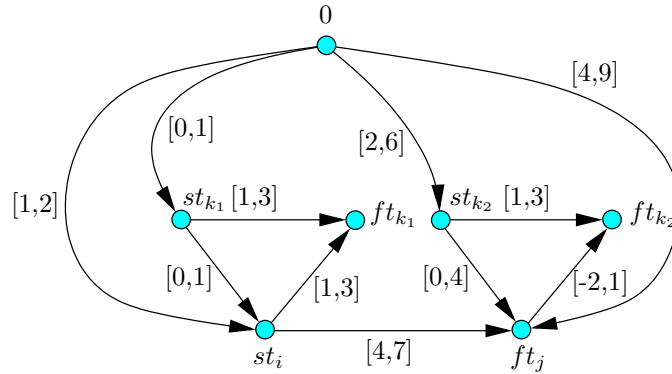


FIG. 13 – Exemple illustratif de la proposition (19)

3.4 Propagation de contraintes d'affectation de ressources

Dans ce paragraphe, les problèmes d'ordonnancement considérés précédemment sont étendus au cas où l'affectation des tâches aux ressources n'est pas fixée a priori. A chaque tâche i est associé un ensemble de ressources $M_i \subseteq \mathcal{M}$ capable de réaliser cette tâche. Pour chaque ressource μ de M_i , la durée de réalisation de la tâche i (qui dépend de la ressource choisie) est notée $p_{i,\mu}$. En conservant l'hypothèse considérée jusqu'à présent, cette durée est incertaine et bornée : $p_{i,\mu} \in [\underline{p}_{i,\mu}, \bar{p}_{i,\mu}]$. Comme pour les problèmes à machines parallèles, on distingue les problèmes en fonction des caractéristiques des ressources : lorsque les ressources sont identiques, la durée de chaque tâche ne varie pas selon la ressource à laquelle elle est affectée (une tâche i a une durée nominale p_i) ; lorsque les ressources sont uniformes, elles possèdent des vitesses différentes, qui ne dépendent pas des tâches exécutées (une tâche i sur une machine μ de vitesse V_μ a une durée $p_{i,\mu} = \frac{p_i}{V_\mu}$) ; enfin, lorsque les ressources sont indépendantes (ou « non reliées »), chaque ressource a une vitesse différente et celle-ci dépend de la tâche exécutée (une tâche i a une durée $p_{i,\mu} = \frac{p_i}{V_{i,\mu}}$). On enrichit la classification suivant le fait que la ressource peut exécuter différents types d'actions (ressource polyvalente) ou que son utilisation est limitée à un type d'action bien défini (ressource spécialisée).

Des algorithmes efficaces existent pour traiter séparément les problèmes d'ordonnancement et d'affectation, mais ils ne garantissent pas de trouver une solution pertinente pour le problème pris dans son ensemble. Dans cette optique de résolution intégrée des problèmes d'ordonnancement et d'affectation, un certain nombre de travaux ont été menés. On peut citer, par exemple, les travaux sur les problèmes d'ateliers flexibles de type « Flow Shop Hybride » [BCNO01] et ceux de type « Job Shop » avec contraintes d'affectation [Bra93, HJT94], intégrant éventuellement des contraintes de transport [Pau95]. Des résultats sur la complexité de tels problèmes sont présentés dans [BJK97]. Des problèmes d'ordonnancement plus généraux incluant des contraintes d'affectation ont également été étudiés, par exemple des problèmes d'ordonnancement multi-ressources dont les ressources peuvent être cumulatives et les gammes non linéaires en considérant ou non des temps de préparation [Nui94, BR96, Kol95, DPRL98, AR99, Kis03], ou encore des problèmes d'ordonnancement de projet multi-modes [SSW94, SD98, Hei03]. Dans la plupart des cas, les

méthodes utilisées sont des méthodes exactes arborescentes ou des méthodes approchées, principalement des métaheuristiques.

Jusqu'à peu, les seuls travaux de notre connaissance effectivement basés sur des techniques de propagation de contraintes pour les problèmes d'ordonnancement et d'affectation étaient ceux de Nuijten d'un côté, et les nôtres de l'autre. Depuis, on peut trouver dans [Hei03], quelques similitudes en ce qui concerne des règles d'élagage de l'arbre de recherche de solutions d'un problème de type ordonnancement de projet sous contraintes de ressources, multi-modes, et intégrant des contraintes temporelles du type « time lags » (problème *MRCPSP/max*). Dans [Nui94], pour chaque tâche et à chaque possibilité d'affectation, on associe une fenêtre temporelle. Les techniques de propagation de contraintes développées s'appuient sur des mécanismes de propagation classiques en ordonnancement et permettent de restreindre les fenêtres temporelles des tâches; cela peut entraîner la suppression de choix d'affectation. Une règle de propagation dédiée aux contraintes d'affectation est également mise en œuvre; elle s'appuie sur une agrégation de ressources auxquelles différentes tâches peuvent être affectées en une ressource cumulative. Par exemple, considérons trois tâches i, j, k , pouvant être affectées soit à la ressource μ_1 , soit à la ressource μ_2 . Dans le cas disjonctif, deux tâches au plus pourront être exécutées simultanément par le groupe de ressources $\{\mu_1, \mu_2\}$. Ce groupe de ressources est alors vu comme une ressource agrégée de capacité 2 et des mécanismes de propagation de contraintes de ressources cumulatives basés sur des règles du type (3) (cf. page 12) sont proposés.

Dans [HL99], les choix d'affectation sont interprétés en termes de contraintes de durées possibles pour les tâches. Cette interprétation ramène le problème d'ordonnancement et d'affectation à un problème d'ordonnancement dans lequel les durées des tâches sont variables. Des mécanismes de propagation de contraintes proches de ceux utilisés dans [Nui94] sont mis en œuvre. De plus, des règles spécifiques basées sur un raisonnement énergétique sont proposées pour la propagation de contraintes d'affectation [HL00].

Dans nos travaux, le traitement effectif des contraintes d'affectation est effectué par deux règles qui tirent parti des règles de propagation des contraintes de partage pour la déduction d'affectations interdites. Le principe de la première règle est décrit par la proposition (21).

Soit P_μ l'ensemble des tâches non affectées qui peuvent être réalisées sur la machine μ .

$$\forall i \in T_\mu, \forall j \in P_\mu, (j \not\prec i) \wedge (i \not\prec j) \implies M_j \leftarrow M_j \setminus \{\mu\} \quad (21)$$

Cela signifie que pour toute tâche $i \in T_\mu$ et $j \in P_\mu$, si $j \not\prec i$ et $i \not\prec j$ sont conjointement déduites en appliquant les règles (5), (18), ou d'autres... alors j ne peut être affectée à μ .

Pour une tâche j qui peut être exécutée sur la machine μ , la seconde règle supprime les affectations qui entraînent des durées inconsistantes vis-à-vis de l'intervalle de durée $[p_j, \bar{p}_j]$ déduit par vérification de la 3-consistance sur l'ensemble conjonctif des contraintes de temps (les bornes inférieure et supérieure de l'intervalle sont déduites en considérant les durées minimum et maximum sur les différentes alternatives d'affectation) :

$$\forall j \in P_\mu, p_{j,\mu} \notin [p_j, \bar{p}_j] \implies M_j \leftarrow M_j \setminus \{\mu\} \quad (22)$$

La suppression d’alternatives de ressources pour des tâches non encore affectées peut permettre d’actualiser leur durée. Cela peut aussi contribuer à résoudre le problème d’affectation, certaines tâches se trouvant nécessairement affectées à la machine qui n’est plus que le seul élément d’un ensemble de machines permises.

Ces déductions d’affectation produisent de nouvelles contraintes de partage qui peuvent déclencher les règles de propagation appropriées. Enfin, une inconsistance globale est détectée s’il n’existe plus aucune possibilité d’affectation pour une tâche.

3.5 Propagation de contraintes et recherche de solutions

L’utilisation de mécanismes de propagation de contraintes au sein d’un algorithme de résolution par tentative et retour arrière permet de restreindre l’espace de recherche de solutions. Un compromis est à réaliser ; des mécanismes de propagation puissants en termes de réduction de l’espace des solutions peuvent simplifier la résolution (celle-ci pouvant même dans certains cas s’effectuer sans aucun retour arrière), mais s’avèrent parfois aussi très coûteux en temps de calcul ; de même, une absence de mécanismes de propagation conduit à des explorations inutiles de l’espace des solutions et entraîne de nombreux retours arrière.

Pour un problème d’ordonnancement, un schéma général d’algorithme de résolution par tentative et retour arrière est le suivant [SF96, Nui94, ELH01] :

★

Algorithm 3 Backtrack

```

1: tant que il reste des tâches à ordonnancer et qu’une inconsistance n’est pas détectée
   faire
2:   appliquer les règles de propagation
3:   si une inconsistance est détectée alors
4:     si il existe des choix non explorés alors
5:       retour arrière sur un choix non exploré
6:     sinon
7:       inconsistance globale détectée
8:     fin si
9:   sinon
10:    sélectionner une tâche
11:    sélectionner un placement pour cette tâche
12:   fin si
13: fin tant que

```

Dans cet algorithme, il est important de concevoir une stratégie pertinente pour l’application des règles de propagation. De même, les heuristiques de sélection des variables et d’affectation de valeur ont un impact important sur les performances des procédures de recherche de solutions [SC93, Sad94, BLN95, SF96]. Ces heuristiques peuvent être statiques ou évoluer dynamiquement (en fonction de résultats de la propagation par exemple). A titre d’exemple, des heuristiques générales de sélection de variables sélectionnent priori-

tairement la variable la plus contrainte (celle pour laquelle il reste le moins de choix de valeurs ou « first failed ») ou la variable la plus contraignante (celle liée au plus grand nombre d'autres variables ou « most constrained »).

Dans le cas des problèmes disjonctifs (job-shop), pour lesquels la résolution du problème est basée sur la recherche d'un séquençement des tâches en disjonction, Smith et Cheng définissent une série d'heuristiques basées sur la flexibilité temporelle associée à différentes décisions de séquençement [SC93]. Ce sont des heuristiques de choix de tâches basées sur la mesure de marges temporelles liées à un séquençement : $slack(i \prec j) = d_j - r_i - (\underline{p}_i + \underline{p}_j)$; cette marge est donc directement reliée à la condition de déclenchement de la règle (1).

La stratégie retenue dans [BLN95, TL00a] consiste à sélectionner la ressource de plus petite marge, où la marge d'une ressource est définie par la différence entre l'offre et la demande d'utilisation de cette ressource sur la fenêtre temporelle de chaque tâche non ordonnée. Les règles de propagation sont alors appliquées sur cette ressource jusqu'à ce qu'il n'y ait plus de déduction, avant de passer à la ressource suivante dans l'ordre décroissant des marges. La sélection des tâches se fait au sein des possiblement premières (ou dernières), identifiées par application des règles du type (3), et ordonnées dans l'ordre croissant des dates de début au plus tôt (voir également § E.1). ★

Enfin, l'optimisation d'un critère peut être réalisée à l'aide d'une procédure de recherche dichotomique exploitant l'algorithme 3. Par exemple, pour la minimisation de la durée totale, la procédure d'optimisation détermine un horizon d'ordonnancement initial H situé au milieu d'une borne inférieure triviale et d'une borne supérieure, déduite par une (méta)heuristique. Cette valeur de l'horizon est prise comme durée totale de l'ordonnancement et contraint les dates de fin des différentes tâches du problème. Si l'algorithme par tentative et retour arrière ne détecte pas d'inconsistance globale du problème sur l'horizon H , c'est-à-dire si une solution est trouvée, la valeur de H est alors prise comme borne supérieure; sinon la borne inférieure devient $H + 1$. La procédure d'optimisation reprend en déterminant une nouvelle valeur de H par dichotomie. L'optimum est atteint lorsque bornes inférieure et supérieure sont égales.

Selon les combinaisons de règles utilisées par le processus de propagation et les algorithmes employés pour les propager, les résultats obtenus peuvent fortement varier. En effet, différents paramètres entrent en jeu lors de la propagation :

- Toutes les règles n'ont pas le même pouvoir de « coupe ». Cela dépend de la force intrinsèque de la règle mais aussi des données de l'instance. Certaines règles très efficaces sur une instance peuvent se révéler trop coûteuses pour une autre. La complémentarité des règles utilisées est aussi un paramètre important dans le choix de la base employée. Les réductions effectuées par les différentes règles peuvent soit être redondantes (ce qui est fortement à éviter), soit être totalement disjointes (ce qui est une des propriétés de la base de règles idéale), soit encore être partiellement complémentaires.
- L'efficacité « pure » des algorithmes est parfois bruitée par le montant de propagation désiré par l'utilisateur. Ce paramètre offre un réglage très utile pour ajuster

le temps accordé à la propagation en fonction du temps disponible ou de l'efficacité présumée de la propagation — par une estimation préalable du *niveau de contraintes* du problème¹⁰ par exemple — et permet d'obtenir un compromis entre le temps de propagation et le montant de réduction obtenu. Par contre, on ne respecte alors plus le paradigme du point fixe qui assure un résultat de propagation identique quel que soit l'ordre de propagation et l'algorithme employé pour chaque règle. Dans notre travail, nous cherchons avant tout à assurer la complétude de la propagation à des fins d'évaluation et de comparaison entre différentes règles et algorithmes. Remarquons toutefois que l'objectif de complétude peut être facilement dégradé en limitant le nombre de passes pour la propagation d'une règle.

Dans [TL00a], notre objectif a été d'évaluer, sur un nombre significatif d'instances de job-shops de la littérature considérées comme difficiles (voir [JM99]), le pouvoir de coupe individuel des règles classiques de la famille des opérations locales, la complémentarité de différentes combinaisons de ces règles (opérations locales et globales) par rapport aux réductions de domaines effectuées et les performances en termes de temps de calcul de différents algorithmes (« edge-finder » de Nuijten, algorithme original de détection de tâches non-premières/non-dernières, algorithmes complets implémentant d'une part les règles « exclusion » et « case is-after » définies dans [CL95] et d'autre part la règle énergétique (17)).

Dans [HL02], nous traitons les problèmes mixtes d'ordonnancement et d'affectation avec machines polyvalentes et non reliées, et un objectif de minimisation de la durée totale de l'ordonnancement (*makespan*). L'approche développée exploite un algorithme par tentative et retour arrière intégrant des règles de propagation de contraintes de partage et d'affectation de ressources, et des heuristiques de choix des variables et des valeurs. Des expériences sur des instances générées aléatoirement montrent l'intérêt d'une approche intégrée (vs. une approche résolvant de manière séquentielle le problème d'affectation et d'ordonnancement) en ce qui concerne l'obtention d'une borne supérieure et le temps de résolution.

4 Structures temporelles pour la propagation

Les règles de propagation de contraintes utilisées manipulent généralement des sous-ensembles de tâches vérifiant des propriétés particulières [CP94, Nui94, BL95, TL00a]. La mise en évidence des sous-ensembles pertinents pour l'analyse se heurte souvent à une forte combinatoire. On cherche donc à élaborer des stratégies de contrôle des règles assurant un raisonnement aussi complet que possible pour un coût de calcul raisonnable. A cet effet, il nous a semblé particulièrement intéressant de débiter l'étude d'un problème par l'examen de sa structuration temporelle, afin de dégager des conditions restrictives sur les séquences admissibles. Cela peut permettre une réduction de la combinatoire du

¹⁰Dans [Bar03], plusieurs classes d'indicateurs sont proposés pour mesurer ce niveau ; ceux-ci sont établis pour des problèmes d'ordonnancement avec ou sans contraintes d'affectation et caractérisent, suivant le cas, la densité d'un ordre de fabrication ou le taux de charge d'une ressource.

problème avant de lancer le processus de propagation de contraintes. Ainsi, cela consiste à mettre en œuvre une « pré-analyse » des problèmes liée à leur structuration temporelle.

Nous proposons deux structures temporelles pour un contrôle efficace de techniques de propagation de contraintes en ordonnancement disjonctif :

1. Les groupes de tâches à rangs inclus présentent la particularité d’agir non pas sur un ensemble de fenêtres temporelles, mais sur un ensemble d’intervalles de rangs admissibles. Sur la base de cette structure, on propose une décomposition temporelle en regroupant des tâches présentant des caractéristiques séquentielles particulières.
2. Un intervalle de tâches représente un ensemble de tâches en conflit de ressource dont l’exécution est bornée par un intervalle temporel. Un treillis est ensuite proposé pour organiser la structuration de tels intervalles de tâches. Nous nous attachons à montrer l’intérêt de ce treillis pour l’application de techniques de renforcement de la consistance en ordonnancement.

4.1 Groupes de tâches à rangs inclus

L’intervalle de rangs admissibles $\mathcal{R}(i)$ d’une tâche i représente l’ensemble des positions non démontrées interdites pour i dans une séquence admissible [LLP96, LLP98]. Il est borné par un rang à gauche $R_g(i)$ et un rang à droite $R_d(i)$: $\mathcal{R}(i) = [R_g(i), R_d(i)]$. Nous avons mis en évidence trois niveaux de regroupement des tâches (figure 14) :

- les *groupes autonomes de tâches* (GAT) qui ont l’avantage d’être indépendants au sens du séquençement mais qui présentent souvent une taille trop importante pour pouvoir facilement ordonnancer les tâches ;
- les *groupes de tâches à intervalles de rangs égaux* (GTRE) permettent un ordonnancement facile de leurs tâches (elles sont totalement permutables), mais sont trop couplés ;
- les *groupes de tâches à intervalles de rangs inclus* (GTRI) dont la caractérisation qui utilise la relation d’inclusion — au sens large — entre intervalles de rangs se rapproche d’une caractérisation en B-pyramides [ELM99]. Elle réalise un bon compromis entre l’indépendance des groupes et la permutabilité des tâches au sein d’un groupe.

Deux tâches i et j distinctes appartiennent au même GTRI I si et seulement si i et j ont des intervalles de rangs $\mathcal{R}(i)$ et $\mathcal{R}(j)$ inclus, ou il existe une tâche $k \in I$ dont l’intervalle de rangs $\mathcal{R}(k)$ inclut strictement celui de i et de j :

$$\mathcal{R}(i) \subseteq \mathcal{R}(j) \text{ ou } \mathcal{R}(j) \subseteq \mathcal{R}(i) \text{ ou } \left\{ \begin{array}{l} \exists k \in I, \quad k \neq i, j, \text{ tq :} \\ \mathcal{R}(i) \subset \mathcal{R}(k) \text{ et } \mathcal{R}(j) \subset \mathcal{R}(k) \end{array} \right.$$

La décomposition en GTRI est utilisée dans les problèmes disjonctifs (problèmes à une machine et flow shops). L’avantage de ce type de structuration est sa simplicité, la complexité de sa mise en œuvre étant celle d’un tri. De plus, si l’on autorise l’appartenance

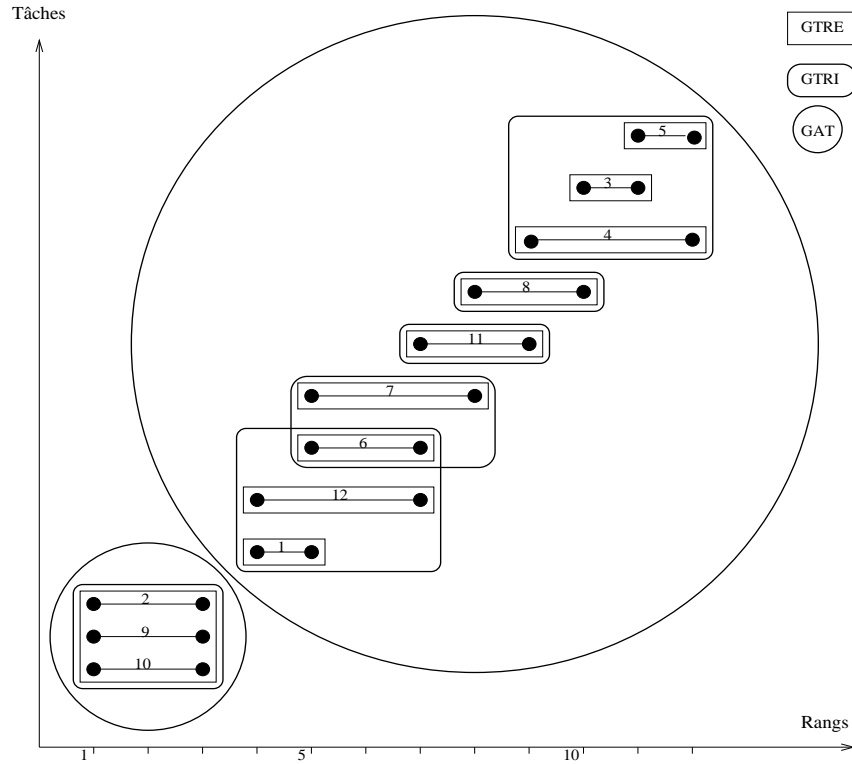


FIG. 14 – Exemple de groupes de tâches

d'une tâche à plusieurs GTRI, la décomposition est unique. En outre, cette décomposition met en évidence des conditions nécessaires d'admissibilité, exprimées à l'aide d'actualisations de dates limites et de rangs, puis fournit une structure décomposée du problème en accord avec les conditions d'admissibilité [LLP99].

★

Cette décomposition peut là encore servir de structure de base pour l'application de techniques de « edge-finding » déjà évoquées. Les précédences entre une tâche et un ensemble de tâches ne sont recherchées que sur les sous-problèmes que constituent les GTRI, de manière à réduire le temps de calcul requis pour caractériser les solutions admissibles. Du point de vue de l'aide à la décision, cela constitue également un avantage puisque des ensembles de tâches à ordonnancer de taille réduite sont considérés en vue de leur traitement.

4.2 Treillis d'intervalles de tâches

Soit une famille d'intervalles $F(\mathcal{T})$ représentant les fenêtres associées à un ensemble de tâches \mathcal{T} . Soit $P \subseteq F(\mathcal{T})$, avec $P = \{x_1, x_2, \dots, x_n\}$. L'intervalle sous-tendu par P , noté \overline{P} , est défini par : $\overline{P} = \overline{x_1, x_2, \dots, x_n}$. On appelle *intervalle de tâches* [CL95] un sous-ensemble maximal P non vide de $F(\mathcal{T})$, tel que :

$$\forall x \in F(\mathcal{T}) \quad x \subseteq \overline{P} \Rightarrow x \in P$$

Un intervalle de tâches P contient toute tâche dont la fenêtre est recouverte par l'intervalle sous-tendu par P . Un intervalle de tâches représente donc un ensemble de tâches en conflit de ressource dont l'exécution est bornée par un intervalle temporel.

La relation d'inclusion ensembliste \subseteq sur l'ensemble des intervalles de tâches de F forme un sous-sup-demi-treillis (tout couple d'intervalles de tâches admet un majorant) [TLE99]. Pour représenter cette relation, on peut donc faire appel au diagramme de Hasse. La figure 16 représente le diagramme de Hasse de la relation d'inclusion ensembliste entre les intervalles de tâches de l'exemple de la figure 15. ★

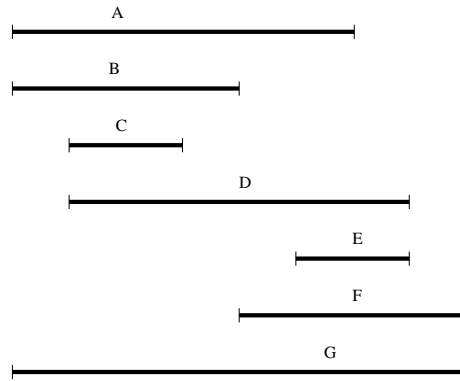


FIG. 15 – Exemple de fenêtres temporelles d'un problème

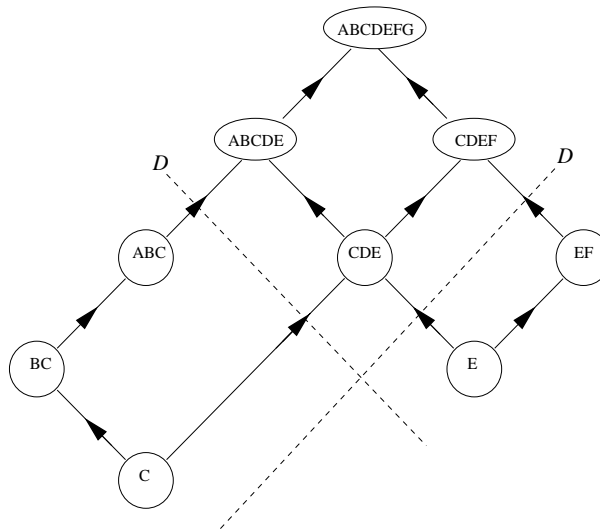


FIG. 16 – Sous-sup-demi-treillis de l'inclusion entre intervalles de tâches

Certaines règles de propagation de contraintes de ressources sont basées sur l'étude du positionnement relatif entre une tâche x et un sous-ensemble de tâches S tels que $x \notin S$ (techniques de « edge-finding ») [CP94, Nui94, BL95, TL00a]. La recherche des sous-ensembles S s'appuie alors sur le sous-sup-demi-treillis entre intervalles de tâches (figure 16). L'énumération des intervalles temporels sous-tendus par les intervalles de tâches limite considérablement la complexité de l'analyse (au maximum $\frac{n(n-1)}{2}$ si toutes ★

les dates de début et de fin diffèrent et s'il n'y a aucune inclusion entre elles). De plus, la structuration en intervalles de tâches tient compte de la structure temporelle du problème, c'est-à-dire que seuls des groupes de tâches pouvant réellement influencer le placement d'une tâche sont générés.

Ainsi, la relation $x \notin S$ implique $r_x < \min_{s \in S} r_s$ ou $d_x > \max_{s \in S} d_s$. Sur un treillis d'intervalles de tâches classé selon un axe de dates début au plus tôt croissantes et un axe de dates de fin au plus tard décroissantes, les deux conditions précédentes divisent le treillis en quatre régions distinctes (figure 17) :

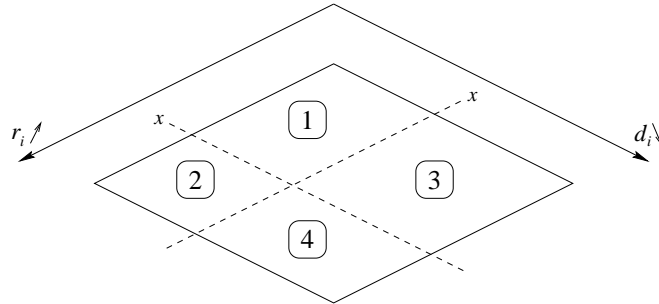


FIG. 17 – Zones dans le treillis d'intervalles de tâches

- la région ① regroupe les intervalles de tâches incluant x . Cette zone n'est donc pas à considérer pour le déclenchement des règles ;
- dans la région ②, la condition $r_x < \min_{s \in S} r_s$ est vérifiée. Si l'on a déduit par ailleurs que x ne pouvait être exécutée au sein de S ($x \not\vdash S$) alors $x \prec S$.
- dans la région ③, la condition $d_x > \max_{s \in S} d_s$ est vérifiée. Si $x \not\vdash S$ alors $S \prec x$.
- dans la région ④, les deux conditions sont vérifiées. Si $x \not\vdash S$ alors $x \prec S$ ou $S \prec x$.

La figure 16 illustre la partition du treillis d'intervalles de tâches pour la tâche D . La recherche de déductions sur D amène à considérer des intervalles de tâches situés dans les zones ②, ③ et ④.

Des sous-treillis d'intervalles de tâches sont définis pour chaque tâche du problème. L'application des règles de edge-finding se fait par une recherche en largeur d'abord. En effet, de par la relation d'inclusion entre les éléments d'un sous-treillis, ce type de stratégie teste en premier les intervalles de tâches de plus grande extension desquels sont attendues les déductions les plus fortes.

5 Orientations de recherche

5.1 Bilan

Notre travail de recherche porte sur l'étude des techniques de propagation de contraintes en ordonnancement. Notre contribution dans ce domaine peut être résumée en considérant les points suivants :

- « Pré-analyse » de problèmes d’ordonnancement basée sur une structuration temporelle, par Groupes de Tâches à intervalles de Rangs Inclus ou Treillis d’Intervalles de Tâches, permettant une décomposition pour l’application des règles de propagation.
- Proposition d’un modèle, le Graphe Potentiels-Bornes, et d’algorithmes pour la propagation de contraintes temporelles non conjonctives.
- Proposition d’un algorithme générique pour la propagation de conditions « non premières/non dernières ».
- Extension des déductions des opérations locales et du raisonnement énergétique par un rapprochement avec les problèmes de satisfaction de contraintes temporelles.
- Extension des techniques d’opérations globales (« *shaving* »).
- Mécanismes de propagation de contraintes et stratégies de recherche pour des problèmes mixtes (ordonnancement *et* affectation).

5.2 Prospective

Compte tenu des résultats obtenus sur les voies précédentes, notre prospective est désormais guidée par les orientations de recherche suivantes.

1. Problèmes mixtes d’ordonnancement et d’affectation

Nous continuerons à développer des règles et algorithmes de propagation de contraintes permettant de prendre en compte de manière intégrée les deux types de décisions, ordonnancement et affectation, en considérant explicitement leur propre sémantique respective.

Afin d’améliorer les performances de la procédure de résolution de tels problèmes, nous pensons notamment intégrer les mécanismes de propagation durant la phase de résolution et envisager de nouvelles stratégies d’exploration arborescente. Nous cherchons pour cela à adapter les techniques connues sous le nom de LDS (« Limited Discrepancy Search »), procédure où la séparation des nœuds est effectuée selon une déviation croissante, mais limitée, par rapport au choix dicté par une heuristique d’instanciation de variables. Un effort important devra être consacré à la définition de stratégies de recherche adéquates pour la sélection des tâches et des ressources.

Nous poursuivrons également l’extension des mécanismes proposés pour la propagation de contraintes de ressources au cas de problèmes mixtes d’ordonnancement et d’affectation avec tâches de durée *contingente* (*i.e.*, connue en cours d’exécution).

Cette dernière voie de recherche a des applications concrètes importantes. Nombreux sont en effet les cas où les événements ne sont pas sous contrôle complet d’un agent d’exécution. On peut citer par exemple, les dates d’arrivée d’équipements sur un chantier ou dans une entreprise (ces temps pouvant être liés à des durées de transport dépendantes de conditions de circulation, climatiques, économiques, ...), à des aléas liées aux ressources humaines dans les activités de services, à des vitesses de réaction, de fermentation, d’incubation (dans l’industrie agro-alimentaire ou pharmaceutique). Dans de tels cas, l’agent d’exécution n’a pas le pouvoir de sélectionner précisément certains paramètres et est totalement tributaire de temps observés en

cours d'exécution.

2. Génération de colonnes

Nous expérimentons actuellement l'adéquation des techniques de programmation linéaire généralisée (ou *génération de colonnes*) pour des cas à forte explosion combinatoire, notamment celui de la programmation des prises de vue de satellites d'observation de la Terre qui présente des contraintes très spécifiques (contraintes stéréoscopiques notamment) (cf. § III.C).

En génération de colonnes, le problème initial est décomposé en un problème maître résolu par la programmation linéaire classique et en un sous-problème dont la méthode de résolution est fortement dépendante du contexte ; pour cette dernière partie, nous envisageons pour l'instant les algorithmes de recherche de chemins dans un graphe, sous contraintes de ressources.

Cette étude sur la génération de colonnes nous encourage vivement à accentuer dans l'avenir notre effort de recherche sur l'hybridation de méthodes pour la résolution de problèmes d'optimisation combinatoire. Dans cette optique et compte tenu de l'expérience acquise sur les approches par contraintes, nous souhaitons privilégier la combinaison d'une approche par contraintes et de la programmation linéaire.

3. Projet LORA

Le projet LORA concerne la réalisation d'une plate-forme logicielle pour l'ordonnancement et l'affectation (cf. § III.C). Le but est de fournir un outil commun de validation et d'expérimentation de méthodes, et une vitrine de démonstration de notre savoir-faire. La spécification générale de la plate-forme est achevée. De plus, le modèle général et original des « *diagrammes de tâches* » autorise la spécification d'alternatives ensemblistes et temporelles, et permet ainsi la représentation d'un spectre large de problèmes d'ordonnancement (atelier avec gammes alternatives, assemblage, ordonnancement de projet...). Nous avons également défini le formalisme avec lequel un utilisateur déclare les ressources relatives à un problème d'ordonnancement ainsi que leurs caractéristiques (intensité et énergie disponibles, graphe d'états, etc.). Enfin, la notion de *mode d'exécution* a été introduite pour permettre de spécifier les diverses façons avec lesquelles une tâche utilise une ou plusieurs ressources.

Pour ce projet, la définition d'un langage général pour la représentation des processus de réalisation, des moyens, des savoir-faire, des objectifs et des solutions (basé sur des schémas XML) doit être validée. La réflexion sur les modalités d'interaction entre les modules constituant la plate-forme, vue dans sa globalité, est à parachever. Il reste également à mener des travaux de développement des fonctionnalités indispensables à rendre la plate-forme opérationnelle. Il s'agit notamment de créer les outils permettant de classifier un problème, de faire appel à un algorithme de résolution, de stocker la(les) solution(s) produite(s) et de la(les) visualiser.

4. Aide à l'Organisation Coopérative

Nous avons déposé un projet d'Action Thématique Incitative sur Programme (ATIP) Jeunes Chercheurs du département STIC du CNRS sur l'« Aide à l'Organisation Co-opérative ». Si ce projet est accepté, nous prolongerons ainsi des travaux antérieurs sur la coopération en ordonnancement et en gestion de la production. L'objectif de la recherche consiste à proposer et valider des protocoles d'interaction, éventuellement configurables, permettant de supporter la coopération, en vue de résoudre des problèmes de gestion d'activités dans les réseaux d'entreprises.

Nous prônons pour cela l'utilisation d'une approche par contraintes permettant d'intégrer des méthodes d'analyse utiles dans un contexte d'aide à la décision (renforcement de la consistance, caractérisation de l'espace des solutions) et des algorithmes efficaces de résolution (éventuellement de façon exacte).

II – Références bibliographiques

- [AC91] D. Applegate and W. Cook. A computational study of the job-shop scheduling problem. *ORSA Journal on Computing*, 3(2):149–156, 1991.
- [AR99] C. Artigues and F. Roubellat. Real time multi-resource shop floor scheduling with setup times. In *Proc. of International Conference on Industrial Engineering and Production Management (IEPM'99)*, volume 2, pages 127–136, Glasgow, Scotland, 1999.
- [Bap98] P. Baptiste. *Une étude théorique et expérimentale de la propagation des contraintes de ressources*. PhD thesis, Université de Technologie de Compiègne, 1998.
- [Bap02] P. Baptiste. Résultats de complexité et programmation par contraintes pour l'ordonnancement. Habilitation à Diriger des Recherches, Université de Technologie de Compiègne, 2002.
- [Bar03] S. Barkati. Module d'ordonnancement agrégé. Rapport de DEA Systèmes Industriels, Ecole Doctorale Systèmes, LAAS-CNRS, 2003.
- [BBD⁺89] G. Bel, E. Bensana, D. Dubois, J. Erschler, and P. Esquirol. A knowledge-based approach to industrial job-shop scheduling. In A. Kusiak, editor, *Knowledge-based systems in manufacturing*, pages 207–246. Taylor & Francis, 1989.
- [BCNO01] J.-C. Billaut, J. Carlier, E. Néron, and O. Oliver. Ordonnancement d'ateliers à ressources multiples. In P. Lopez and F. Roubellat, editors, *Ordonnancement de la production*, pages 293–329. Hermès Science Publications, Paris, 2001.
- [BEN82] R. Bellman, A.O. Esogbue, and I. Nabeshima. *Mathematical aspects of scheduling and applications*. Pergamon Press, Oxford, 1982.
- [BHJW97] P. Brucker, Johann Hurink, B. Jurisch, and B. Wöstmann. A branch & bound algorithm for the open-shop problem. *Discrete Applied Mathematics*, 76:43–59, 1997.
- [BJK94] P. Brucker, B. Jurisch, and A. Kraemer. The job shop scheduling problem and immediate selections. *Annals of Operations Research*, 50:73–114, 1994.

- [BJK97] P. Brucker, B. Jurisch, and A. Kraemer. Complexity of scheduling problems with multi-purpose machines. *Annals of Operations Research*, 70:57–73, 1997.
- [BJS94] P. Brucker, B. Jurisch, and B. Sievers. A branch and bound algorithm for the job shop scheduling problem. *Discrete Applied Mathematics*, 49(1):107–127, 1994.
- [BKST98] P. Brucker, S. Knust, A. Schoo, and O. Thiele. A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107:272–288, 1998.
- [BL95] P. Baptiste and C. Le Pape. A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling. In *Proc. of the 14th IJCAI*, Montréal, Canada, 1995.
- [BL96] P. Baptiste and C. Le Pape. Edge-finding constraint propagation algorithms for disjunctive and cumulative scheduling. In *Proc. of U.K. planning and scheduling SIG meeting*, Liverpool, November 1996.
- [BLN95] P. Baptiste, C. Le Pape, and W.P.M. Nuijten. Constraint-based optimisation and approximation for job-shop scheduling. In *Proc. of the 14th IJCAI*, Montréal, Canada, 1995.
- [BLN01] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-based scheduling*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2001.
- [BMR88] M. Bartusch, R.H. Möhring, and F.J. Radermacher. Scheduling project networks with resource constraints time windows. *Annals of Operations Research*, 16:201–240, 1988.
- [BR96] J.-C. Billaut and F. Roubellat. Characterization of a set of schedules in a multiresource context. *Journal of Decision Systems*, 5(1-2):95–109, 1996.
- [Bra93] P. Brandimarte. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, 41:157–183, 1993.
- [Bru02] P. Brucker. Scheduling and constraint propagation. *Discrete Applied Mathematics*, 123:227–256, 2002.
- [Car75] J. Carlier. *Ordonnancement à contraintes disjonctives*. Thèse de troisième cycle, Université Paris VI, 1975.
- [CL94] Y. Caseau and F. Laburthe. Improved CLP scheduling with task intervals. In P. van Hentenryck, editor, *Logic Programming, Proceedings of the 11th International Conference on Logic Programming*, pages 369–383, Santa Margherita Ligure, Italy, 1994. MIT Press.
- [CL95] Y. Caseau and F. Laburthe. Improving branch and bound for jobshop scheduling with constraint propagation. In M. Deza, R. Euler, and Y. Manoussakis, editors, *Proc. of the 8th Franco-Japanese–4th Franco-Chinese Conference on Combinatorics and Computer Science*, Brest, France, July 1995.

- [CL96a] Y. Caseau and F. Laburthe. The claire programming language. Technical report, Laboratoire d'Informatique de l'Ecole Nationale Supérieure, 1996.
- [CL96b] Y. Caseau and F. Laburthe. Cumulative scheduling with task intervals. In *Proc. of the Joint International Conference and Symposium on Logic Programming*, 1996.
- [CN00] J. Carlier and E. Néron. A new LP-based lower bound for the cumulative scheduling problem. *European Journal of Operational Research*, 127(2):363–382, 2000.
- [CP89] J. Carlier and E. Pinson. An algorithm for solving the job-shop problem. *Management Science*, 35(2):164–176, 1989.
- [CP90] J. Carlier and E. Pinson. A practical use of Jackson's preemptive schedule for solving the job-shop problem. *Annals of Operations Research*, 26:269–287, 1990.
- [CP94] J. Carlier and E. Pinson. Adjustment of heads and tails for the job-shop problem. *European Journal of Operational Research*, 78:146–161, 1994.
- [CP01] J. Carlier and E. Pinson. Jackson pseudo preemptive schedule and cumulative scheduling problems. To appear in *Annals of Operations Research*, 2001.
- [DB97] R. Debruyne and C. Bessière. Some practical filtering techniques for the constraint satisfaction. In *Proc. of the 15th IJCAI*, pages 412–417, Nagoya, Japan, 1997.
- [Dew92] G. Dewess. An existence theorem for packing problems with implications for the computation of optimal schedules. *Optimization*, 25:261–269, 1992.
- [DH92] E. Demeulemeester and W. Herroelen. A branch and bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38:1803–1818, 1992.
- [DHP99] U. Dorndorf, T. Phan Huy, and E. Pesch. A survey of interval capacity consistency tests for time- and resource-constrained scheduling. In J. Weglarz, editor, *Project Scheduling: Recent Models, Algorithms and Applications*, pages 213–238. Kluwer, 1999.
- [DMP91] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
- [DPH00] U. Dorndorf, E. Pesch, and T. Phan Huy. Constraint propagation techniques for the disjunctive scheduling problem. *Artificial Intelligence*, 122:189–240, 2000.

- [DPRL98] S. Dauzère-Pérès, W. Roux, and J.B. Lasserre. Multi-resource shop scheduling with resource flexibility. *European Journal of Operational Research*, 107:289–305, 1998.
- [DSH90] M. Dincbas, H. Simonis, and P. van Hentenryck. Solving large combinatorial problems in logic programming. *The Journal of Logic Programming*, 8(1-2):74–94, 1990.
- [DHS⁺88] M. Dincbas, P. van Hentenryck, H. Simonis, A. Aggoun, and T. Graf. Application of CHIP to industrial and engineering problems. In *Proc. of the 1st international conference on industrial and engineering applications of artificial intelligence and expert systems*, Tullahoma, Tennessee, 1988.
- [DID97] DIDOM Group. The analysis of distributed decision in an industrial context: A constraint-based approach. In *7th mini EURO conference on Decision Support Systems, Groupware, Multimedia and Electronic Commerce*, Bruges, Belgium, March 1997.
- [EFR79] J. Erschler, G. Fontan, and F. Roubellat. Potentiels sur un graphe non conjonctif et analyse d’un problème d’ordonnement à moyens limités. *RAIRO-Recherche Opérationnelle/Operations Research*, 13(4):363–378, 1979.
- [EHL95] P. Esquirol, M.-J. Huguet, and P. Lopez. Modeling and managing disjunctions in scheduling problems. *Journal of Intelligent Manufacturing*, 6:133–144, 1995.
- [EK92] S.E. Elmaghraby and J. Kamburowski. The analysis of activity networks under generalized precedence relations (GPRs). *Management Science*, 38:1245–1263, 1992.
- [EL95] P. Esquirol and P. Lopez. Programmation logique avec contraintes et ordonnancement. *Automatique, Productique, Informatique Industrielle*, 29(4-5):379–407, 1995.
- [EL99] P. Esquirol and P. Lopez. *L’ordonnement*. Economica, Paris, 1999.
- [ELFS95] P. Esquirol, P. Lopez, H. Fargier, and T. Schiex. Constraint programming. *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Science)*, 35(2):5–36, 1995.
- [ELH01] P. Esquirol, P. Lopez, and M.J. Huguet. Propagation de contraintes en ordonnancement. In P. Lopez and F. Roubellat, editors, *Ordonnement de la production*, pages 131–167. Hermès Science Publications, Paris, 2001.
- [ELHS97] P. Esquirol, P. Lopez, L. Haudot, and M. Sicard. Constraint-oriented cooperative scheduling for aircraft manufacturing. *IEEE Expert, Intelligent Systems and their Applications*, 12(1):32–39, 1997.
- [ELM99] P. Esquirol, P. Lopez, and C. Mancel. Représentation et traitement du temps en ordonnancement. Technical Report 99455, LAAS, Toulouse, France, October 1999.

- [ELT91] J. Erschler, P. Lopez, and C. Thuriot. Raisonement temporel sous contraintes de ressources et problèmes d’ordonnancement. *Revue d’Intelligence Artificielle*, 5(3):7–36, 1991.
- [Ers76] J. Erschler. *Analyse sous contraintes et aide à la décision pour certains problèmes d’ordonnancement*. Thèse de Doctorat d’Etat, Université Paul Sabatier, Toulouse, 1976.
- [ERV76] J. Erschler, F. Roubellat, and J.-P. Vernhes. Finding some essential characteristics of the feasible solutions for a scheduling problem. *Operations Research*, 24(4):774–783, 1976.
- [ERV80] J. Erschler, F. Roubellat, and J.-P. Vernhes. Characterizing the set of feasible sequences for n jobs to be carried out on a single machine. *European Journal of Operational Research*, 4(3):189–194, 1980.
- [Esq87] P. Esquirol. *Règles et processus d’inférence pour l’aide à l’ordonnancement de tâches en présence de contraintes*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1987.
- [Far94] H. Fargier. *Problèmes de satisfaction de contraintes flexibles : application à l’ordonnancement de production*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1994.
- [FT63] H. Fisher and G.L. Thompson. Probabilistic learning combinations of local job-shop scheduling rules. In J.F. Muth and G.L. Thompson, editors, *Industrial Scheduling*, pages 225–251. Prentice Hall, Englewood Cliffs, NJ, 1963.
- [HBE⁺95] L. Haudot, J. Bradshaw, P. Esquirol, P. Lopez, and M. Sicard. A knowledge acquisition approach for the ergonomical design of cooperative systems: The SCOOP project experiment. In *Proc. of Fifth international conference on Human-Machine Interaction and Artificial Intelligence in AeroSpace (HMI-AI-AS’95)*, Toulouse, France, September 27-29 1995.
- [Hei03] R. Heilmann. A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research*, 144:348–365, 2003.
- [Hen89] P. van Hentenryck. *Constraint Satisfaction in Logic Programming*. Logic Programming Series. MIT Press, Cambridge, MA, 1989.
- [Hen94] P. van Hentenryck. Scheduling and packing in the constraint language CC(FD). In M. Zweben et M.S. Fox, editor, *Intelligent scheduling*, pages 137–167. Morgan Kaufmann, 1994.
- [HJT94] J. Hurink, B. Jurisch, and M. Thole. Tabu search for the job shop scheduling problem with multipurpose machines. *OR Spektrum*, pages 205–215, 1994.

- [HL99] M.-J. Huguet and P. Lopez. An integrated constraint-based model for task scheduling and resource assignment. In *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'99)*, Ferrara, Italie, February 1999.
- [HL00] M.-J. Huguet and P. Lopez. Mixed task scheduling and resource allocation problems. In *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'00)*, pages 71–79, Paderborn, Germany, March 2000.
- [HL02] M.-J. Huguet and P. Lopez. Constraint propagation techniques for mixed scheduling and allocation problems. Technical report, LAAS, Toulouse, France, 2002. Submitted to EJOR.
- [HLES95] L. Haudot, P. Lopez, P. Esquirol, and M. Sicard. Ergonomic design for cooperative scheduling system. In *Proc. of international workshop on the design of cooperative systems (COOP'95)*, pages 540–555, Antibes–Juan-les-Pins, France, January 1995.
- [HLV02] M.-J. Huguet, P. Lopez, and T. Vidal. Dynamic task sequencing in temporal problems with uncertainty. In *AIPS 2002 Workshop on "On-line Planning & Scheduling"*, Toulouse, France, April 2002.
- [JM99] A.S. Jain and S. Meeran. Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research*, 113:390–434, 1999.
- [Kis03] T. Kis. Job-shop scheduling with processing alternatives. *European Journal of Operational Research*, 151:307–332, 2003.
- [Kol95] R. Kolisch. *Project scheduling under resource constraints*. Production and logistics. Physica Verlag, Heidelberg, 1995.
- [Lab03] P. Laborie. Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results. *Artificial Intelligence*, 143(2):151–188, 2003.
- [LeP94] C. Le Pape. Implementation of resource constraints in ILOG-SCHEDULE: A library for the development of constraint-based scheduling systems. *Intelligent Systems Engineering*, 3:55–66, 1994.
- [LE96] P. Lopez and P. Esquirol. Consistency enforcing in scheduling: A general formulation based on energetic reasoning. In *Proc. of the 5th International Workshop on Project Management and Scheduling*, pages 155–158, Poznan, Poland, 1996.
- [LEE92] P. Lopez, J. Erschler, and P. Esquirol. Ordonnancement de tâches sous contraintes : une approche énergétique. *Automatique, Productique, Informatique Industrielle*, 26:453–481, 1992.

- [LEHS98] P. Lopez, P. Esquirol, L. Haudot, and M. Sicard. Cooperative system design in scheduling. *International Journal of Production Research*, 36(1):211–230, 1998.
- [LHE97] P. Lopez, L. Haudot, and P. Esquirol. Coopération homme-système en ordonnancement de production. In J.-C. Hennet, editor, *Concepts et outils pour les systèmes de production*, pages 161–186. Cépaduès Editions, Toulouse, 1997.
- [Lho93] O. Lhomme. Consistency techniques for numeric CSPs. In *Proc. of the 13th IJCAI*, pages 232–238, Chambéry, France, August 1993.
- [LLP96] M.-L. Levy, P. Lopez, and B. Pradin. A decomposition approach for the single-machine scheduling problem. *Journal of Decision Systems*, 5(1-2):73–94, 1996.
- [LLP98] P. Lopez, M.-L. Levy, and B. Pradin. Characterisation by decomposition in scheduling. *Computers in industry*, 36(1-2):113–116, 1998.
- [LLP99] M.-L. Levy, P. Lopez, and B. Pradin. Décomposition temporelle et caractérisation de solutions admissibles pour le problème d’ordonnancement à une machine. *RAIRO-Recherche Opérationnelle/Operations Research*, 33(2):185–208, 1999.
- [Lop91] P. Lopez. *Approche énergétique pour l’ordonnancement de tâches sous contraintes de temps et de ressources*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1991.
- [Lop96] P. Lopez. Conception d’un système coopératif en gestion de production. In G. de Terssac and E. Friedberg, editors, *Coopération et Conception*, pages 171–186. Octares Editions, Toulouse, 1996.
- [LR01] P. Lopez and F. Roubellat, editors. *Ordonnancement de la production*. Hermès Science Publications, Paris, 2001.
- [LT02] P. Lopez and P. Torres. Propagation de contraintes en ordonnancement : opérations globales ou techniques de « shaving ». In *ROADEF’2002*, pages 195–196, Paris, 20-22 Février 2002.
- [Mac77] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.
- [Mon74] U. Montanari. Network of constraints: Fundamental properties and applications to picture processing. *Information Science*, 7:95–132, 1974.
- [MS96] P. Martin and D.B. Shmoys. A new approach to computing optimal schedules for the job-shop scheduling problem. In *Proc. of the 5th International Conference on Integer Programming and Combinatorial Optimization, IPCO’96*, pages 389–403, Vancouver, British Columbia, June 1996. LNCS, volume 1084, Springer Verlag. W.H. Cunningham, S.T. McCormick, and M. Queyranne (editors).

- [Nui94] W.P.M. Nuijten. *Time and resource constrained scheduling - A constraint satisfaction approach*. PhD thesis, Eindhoven University of Technology, 1994.
- [Pau95] J. Paulli. A hierarchical approach for the FMS scheduling problem. *European Journal of Operational Research*, 86:32–42, 1995.
- [Pha00] T. Phan Huy. *Constraint propagation in flexible manufacturing*. LNEMS, volume 492, Springer Verlag, M. Beckmann, H. P. Künzi (editors), 2000.
- [Pér96] L. Péridy. *Le problème de job-shop : arbitrages et ajustements*. Thèse de Doctorat, Université de Technologie de Compiègne, 1996.
- [Roy70] B. Roy. *Algèbre moderne et théorie des graphes*, volume II. Dunod, Paris, 1970.
- [RS64] B. Roy and B. Sussman. Les problèmes d’ordonnancement avec contraintes disjonctives. Technical Report DS No. 9 bis, SEMA, Montrouge, 1964.
- [Sad94] N. Sadeh. Micro-opportunistic scheduling: The micro-boss factory scheduler. In M. Zweben and M.S. Fox, editors, *Intelligent scheduling*, pages 99–135. Morgan Kaufmann, San Francisco, 1994.
- [SC93] S. Smith and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proc. of AAAI-93*, pages 139–144, Washington, DC, 1993.
- [Sch98] E. Schwalb. *Temporal reasoning with constraints*. PhD thesis, University of California, Irvine, 1998.
- [SD97] E. Schwalb and R. Dechter. Processing disjunctions in temporal constraint networks. *Artificial Intelligence*, 93:29–61, 1997.
- [SD98] A. Sprecher and A. Drexl. Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 107:431–450, 1998.
- [SF94] D. Sabin and E. Freuder. Contradicting conventional wisdom in constraint satisfaction. In A. Borning, editor, *Proc. of PPCP-94*, Seattle, USA, May 1994.
- [SF96] N. Sadeh and M.S. Fox. Variable and value ordering heuristics for the job shop scheduling constraint satisfaction problem. *Artificial Intelligence*, 86:1–41, 1996.
- [SK00] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120:81–117, 2000.
- [SSW94] R. Słowiński, B. Soniewicki, and J. Węglarz. DSS for multiobjective project scheduling. *European Journal of Operational Research*, 79:220–229, 1994.
- [TL00a] P. Torres and P. Lopez. On not-first/not-last conditions in disjunctive scheduling. *European Journal of Operational Research*, 127(2):332–343, 2000.

- [TL00b] P. Torres and P. Lopez. Overview and possible extensions of shaving techniques for job-shop problems. In *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'00)*, pages 181–186, Paderborn, Germany, March 2000.
- [TLE99] P. Torres, P. Lopez, and P. Esquirol. Lattice of task intervals: A support for edge-finding in disjunctive scheduling. In *Proc. of International Conference on Industrial Engineering and Production Management (IEPM'99)*, pages 158–166 (Book 2), Glasgow, Scotland, 1999.

III – Annexes

A Curriculum vitæ

Etat civil

Nom : LOPEZ Pierre
Né le : 21 Août 1964 à Toulouse (39 ans)
Nationalité : française
Situation de famille : vie maritale (2 enfants)
Coordonnées professionnelles : LAAS-CNRS
7, av. du Colonel Roche – 31077 Toulouse cedex 4
Tél. 05 61 33 62 98 – Fax : 05 61 33 69 36
Mél : lopez@laas.fr
Url : <http://www.laas.fr/~lopez>

Titres universitaires

- Doctorat d'Automatique, Université Paul Sabatier (UPS), Toulouse, 1991.
Directeur de thèse : Prof. Jacques Erschler – Rapporteurs : J. Carlier, D. Dubois, J. Favrel – Examineurs : G. Bel, R. Companys, M. Courvoisier
- Diplôme d'Etudes Approfondies en Automatique, Informatique Industrielle et Traitement du Signal, UPS, Toulouse, 1987.
- Maîtrise d'Electronique, Electrotechnique et Automatique, UPS, Toulouse, 1986.

Historique professionnel

- Attaché Temporaire d'Enseignement et de Recherche à l'INPT/ENSEEIHHT du 01/10/91 au 30/09/92.
- Chargé de Recherche de deuxième classe au CNRS (CR2) du 01/10/92 au 30/09/96.
- Chargé de Recherche de première classe au CNRS (CR1) depuis le 01/10/96.
- Unité de Recherche :

UPR 8001, Laboratoire d'Analyse et d'Architecture des Systèmes
Directeur : Malik Ghallab
Toulouse (Tél. 05 61 33 62 00)
<http://www.laas.fr>

Langues

Anglais, Espagnol : lu, écrit, parlé

B Publications

Livres et ouvrages

- [1] Y. Dallery, J.-C. Hennet et P. Lopez, éditeurs. *Organisation et Conduite d'Activités dans l'Industrie et les Services*, Actes de la 4ème Conférence Francophone de Modélisation et Simulation (MOSIM'03), Toulouse, 23-25 avril 2003. SCS, The Society for Modeling and Simulation International, Erlangen, 2003.
- [2] P. Lopez et F. Roubellat, éditeurs. *Ordonnancement de la production*. Hermès Science Publications, Paris, 2001.
- [3] P. Esquirol et P. Lopez. *L'ordonnancement*. Economica, Paris, 1999.
- [4] P. Lopez. *Approche énergétique pour l'ordonnancement de tâches sous contraintes de temps et de ressources*. Thèse de Doctorat, Université Paul Sabatier, Toulouse, 1991.

Reuves à comité de lecture

- [5] P. Esquirol, P. Lopez, L. Haudot et M. Sicard. Constraint-oriented cooperative scheduling for aircraft manufacturing. *IEEE Expert, Intelligent Systems and their Applications*, 12(1):32–39, 1997.
- [6] P. Torres et P. Lopez. On not-first/not-last conditions in disjunctive scheduling. *European Journal of Operational Research*, 127(2):332–343, 2000.
- [7] P. Lopez, P. Esquirol, L. Haudot et M. Sicard. Cooperative system design in scheduling. *International Journal of Production Research*, 36(1):211–230, 1998.
- [8] P. Lopez, J. Erschler et P. Esquirol. Ordonnancement de tâches sous contraintes : une approche énergétique. *Automatique, Productique, Informatique Industrielle*, 26:453–481, 1992.

- [9] M.-L. Levy, P. Lopez et B. Pradin. Décomposition temporelle et caractérisation de solutions admissibles pour le problème d’ordonnancement à une machine. *RAIRO-Recherche Opérationnelle/Operations Research*, 33(2):185–208, 1999.
- [10] P. Esquirol, M.-J. Huguet et P. Lopez. Modeling and managing disjunctions in scheduling problems. *Journal of Intelligent Manufacturing*, 6:133–144, 1995.
- [11] P. Esquirol, P. Lopez, H. Fargier et T. Schiex. Constraint programming. *JORBEL (Belgian Journal of Operations Research, Statistics and Computer Science)*, 35(2):5–36, 1995.
- [12] J. Erschler, P. Lopez et C. Thuriot. Raisonnement temporel sous contraintes de ressources et problèmes d’ordonnancement. *Revue d’Intelligence Artificielle*, 5(3):7–36, 1991.
- [13] GOTH A. Les problèmes d’ordonnancement. *RAIRO-Recherche Opérationnelle / Operations Research*, 27(1):77–150, 1993.
- [14] M.-L. Levy, P. Lopez et B. Pradin. A decomposition approach for the single-machine scheduling problem. *Journal of Decision Systems*, 5(1-2):73–94, 1996.
- [15] P. Esquirol et P. Lopez. Programmation logique avec contraintes et ordonnancement. *Automatique, Productique, Informatique Industrielle*, 29(4-5):379–407, 1995.
- [16] C. Thuriot, J. Erschler et P. Lopez. Decision approach to workload distribution. *Production Planning and Control*, 5(6):533–542, 1994.
- [17] P. Lopez, M.-L. Levy et B. Pradin. Characterisation by decomposition in scheduling. *Computers in industry*, 36(1-2):113–116, 1998.

Conférences invitées dans des congrès

- [18] P. Lopez. Constraint Programming and Scheduling. In *XI Congreso Latino Iberoamericano de Investigacion de Operaciones (XI CLAIO)*, Concepción (Chili), 27-31 Octobre 2002. Conférence plénière invitée.
- [19] P. Lopez. Le concept de contraintes : propagation, satisfaction, programmation. In *Journées du Groupement de Recherche en Productique*, Toulouse, November 2001. Conférence plénière invitée.
- [20] P. Lopez, L. Haudot, P. Esquirol et M. Sicard. Conception d’un système coopératif en ordonnancement de production : une approche pluridisciplinaire. *5ème Congrès International de Génie Industriel (GI 5)*, pages 19–28, Grenoble, 1996. Conférence invitée.

Communications à des congrès, workshops (ordre anti-chronologique)

- [21] C. Mancel et P. Lopez. Complex optimization problems in space systems. In *13th International Conference on Automated Planning and Scheduling (ICAPS'03)*, Trento, Italie, June 2003.
- [22] C. Mancel, P. Lopez et R. Valette. Approche par génération de colonnes d'un problème d'optimisation de prises de vue satellite. In *4ème Conférence Francophone de MOdélisation et SIMulation (MOSIM'03)*, pages 122–127, Toulouse, 23-25 Avril 2003.
- [23] C. Mancel, M.-J. Huguet, P. Lopez, M. Medjoudj et R. Valette. Résolution intégrée d'ordonnancement et d'affectation : adaptation d'une méthode de recherche à divergence limitée à une approche fondée sur les réseaux de petri et la logique linéaire. In *4ème Conférence Francophone de MOdélisation et SIMulation (MOSIM'03)*, pages 293–298, Toulouse, 23-25 Avril 2003.
- [24] C. Mancel et P. Lopez. Approche par génération de colonnes du problème Pléiades. In *5ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'2003)*, pages 226–227, Avignon, 26-28 Février 2003.
- [25] C. Artigues et P. Lopez. On dominance and on-line properties of schedule generation schemes for the job-shop problem with sequence dependent-setup times. In *XI Congreso Latino Iberoamericano de Investigacion de Operaciones (XI CLAIO)*, Concepcion (Chili), 27-31 Octobre 2002.
- [26] O. Fournier, P. Lopez et J.-D. Lan Sun Luk. Cyclic scheduling following the social behavior of ant colonies. In *2002 IEEE International Conference on Systems Man and Cybernetics (SMC'02)*, Hammamet (Tunisie), 6-9 Octobre 2002.
- [27] C. Mancel, P. Lopez, N. Rivière et R. Valette. Relationships between Petri nets and constraint graphs: application to manufacturing. In *15th IFAC World congress on Automatic Control*, Barcelona, Spain, July 2002.
- [28] C. Artigues, P. Lopez et F. Buscaylet. Schedule generation schemes for the job-shop problem with setup times. In *Sixteenth triennial conference of the International Federation of Operational Research Societies (IFORS 2002)*, Edinburgh, Scotland, July 2002.
- [29] M.-J. Huguet et P. Lopez. A constraint-based procedure for scheduling and allocation with unrelated machines. In *8th International Workshop on Project Management and Scheduling*, pages 190–193, Valencia, Spain, April 2002.
- [30] M.-J. Huguet, P. Lopez et T. Vidal. Dynamic task sequencing in temporal problems with uncertainty. In *AIPS 2002 Workshop on On-line Planning and Scheduling*, Toulouse, April 2002.

- [31] P. Lopez et P. Torres. Propagation de contraintes en ordonnancement : opérations globales ou techniques de shaving. In *4ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF'2002)*, pages 195–196, Paris, 20-22 février 2002.
- [32] I. Sellami, M.J. Huguet et P. Lopez. A backtracking algorithm for solving mixed task scheduling and resource allocation problems. In *8th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'2001)*, volume 2, pages 741–744, Antibes-Juan-les-Pins, October 2001.
- [33] A. Helias, F. Guerrin, P. Lopez et J.-P. Steyer. Simulating management policies on stock supplied by multiple production units: application to a pig slurry treatment plant. In *3rd European Conference of the European Federation for Information Technology in Agriculture, Food and the Environment (EFITA 2001)*, Montpellier, June 2001.
- [34] C. Mancel, P. Lopez, R. Valette, M. Saleman et N. Bataille. Approaches for the optimization of the resources management in a satellite constellation. In *6th International Symposium on Artificial Intelligence, Robotics and Automation in Space: A New Space Odyssey (i-SAIRAS 2001)*, Montréal, Canada, June 2001.
- [35] T. Tournesol (nom collectif). Réactivité en entreprise. In *4th International Industrial Engineering Conference*, Aix-Marseille, June 2001.
- [36] M. Saleman, M. Faup, C. Mancel, P. Lopez et R. Valette. Optimization of the resource management in a satellite constellation: Progressive approaches and perspective for a hybrid global method. In *2nd International Workshop on Satellite Constellations and Formation Flying*, pages 1–8, Haifa, Israel, February 2001.
- [37] DIDOM Group. Planning by repairing and cooperation in multi-project management. In *2nd conference on Management and Control of Production and Logistics (MCPL'2000)*, Grenoble, July 2000.
- [38] M.-J. Huguet et P. Lopez. Approche par contraintes pour l'ordonnancement de tâches avec partage et affectation de ressources. In *Première Conférence Internationale Francophone d'Automatique (CIFA'2000)*, Lille, July 2000.
- [39] P. Torres et P. Lopez. Overview and possible extensions of shaving techniques for job-shop problems. *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'00)*, pages 181–186, Paderborn, Germany, March 2000.
- [40] M.-J. Huguet et P. Lopez. Mixed task scheduling and resource allocation problems. *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'00)*, pages 71–79, Paderborn, Germany, March 2000.

- [41] P. Torres et P. Lopez. Treillis d'intervalles de tâches et propagation de contraintes. *3ème congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF'2000)*, pages 144–145, Nantes, 26-28 Janvier 2000.
- [42] P. Torres, P. Lopez et P. Esquirol. Lattice of task intervals: A support for edge-finding in disjunctive scheduling. *International Conference on Industrial Engineering and Production Management (IEPM'99)*, pages 158–166 (Book 2), Glasgow, Scotland, 1999.
- [43] M.-J. Huguet et P. Lopez. An integrated constraint-based model for task scheduling and resource assignment. *Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'99)*, Ferrara, Italie, February 1999.
- [44] J. Erschler, P. Esquirol, M.-J. Huguet, P. Lopez et F. Roubellat. Aide à la décision et à la coopération basée sur les contraintes : illustration en ordonnancement. *2ème congrès de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF'99)*, page 72, Autrans, 13-15 Janvier 1999.
- [45] P. Torres et P. Lopez. A generic algorithm for solving the not-first/not-last problem in disjunctive scheduling. *6th International Workshop on Project Management and Scheduling (PMS'98)*, pages 305–308, Istanbul, Turkey, 1998.
- [46] J.-C. Billaut et P. Lopez. Enumeration of all optimal sequences in the two-machine flowshop. *Computational Engineering in Systems Applications (CESA'98)*, pages 378–382, Hammamet, Tunisia, April 1998.
- [47] DIDOM Group. The analysis of distributed decision in an industrial context: A constraint-based approach. *7th mini EURO conference on Decision Support Systems, Groupware, Multimedia and Electronic Commerce*, Bruges, Belgium, March 1997.
- [48] P. Lopez et P. Esquirol. Consistency enforcing in scheduling: A general formulation based on energetic reasoning. *5th International Workshop on Project Management and Scheduling (PMS'96)*, pages 155–158, Poznan, Poland, 1996.
- [49] M.-L. Levy, P. Lopez et B. Pradin. Characterisation by decomposition of the one-machine scheduling problem. *1996 Advanced Summer Institute (Annual conference of ICIMS-NOE) (ASI'96)*, pages 148–153, Toulouse, Juin 1996.
- [50] M.-L. Levy, P. Lopez et B. Pradin. Décomposition temporelle du problème à une machine. *AGI'96*, pages 159–162, Tours, Juin 1996.
- [51] L. Haudot, J. Bradshaw, P. Esquirol, P. Lopez et M. Sicard. A knowledge acquisition approach for the ergonomical design of cooperative systems: The SCOOP project experiment. *Fifth international conference on Human-Machine Interaction and Artificial Intelligence in AeroSpace (HMI-AI-AS'95)*, Toulouse, France, September 27-29 1995.

- [52] P. Lopez, L. Haudot, M. Sicard et P. Esquirol. A constraint-based approach to design a DSS for scheduling. *3rd International Conference on the Practical Application of Prolog (PAP'95)*, pages 405–422, Paris, France, 1995.
- [53] M.-L. Levy, P. Lopez et B. Pradin. Décomposition temporelle pour la résolution de problèmes d'ordonnancement d'atelier. *International conference on ordinal and symbolic data analysis (OSDA '95)*, pages 45–48, Paris, Juin 1995.
- [54] L. Haudot, M. Sicard, P. Esquirol et P. Lopez. Approche par contraintes : application à un problème de gestion de production. *IVèmes Journées Francophones de Programmation en Logique (JFPL'95)*, pages 329–332, Dijon, Mai 1995.
- [55] M.-L. Levy, P. Lopez et B. Pradin. A decomposition approach to characterize feasible schedules for the one-machine problem. *International Conference on Industrial Engineering and Production Management (IEPM'95)*, volume 1, pages 402–411, Marrakech, Morocco, April 4-7 1995.
- [56] L. Haudot, P. Lopez, P. Esquirol et M. Sicard. Ergonomic design for cooperative scheduling system. *International workshop on the design of cooperative systems (COOP'95)*, pages 540–555, Antibes–Juan-les-Pins, France, January 1995.
- [57] M.-L. Levy, P. Lopez et B. Pradin. Characterization of feasible schedules for the flow-shop problem: A decomposition approach. *European Workshop on Integrated Manufacturing Systems Engineering (IMSE'94)*, pages 307–315, Grenoble, France, December 12-14, 1994.
- [58] P. Lopez et P. Esquirol. Using constraint programming to design a cooperative system in production scheduling. *4th International Workshop on Project Management and Scheduling (PMS'94)*, pages 153–157, Leuven, Belgique, July 1994.
- [59] M.-L. Levy, P. Lopez et B. Pradin. Une approche par décomposition pour la caractérisation d'ordonnements admissibles. *AGI'94*, pages 189–192, Poitiers, Juin 1994.
- [60] P. Esquirol, P. Lopez, L. Haudot et M. Sicard. Un système coopératif en ordonnancement de production. *AGI'94*, pages 181–184, Poitiers, Juin 1994.
- [61] P. Esquirol, M.-J. Huguet et P. Lopez. An algorithm for characterizing feasible schedules under non-conjunctive constraints. *International Conference on Industrial Engineering and Production Management (IEPM'93)*, pages 120–129, Mons, Belgique, Juin 1993.
- [62] P. Esquirol, M.-J. Huguet et P. Lopez. Time bounds on node graph for scheduling. *3rd International Workshop on Project Management and Scheduling (PMS'92)*, Como, Italie, 1992.
- [63] J. Erschler et P. Lopez. Energy-based approach for task scheduling under time and resource constraints. *2nd International Workshop on Project Management and Scheduling (PMS'90)*, Compiègne, France, 20-22 Juin 1990.

- [64] C. Thuriot, J. Erschler et P. Lopez. Problème d’ordonnancement : nouveaux concepts et modèle pour l’aide à la décision de répartition des charges. *International Conference CIM90. Productique & Integrations*, pages 151–157, Bordeaux, Juin 1990.
- [65] J. Erschler, P. Lopez et C. Thuriot. Scheduling under time and resource constraints. *Workshop on Manufacturing Scheduling, 11th IJCAI*, Detroit, USA, 1989.

Contribution à ouvrages

- [66] P. Esquirol et P. Lopez. Concepts et méthodes de base en ordonnancement de la production. In P. Lopez et F. Roubellat, éditeurs, *Ordonnancement de la production*, pages 25–53. Hermès Science Publications, Paris, 2001.
- [67] P. Esquirol, P. Lopez et M.J. Huguet. Propagation de contraintes en ordonnancement. In P. Lopez et F. Roubellat, éditeurs, *Ordonnancement de la production*, pages 131–167. Hermès Science Publications, Paris, 2001.
- [68] P. Esquirol et P. Lopez. Analyse sous contraintes des problèmes d’ordonnancement. In J.-C. Hennet, éditeur, *Concepts et outils pour les systèmes de production*, pages 133–160. Cépaduès Editions, Toulouse, 1997.
- [69] P. Lopez, L. Haudot et P. Esquirol. Coopération homme-système en ordonnancement de production. In J.-C. Hennet, éditeur, *Concepts et outils pour les systèmes de production*, pages 161–186. Cépaduès Editions, Toulouse, 1997.
- [70] P. Lopez. Conception d’un système coopératif en gestion de production. In G. de Terssac et E. Friedberg, éditeurs, *Coopération et Conception*, pages 171–186. Octares Editions, Toulouse, 1996.
- [71] J. Erschler, P. Lopez et C. Thuriot. Temporal reasoning under resource constraints: Application to task scheduling. G.E. Lasker & R.R. Hough, éditeurs, *Advances in support systems research*, pages 189–194. The International Institute for Advanced Studies in Systems Research and Cybernetics, 1990.

Séminaires

- [72] P. Lopez. Propagation de contraintes en ordonnancement. *Groupe de travail Bermudes*, Toulouse, octobre 1998.
- [73] P. Esquirol, L. Haudot, P. Lopez, M. Sicard. Coopération Homme-Système en ordonnancement de production. *Journée “Ordonnancement coopératif” du GDR Automatique*, Bordeaux, Janvier 1996.
- [74] M. Sicard, L. Haudot, P. Lopez, P. Esquirol, G. Boy et J. Bradshaw. SCOOP - Système coopératif pour l’ordonnancement de production. *3ème Colloque Bilan Actions de Recherche en Productique et Robotique (A2RP)*, pages 341–355, Paris, 1995.

- [75] P. Lopez. Travail coopératif et gestion de production. *Séminaire GDR-CNRS FROG "Coopération et Conception"*, Toulouse, Décembre 1994.

Logiciels

- [76] C. Briand, J.-E. Doucet, P. Esquirol, M.-J. Huguet et P. Lopez. Projet de plate-forme logicielle LORA. Spécification 1.1. Représentation de problèmes d'ordonnement de tâches et d'affectation de ressources. Rapport LAAS No 01551, December 2001.

Autres

- [77] C. Artigues, P. Lopez et P.-D. Ayache. Schedule generation schemes and priority rules for the job-shop problem with sequence-dependent setup times: dominance properties and computational analysis. Rapport LAAS No 03225, Toulouse, Mai 2003. Submitted to *Annals of Operations Research*
- [78] M.-J. Huguet et P. Lopez. Constraint propagation techniques for mixed scheduling and allocation problems. Rapport LAAS No 03285, Toulouse, 2003. Submitted to *Constraints: An International Journal*.
- [79] C. Mancel, P. Lopez et R. Valette. Formalisation des besoins pour la spécification fonctionnelle d'un outil d'aide à l'élaboration de plans de travail dans le cadre du projet Netlander. Contrat CNES, rapport LAAS No 01473, Toulouse, Novembre 2001.
- [80] P. Lopez et R. Valette. Propagation de contraintes pour l'optimisation des ressources satellites dans les constellations. Contrat CNES, rapport LAAS No 00419, Toulouse, Octobre 2000.
- [81] R. Valette, R. Champagnat et P. Lopez. Modélisation des "handovers" par réseaux de Petri et optimisation des ressources satellites. Contrat CNES, rapport LAAS No 99350, Toulouse, Septembre 1999.
- [82] P. Lopez. Programmation avec contraintes pour l'ordonnement ? *Bulletin de l'AFIA n° 23*, 1995.

C Participation à projets



Intitulé : Système COopératif pour l'Ordonnement de Production

Dates : 1994–1996

Partenaires : DASSAULT Aviation, EURISCO (Institut Européen de l'Ingénierie et des Sciences Cognitives)

Financement : Ministère de l'Enseignement Supérieur et de la Recherche

Objectifs : Gestion d'un atelier (fortement perturbé) de pièces de tôlerie cambrée.

Les principes mis en œuvre sont ceux de l'analyse sous contraintes dont le but est de caractériser l'ensemble des ordonnancements admissibles, voire de détecter une inconsistance dans le graphe de contraintes. Des techniques de classification hiérarchique des données sont utilisées de manière à guider l'opérateur humain dans sa décision de placement et d'ordonnement parmi les choix offerts par les résultats d'analyse.

Ceci a donné lieu à la réalisation d'une première maquette pour le problème de l'engagement des pièces dans l'atelier. Parallèlement, une deuxième maquette permet d'aborder le problème d'ordonnement de type Job-shop à l'aide d'un diagramme de Gantt interactif où l'utilisateur peut obtenir une solution, mais aussi tester l'impact d'une décision, remettre en cause un choix, etc.

Ce projet a permis d'initier un travail de recherche plus fouillé sur la coopération Homme-Système. On formalise ainsi les processus de coopération en termes de dialogue, de soutien, de contrôle et de distribution des décisions.

La programmation utilise le langage CHIP.

Références : [5, 7, 20, 51, 52, 54, 56, 58, 60, 69, 70, 73, 74, 75].

★

◇ DIDOM

Intitulé : DIstribution de la Décision dans les Organisations Matricielles

Dates : 1997–1999

Partenaires : IRIT, ONERA CEntRe de Toulouse, Matra Marconi Space, CERTOP

Financement : Département « Sciences Pour l'Ingénieur » du CNRS (programme Prosper), MENRT et région Midi-Pyrénées

Objectifs : Coordination des décisions dans des structures mettant en relation plusieurs centres de décision.

Etude des caractéristiques des organisations matricielles où cohabitent des « structures métiers » (regroupement des savoir-faires et capitalisation de l'expertise) et des « structures projets » (regroupement au sein d'équipes pluridisciplinaires des compétences diverses nécessaires à la réalisation du projet).

Modélisation du problème de gestion multiprojets.

Contexte applicatif : intégration de satellites.

Références : [37, 47].

★

◇ Handoffs

Intitulé : Optimisation des ressources satellites dans une constellation : modélisation des *handovers*

Dates : 1999–2000

Financement : Centre National d'Etudes Spatiales

Objectifs : Optimisation de l'affectation de faisceaux de satellites de communication à des utilisateurs relayés par des stations au sol.

Les satellites étant « défilants », des commutations — ou « handovers » — de satellites sont nécessaires pour garantir une bonne qualité de transmission. Pendant la commutation, deux satellites doivent être visibles simultanément par la station au sol. Un objectif est de minimiser le nombre de handovers.

Un tel problème a été considéré comme un problème d'ordonnancement dans lequel il s'agit de déterminer les dates de début et de fin des affectations des satellites aux utilisateurs, affectations qui assurent la continuité des communications et qui minimisent le nombre de commutations.

Formulation d'un programme linéaire permettant d'obtenir une solution optimale pour un sous-problème de taille restreinte par rapport à la couverture terrestre globale.

Références : [34, 36, 80, 81].

★

◇ **Netlander**

Dates : 2001 (1 an)

Financement : Centre National d'Etudes Spatiales

Objectifs : Optimisation du processus de programmation des expériences d'une mission d'exploration martienne.

Décomposition du problème global en deux sous-problèmes : affectation de communications sonde/satellite et planification des expériences.

Pour modéliser et résoudre le premier sous-problème, la programmation linéaire a été utilisée avec succès.

Le problème de planification des expériences de la mission est complexe et combinatoire car il met en jeu de nombreuses variables et contraintes hétérogènes et il s'agit plus d'un problème d'aide à la décision que d'optimisation. Analyse des besoins en termes d'outil et proposition de mécanismes de propagation de contraintes adaptés (raisonnement énergétique).

Références : [21, 79].

★

◇ **Pléiades**

Dates : 2002 (1 an)

Contexte : Challenge ROADEF'2003 (Société Française de Recherche Opérationnelle et d'Aide à la Décision)

Objectifs : Optimisation de la programmation des prises de vue de satellites d'observation — de la Terre — de nouvelle génération

Utilisation de la programmation linéaire et de certaines de ses extensions, comme la génération de colonnes, ainsi que différentes techniques de propagation de contraintes.

Références : [21, 22, 24].

★

◇ LORA

Intitulé : Logiciel d'ORdonnancement et d'Affectation

Dates : octobre 2000–en cours

Partenaires : Service « Informatique et Instrumentation » du LAAS

Financement : LAAS–CNRS

Objectifs : Réalisation d'une plateforme logicielle d'expérimentation, de validation et d'évaluation de méthodes d'ordonnancement de tâches et d'allocation de ressources. Définition des relations qui existent entre processus, ressources et temps. La représentation des processus dans le temps est considérée en introduisant un modèle général et original, les « diagrammes de tâches », permettant de spécifier des alternatives. Ce modèle intègre de façon inédite dans un même formalisme graphique trois types d'exigences : représentation précise de contraintes temporelles entre deux événements (début ou fin de tâches), représentation d'alternatives de réalisation et de contraintes de séquençage, et représentation de relations de composition-abstraction (tâches, macro-tâches).

Définition d'un langage général pour la représentation des processus de réalisation (d'un produit ou d'un service), des moyens, des savoir-faire, des objectifs et des solutions. A cet égard, l'utilisation de langages existants tels que PSL (Process Specification Language), KIF (Knowledge Interchange Format) ou XML (Extensible Markup Language) ont été étudiés. Le choix s'est porté sur ce dernier.

Références : [76].

★

D Actions pour la communauté scientifique

D.1 Encadrement doctoral

- Catherine MANCEL sur « Optimisation de la gestion de ressources : contraintes spatio-temporelles et approches hybrides », Doctorat de l'INSAT – convention CIFRE (GFI/LAAS), depuis novembre 2000. *Encadrement à 80 %*.
- Philippe TORRES sur « Structures et mécanismes pour la propagation de contraintes en ordonnancement », octobre 1997–septembre 2001. *Encadrement à 100 %*.
- Arnaud HELIAS sur « Agrégation/abstraction de modèles pour le pilotage de réseaux de flux hybrides à différents niveaux d'organisation et échelles spatiales – Cas des flux de biomasse agricole à la Réunion », depuis janvier 2001, Doctorat INRA/CIRAD financé par la Région Réunion et le Fonds Social Européen. *Comité de thèse*.
- Marie-Luce LEVY sur « Méthodes par décomposition temporelle et problèmes d'ordonnancement », Doctorat de l'INPT, mars 1996. *Encadrement à 60 %*.
- Luc HAUDOT sur « Une approche orientée utilisateur pour la conception de systèmes coopératifs en ordonnancement de production », Doctorat de l'INSAT, mars 1996. *Encadrement à 30 %*.

D.2 Rayonnement scientifique

Jurys de thèse

- Olivier Fournier : « Conception de la commande d'un système automatisé de production : Apport des graphes et de l'ordonnancement cyclique », Université de la Réunion, mars 2002.
- Vincent T'kindt, « Etude des problèmes d'ordonnancement multicritères », E3I, Université François Rabelais de Tours, décembre 1999.
- Marie-Luce Levy, « Méthodes par décomposition temporelle et problèmes d'ordonnancement », LAAS, Toulouse, Mars 1996.
- Bernard Penz, « Constructions agrégatives d'ordonnements pour des job-shops dynamiques et réactifs », Artemis-IMAG, Grenoble, Décembre 1994.

Invitations

- Invitation au Laboratoire de Génie Industriel (LGI/ACTES), Université de La Réunion, St-Denis, 1-12 octobre 2001 et 9-18 décembre 2002.
- Conférence plénière invitée à CLAIO-XI (congrès ALIO, association latino-ibéroaméricaine de Recherche Opérationnelle), 29 octobre 2002, Concepción, Chili, « Constraint Programming and Scheduling ».
- Conférence plénière invitée aux Journées du Groupement de Recherche en Productique, 8 novembre 2001, Toulouse, « Le concept de contraintes : propagation, satisfaction, programmation ».
- Organisateur et président de séance d'une session invitée de « Advanced Summer Institute (ASI'96) », conférence annuelle du réseau européen ICIMS-NOE, sur : « Some applications of scheduling techniques ».

Communautés

- Co-animateur du groupe de travail « Contraintes et Recherche Opérationnelle » du pôle « Optimisation Combinatoire » du GDR Algorithmique, Langage et Programmation¹¹.
- Co-éditeur des Actes de la Conférence MOSIM'03, Toulouse, 23-25 avril 2003 ; membre du comité d'organisation, gestion du site Web¹².
- Membre de la Société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF).
- Participation au groupe de travail Bermudes sur les problèmes d'ordonnancement. Organisation locale de la réunion à Toulouse en octobre 1998 et en septembre 2003.
- Correspondant du GOTHa (Groupe de recherche en Ordonnancement Théorique et Appliqué) jusqu'en 1999 : organisation de séminaires à l'Institut Blaise Pascal, gestion du site Web¹³, diffusion de l'information, etc.

¹¹<http://www.emn.fr/x-info/jussien/GT>

¹²<http://www.laas.fr/mosim03>

¹³<http://www.laas.fr/~lopez/gotha/Gotha.html>

Expertises

- Lecteur d'articles pour des revues scientifiques : Computers & Industrial Engineering, Discrete Applied Mathematics, European Journal of Control, European Journal of Operational Research, IEEE Transactions on Neural Networks, International Journal of Computer Integrated Manufacturing, International Journal of Production Research, Journal Européen des Systèmes Automatisés, Journal of Heuristics, Journal of Intelligent Manufacturing.
- Expert de projets scientifiques déposés à des Conseils Régionaux.

D.3 Gestion de la recherche

- Responsable du groupe « Modélisation, Optimisation et Gestion Intégrée de Systèmes d'Activités » (14 personnes) du LAAS, depuis avril 2003.
- Responsable de l'équipe « Ordonnancement de Tâches et Allocation de Ressources » (12 personnes) au sein du groupe « Organisation et Conduite de Systèmes Discrets » du LAAS, d'avril 1997 à mars 2003.
- Membre nommé de la commission de spécialistes (27ème section), Ecole Polytechnique de l'Université de Tours.

D.4 Enseignements dispensés

- Enseignements (volume horaire annuel)
 - * « Théorie des Graphes », CM + TD, 25h, 2^{ème} cycle.
 - * « Ordonnancement », CM, 20h, 3^{ème} cycle (DEA).
 - * « Gestion de Production », CM + TP, 56h, 2^{ème} et 3^{ème} cycles.
 - * « Simulation des Systèmes à événements discrets », TP, 26h, 3^{ème} cycle.
 - * « Graphes, Flots et Ordonnancement », CM + TD, 32h, 3^{ème} cycle.
- Etablissements : Ecole Nationale d'Ingénieurs en Electrotechnique, Electronique Industrielle, Informatique, Hydraulique de Toulouse (INPT/ENSEEIH) (de 1989 à 1994) ; Institut National des Sciences Appliquées de Toulouse (INSAT) (de 1989 à 2000) ; Université Polytechnique de Timisoara (Roumanie) (1998) ; Université Paul Sabatier (UPS) (depuis 1999) ; Université de la Réunion (depuis 2001) ; Faculté des Sciences de Tunis (depuis 2001) ; Ecole d'Eté TEC de Monterrey, Mexique (2002 et 2003).

E Propagation des conditions « non première / non dernière »

E.1 Travaux existants

A notre connaissance, Carlier-Pinson furent les premiers à utiliser les conditions NP/ND [CP89]. Encapsulées dans leur schéma de branchement, les conditions NP/ND sont utilisées à chaque nœud pour éliminer des tâches ne pouvant être séquencées premières ou dernières parmi les tâches non encore ordonnées. Ainsi, une liste de tâches possible-ment premières est caractérisée pour le branchement. Cela est effectué très rapidement et concourt à une réduction drastique de l'arbre de recherche. Néanmoins, les déductions sont incomplètes et l'unicité du point fixe n'est pas garantie si ces règles sont utilisées en conjonction avec d'autres règles. De plus, ces informations de séquençement impossibles ne sont utiles que dans le cadre d'un schéma de branchement « premier/dernier », qui consiste à séquencer une clique de disjonctions par ses extrémités [BL95]. Ainsi, l'incomplétude des ajustements par rapport à la règle, ainsi que sa dépendance à la technique de résolution ne permet pas d'utiliser de façon modulaire la propagation des conditions NP/ND pour réduire les fenêtres des tâches.

Nuijten [Nui94] présente une version modifiée de son algorithme permettant de trouver la totalité des ajustements liés à la règle (7) afin d'effectuer aussi des ajustements liés à la règle (3). Cet algorithme hybride (il propage deux types de contraintes — conjonctives et non conjonctives) a une complexité temporelle en $O(n^3)$ mais ne déduit pas tous les ajustements possibles liés à la règle (3).

Nous proposons ici une version modifiée de ce dernier algorithme, permettant désormais d'effectuer tous les ajustements possibles, tout en conservant la même complexité. Nous proposons également un nouvel algorithme quadratique, permettant de trouver tous les ajustements liés aux règles NP/ND (cet algorithme a été présenté initialement dans [TL00a]).

★

E.2 Modification de l'algorithme calcLB2 de Nuijten

Dans [Nui94], un nouvel algorithme de propagation des sélections immédiates de Carlier-Pinson permettant de calculer les ajustements (8), nommé calcLB1, est proposé. Il est ensuite modifié pour rechercher des ajustements liés à la condition (3) en plus des ajustements conjonctifs. Cette extension, appelée calcLB2, permet de détecter tous les ajustements liés à la règle (7) mais reste incomplète vis-à-vis de la règle (3). Nous proposons ici une nouvelle modification de cet algorithme le rendant complet vis-à-vis des deux règles.

Tout d'abord, l'algorithme 4 (calcLB2) est présenté et commenté.

- Si l'on supprime les lignes 27-38, l'algorithme 4 calcule tous les ajustements (8) liés aux ensembles ascendants (7). Cette partie de l'algorithme est équivalente à l'algorithme

Algorithm 4 calcLB2**requiert** : Liste X des tâches de \mathcal{T} triées dans l'ordre non décroissant des r_i **requiert** : Liste Y des tâches de \mathcal{T} triées dans l'ordre non décroissant des d_i

```

1: Vecteur d'entiers  $ECT$  de taille  $n$  initialisé à  $+\infty$ 
2: pour  $y \leftarrow 1$  à  $|T|$  faire
3:   si  $y = |T|$  ou  $d_{Y[y]} \neq d_{Y[y+1]}$  alors
4:      $P \leftarrow 0; g \leftarrow -\infty$ 
5:      $ECT[|S|] \leftarrow +\infty$ 
6:     pour  $i \leftarrow |T|$  à 1 faire
7:       si  $d_{X[i]} \leq d_{Y[y]}$  alors
8:          $P \leftarrow P + p_{X[i]}$ 
9:          $g \leftarrow \max(g, r_{X[i]} + P)$ 
10:         $ECT[i] \leftarrow \min(ECT[i], r_{X[i]} + p_{X[i]})$ 
11:      fin si
12:       $G[i] \leftarrow g$ 
13:    fin pour
14:     $H \leftarrow -\infty$ 
15:    pour  $x \leftarrow 1$  à  $|T|$  faire
16:      si  $d_{X[x]} > d_{Y[y]}$  alors
17:        si  $r_{X[x]} + P + p_{X[x]} > d_{Y[y]}$  alors
18:           $LB[X[x]] \leftarrow \max(r_{X[x]}, G[x])$ 
19:        fin si
20:        si  $H + p_{X[x]} > d_{Y[y]}$  alors
21:           $LB[X[x]] \leftarrow \max(r_{X[x]}, G[1])$ 
22:        fin si
23:      sinon
24:         $H \leftarrow \max(H, r_{X[x]} + P)$ 
25:         $P \leftarrow P - p_{X[x]}$ 
26:      fin si
27:       $P' \leftarrow P$ 
28:      pour  $w \leftarrow x - 1$  à 1 faire
29:        si  $d_{X[w]} \leq d_{Y[y]}$  alors
30:          si  $ECT[w] \leq r_{X[x]}$  alors
31:            break
32:          fin si
33:           $P' \leftarrow P' + p_{X[w]}$ 
34:          si  $P' + p_{X[x]} > d_{Y[y]} - r_{X[x]}$  alors
35:             $LB[X[x]] \leftarrow \max(r_{X[x]}, ECT[w])$ 
36:          fin si
37:        fin si
38:      fin pour
39:    fin pour
40:  fin si
41: fin pour

```

primal de Martin et Shmoys qui permet d'obtenir tous les ajustements (8), en utilisant implicitement une structure d'intervalles de tâches.

– Lignes 27-38 : calcul d'une partie des ajustements liés à la règle (3).

La complexité temporelle de `calcLB2` est $O(n^3)$. Pour permettre l'obtention de tous les ajustements liés à la règle (3), `calcLB2` est modifié, tout en gardant une complexité en $O(n^3)$, en remplaçant les lignes 27-38 par les lignes suivantes :

Algorithm 5 `calcLB2` modifié

```

1:  $P' \leftarrow P; ECT_{min} \leftarrow +\infty$ 
2: pour  $w \leftarrow x - 1$  à 1 faire
3:   si  $d_{X[w]} \leq d_{Y[y]}$  alors
4:     si  $ECT[w] > r_{X[x]}$  alors
5:        $ECT_{min} \leftarrow \min(ECT_{min}, ECT[w])$ 
6:        $P' \leftarrow P' + p_{X[w]}$ 
7:     si  $P' + p_{X[x]} > d_{Y[y]} - r_{X[x]}$  alors
8:        $LB(r_{X[x]}) \leftarrow \max(LB(r_{X[x]}), ECT_{min})$ 
9:     fin si
10:  fin si
11: fin si
12: fin pour

```

S'il est intéressant dans le cadre d'un outil de propagation de contraintes générique de disposer d'un algorithme complet pour chaque type de contrainte propagée afin de respecter la propriété d'unicité du point fixe, il est aussi souhaitable de pouvoir disposer d'outils de propagation efficaces et flexibles. Pour cela, on cherche généralement à propager chaque type de contraintes de façon séparée à l'aide d'algorithmes dédiés en respectant le principe « un algorithme par règle ». Or, si l'algorithme 5 respecte l'unicité du point fixe par rapport aux règles (7) et (3), sa complexité temporelle est relativement mauvaise et il ne respecte pas le principe de propagation séparée pour chaque type de contraintes.

E.3 Un nouvel algorithme

La formulation classique de la règle (3) utilise la borne $lst(S) = \max_{s \in S} d_s - \sum_{s \in S} p_s$ pour estimer la date de début au plus tard d'une clique de disjonction non ordonnée S . Une borne plus précise est définie par $lst_2(S)$, calculée de la façon suivante :

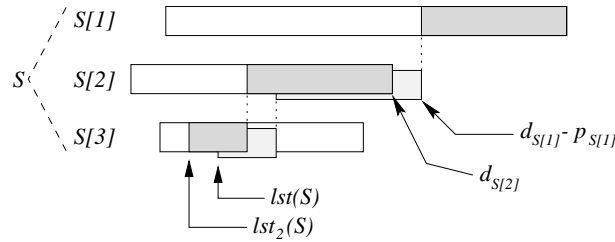
Algorithm 6 lst_2

requiert : Liste S des tâches de \mathcal{T} triées selon l'ordre non croissant des d_i

```

1:  $lst_2(S) \leftarrow +\infty$ 
2: pour  $k := 1$  à  $|S|$  faire
3:    $lst_2(S) \leftarrow \min[d_{S[k]}, lst_2(S)] - p_{S[k]}$ 
4: fin pour

```

FIG. 18 – Borne supérieure $lst_2(S)$

La borne supérieure $lst_2(S)$ est meilleure que $lst(S)$ (figure 18) ; dans le pire des cas, $lst_2(S) = lst(S)$ et s'il existe une itération k (ligne 2) telle que $d_{S[k]} < lst_{\{S[1], \dots, S[k-1]\}}$, alors $lst_2(S) < lst(S)$. Si les tâches sont déjà ordonnées, $lst_2(S)$ peut remplacer $lst(S)$ dans la règle pour le même coût linéaire.

L'intérêt principal de cette borne est de fournir une condition suffisante d'inexistence de sous-ensembles de S susceptibles de déclencher la règle (3). Ainsi, si un couple (i, S) ne déclenche pas la règle NP, il n'existe pas de sous-ensemble S' de S tel que (i, S') déclenche la règle (3). Cela entraîne la proposition suivante :

Soit $S \subset \mathcal{T}$, $i \in \mathcal{T} \setminus S$:

$$r_i + p_i \leq lst_2(S) \implies \nexists S' \subset S \text{ tel que } r_i + p_i > lst_2(S') \quad (23)$$

Preuve. Evident puisque $S' \subset S \implies lst_2(S) \leq lst_2(S')$. \square

Cas Particulier. $\forall i \in \mathcal{T}$, soit $T^i = \mathcal{T} \setminus \{i\}$:

$$\left. \begin{aligned} r_i + p_i \leq lst_2(T^i) &\implies \nexists S' \subset T^i \text{ tel que } r_i + p_i > lst_2(S') \\ &\implies \nexists S' \subset T^i \text{ tel que } i \text{ NP dans } S' \cup \{i\} \end{aligned} \right\} \quad (24)$$

La condition (24) fournit en temps linéaire (si les tâches sont déjà ordonnées) une condition suffisante d'arrêt à la recherche d'ajustements NP pour une tâche donnée.

Un algorithme exploitant cette propriété pour trouver tous les ajustements (4) est décrit ci-dessous.

- La ligne 2 initialise le tableau S .
- La boucle principale (lignes 3-17) itère sur les tâches que l'on essaie de prouver « non première ».
- La boucle interne (lignes 6-16) examine les tâches pouvant contribuer à définir un ensemble sur lequel une condition NP pourra être détectée pour $S[i]$. Cette boucle est itérée jusqu'à ce que toutes les tâches soient examinées ou que le drapeau « Continue » soit faux.
- Le test de la ligne 7 sélectionne une tâche $S[j]$ si et seulement si $S[j]$ peut entraîner un ajustement de $r_{S[i]}$. Cela permet de ne chercher que les conditions NP actualisantes.
- La ligne 8 calcule la borne présentée en (18). Le test de la ligne 10 permet la détection d'un ajustement NP pour $S[i]$ qui est mémorisé ligne 11.

Algorithm 7 NP

requiert : Liste S des tâches de \mathcal{T} triées dans l'ordre non croissant des d_i

```

1: pour  $i \leftarrow 1$  à  $|S|$  faire
2:   Continue  $\leftarrow true$ 
3:    $lst_2 \leftarrow +\infty$ ,  $eftMin \leftarrow +\infty$ ,  $j \leftarrow 1$ 
4:   tant que  $j \leq |S| \wedge Continue = true$  faire
5:     si  $(r_{S[j]} + p_{S[j]} > r_{S[i]}$  et  $i \neq j)$  alors
6:        $lst_2 \leftarrow \min(d_{S[j]}, lst_2) - p_{S[j]}$ 
7:        $eftMin \leftarrow \min(r_{S[j]} + p_{S[j]}, eftMin)$ 
8:       si  $(lst_2 < r_{S[i]} + p_{S[i]})$  alors
9:          $LB(r_{S[i]}) \leftarrow eftMin$ 
10:        Continue  $\leftarrow false$ 
11:      fin si
12:    fin si
13:     $j \leftarrow j + 1$ 
14:  fin tant que
15: fin pour

```

– L'ajustement trouvé n'est pas appliqué au sein de l'algorithme pour ne pas risquer de changer l'ordre de S .

– La recherche d'un ajustement NP pour une tâche $S[i]$ s'arrête au premier (et non pas au meilleur) ajustement trouvé (voir ci-dessous la comparaison avec l'algorithme de Baptiste et Le Pape pour la justification).

Pour chaque tâche $S[i]$, ce très simple algorithme, d'une complexité temporelle $O(n^2)$ et spatiale $O(n)$, sélectionne les tâches candidates à l'ajustement de la date de début au plus tôt des tâches $S[i]$, maintient la borne inférieure lst_2 et stoppe aussitôt qu'un ajustement est possible pour la tâche courante $S[i]$.

La procédure doit être exécutée successivement jusqu'à ce qu'il n'y ait plus d'ajustements possibles. Ainsi, l'algorithme NP et son dual ND doivent être inclus dans une boucle et exécutés jusqu'à stabilisation des fenêtres. La procédure globale NP/ND peut être exécutée au plus $D \times n$ fois où $D = \max_{i=1, \dots, n} (d_i - r_i - p_i)$ est la taille du plus grand domaine. Même si dans la pratique NP/ND est exécutée en moyenne 2 ou 3 fois, l'obtention de tous les ajustements NP/ND se fait en Dn^3 étapes dans le pire des cas.

E.4 Comparaison avec l'algorithme Baptiste-Le Pape

A notre connaissance, un seul algorithme dédié à la propagation de la règle (3) (et sa duale) déduit tous les ajustements possibles. Il a été proposé dans [BL96]. Comme l'algorithme 7, il consiste en deux parties duales : NP et ND. La partie NP (resp. ND) pré-calculé, pour chaque tâche i , des bornes temporelles $\delta_{j,i}$ identifiant la tâche j qui permet le meilleur ajustement possible de r_i (resp. d_j) en temps quadratique. Néanmoins, plusieurs exécutions sont nécessaires pour effectuer tous les ajustements NP/ND. En effet,

l'effort de recherche du meilleur ajustement possible sur une passe est-il nécessaire puisque tout ajustement ultérieur par NP ou ND remet en cause la complétude du processus de déduction ? Plus de deux exécutions successives des algorithmes NP et ND sont nécessaires pour atteindre la stabilisation des fenêtres, *i.e.*, le point fixe. Ainsi, tous les ajustements NP/ND sont aussi obtenus en $O(Dn^3)$ et la différence de comportement des algorithmes est décidée par le temps nécessaire pour une itération NP/ND et la vitesse de convergence vers le point fixe (nombre moyen d'itérations pour l'atteindre).

Ce processus itératif nécessaire pour atteindre le point fixe est pris en compte par l'algorithme 7. La recherche du meilleur ajustement possible pour une tâche sur une passe est abandonnée au profit du premier ajustement trouvé. Ainsi, la phase de pré-calcul des bornes temporelles de meilleure actualisation de Baptiste-Le Pape est délaissée au profit d'une approximation plus simple. La condition suffisante d'arrêt énoncée en (24) assure la complétude du processus de déduction.

D'une complexité similaire, notre algorithme a une formulation plus simple. On montre de plus que sa rapidité de convergence vers le point fixe est supérieure à celle de l'algorithme de Baptiste et Le Pape [TL00a].

★

IV – Copie de quelques publications

1. Cooperative system design in scheduling, *International Journal of Production Research*, 36(1), 211–230, 1998
2. On not-first/not-last conditions in disjunctive scheduling, *European Journal of Operational Research*, 127(2), 332–343, 2000
3. Overview and possible extensions of shaving techniques for job-shop problems, *2nd International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'2000)*, Paderborn (Allemagne), 8-10 Mars 2000, pp.181-186
4. Dynamic task sequencing in temporal problems with uncertainty, *AIPS 2002 Workshop on "On-Line Planning and Scheduling"*, Toulouse (France), 24 Avril 2002, pp.41-48
5. Modeling and managing disjunctions in scheduling problems, *Journal of Intelligent Manufacturing*, 6, 133–144, 1995
6. Mixed task scheduling and resource allocation problems, *2nd International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimization Problems (CP-AI-OR'2000)*, Paderborn (Allemagne), 8-10 Mars 2000, pp.71-79
7. Décomposition temporelle et caractérisation de solutions admissibles pour le problème d'ordonnancement à une machine, *RAIRO Recherche Opérationnelle*, 33(2), 185–208, 1999
8. Lattice of task intervals : a support for edge-finding in disjunctive scheduling, Rapport LAAS 99224, version courte dans *International Conference on Industrial Engineering and Production Management (IEPM'99)*, Glasgow (GB), 12-15 Juillet 1999, Vol.2, pp.158-166
9. Constraint programming *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)*, 35(2), 5–36, 1995