

F. BENDALI

A. QUILLIOT

Représentation de familles ordonnées d'intervalles et applications

RAIRO. Recherche opérationnelle, tome 31, n° 1 (1997),
p. 73-101

http://www.numdam.org/item?id=RO_1997__31_1_73_0

© AFCET, 1997, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

REPRÉSENTATION DE FAMILLES ORDONNÉES D'INTERVALLES ET APPLICATIONS (*)

by F. BENDALI and A. QUILLIOT

Communiqué par Philippe CHRÉTIENNE

Résumé. – Nous traitons d'abord le problème consistant à placer sur une droite les sommets d'un hypergraphe partiellement ordonné de façon à minimiser le nombre de ruptures dans la représentation des arêtes de cet hypergraphe.

Nous présentons deux applications : l'une concerne la programmation linéaire entière et l'autre un problème d'ordonnement.

Dans la deuxième partie de ce travail, nous caractérisons les graphes d'intervalles dont les arêtes sont orientées de façon à représenter des relations d'inclusion, de recouvrement et d'antériorité.

Abstract. – We first deal with the problem which consists in finding a linear ordering of the vertices of a partially ordered hypergraph H in a way which minimizes the number of "breaks" in the edges of H . We present two applications of this problem: the first one is related with scheduling and the second one has to do with integer linear programming.

In the second part of this paper, we characterize interval graphs whose edges are oriented in order to represent inclusion, overlapping and anteriority relationships.

1. INTRODUCTION

Un hypergraphe d'intervalles est un hypergraphe $H = (X, E)$ dont les sommets peuvent être ordonnés de façon linéaire et de telle sorte que les sommets de chaque arête apparaissent comme consécutifs.

Ce type de structure intervient dans la gestion de problèmes de stockage d'informations (voir Lucio et Preparata [14], Ghosh [9]), d'ordonnement (voir Baptiste et Favrel [2]), d'archéologie (voir Kindall [13]). Les

(*) Reçu en février 1994.

(¹) ISIMA, BP 125, Campus des Cézeaux, 63173 Aubière, France.

hypergraphes d'intervalles ont été caractérisés de multiples façons, par sous-hypergraphes interdits (*voir* Duchet [7]), ou de manière algorithmique (*voir* Booth et Lueker [6]). Leur reconnaissance est polynomiale (*voir* [6]).

Au cours de la première partie, (paragraphes II, III, IV), de ce travail, nous nous intéressons à la caractérisation des hypergraphes ordonnés $H = (X, E, <)$, c'est-à-dire des hypergraphes dont les sommets sont munis d'une relation d'ordre partiel, pour lesquels la relation d'ordre linéaire permettant une représentation de l'hypergraphe simple $H = (X, E)$ comme hypergraphe d'intervalle peut être choisie compatible avec l'ordre partiel $<$. Notre motivation (paragraphe II) pour cette extension du concept d'hypergraphe d'intervalles provient de l'étude de problèmes d'ordonnancement pour lesquels les coûts sont liés à des coûts d'interruption du fonctionnement de certaines ressources. Les méthodes que nous introduisons, (paragraphe III), pour obtenir une caractérisation des hypergraphes ordonnés d'intervalles nous permettent de proposer des heuristiques simples et rapides, (paragraphe IV), pour le problème pratique qui consiste à compléter un hypergraphe ordonné en un hypergraphe linéairement ordonné de façon à créer le moins de ruptures possible sur la représentation des arêtes.

Au cours de la deuxième partie, (paragraphe V), de notre travail, nous reprenons le point de vue dual mis en évidence par Fulkerson et Gross [8], ou Benzer [3], et qui consiste à s'intéresser aux graphes qui sont des graphes d'intersection des arêtes d'un hypergraphe d'intervalles. Ces graphes, nommés graphes d'intervalles, caractérisés par configurations interdites par Lekkerkerker et Boland [5] ou de façon algorithmique par Golubic [11], permettent en fait de travailler sur des intervalles de la droite réelle et donc sur des systèmes de phases temporelles qui décrivent des actions effectuées sans interruption. Il arrive que l'on ait à gérer de tels systèmes de phases en robotique, au cours de processus de génération et validation de plans (*voir* Allen [1]). Les instants essentiels à l'intérieur de tels systèmes correspondent aux début et fin de phases et une relation d'ordre partiel sur ces instants se traduit au niveau des phases par des relations d'antériorité, d'inclusion ou de recouvrement. Nous caractérisons alors, de la même façon qu'au cours de la première partie, les configurations « parfaites », c'est-à-dire les graphes d'intervalles qui peuvent être représentés de façon à respecter de telles relations additionnelles. Le problème pratique qui se pose alors consiste non plus comme en première partie, à représenter les instants de façon à ce que les phases ou actions apparaissent le plus possible comme formées d'éléments consécutifs, mais cette fois-ci à représenter les phases comme

des intervalles de telle sorte que la relation d'adjacence du graphe utilisé représente le plus possible la relation d'intersection entre ces phases.

2. REPRÉSENTATION D'HYPERGRAPHES COMME FAMILLES D'INTERVALLES DISCRETS : DEUX EXEMPLES

2.1. Un problème d'ordonnement

Supposons un ensemble X de tâches, dont les durées $d(x)$, x dans X , sont connues à l'avance, soumises à des contraintes d'antériorité, et à effectuer par un même opérateur. Il peut se faire que chacune de ces tâches nécessite pour son exécution un certain nombre de ressources (machines), et qu'un coût relatif à chaque machine soit imputé au système. Ce coût peut correspondre soit au temps de location de la machine, supposé couvrir l'intervalle entre sa première et sa dernière utilisation, soit au nombre de fois où le fonctionnement de cette machine doit être interrompu (Set-up) (voir par exemple [2]).

Une modélisation de ce problème vient alors comme suit :

Au système (Tâches, Machines, Relation d'antériorité) correspond un triplet $(X, E, <)$, que nous nommerons *hypergraphe ordonné*, dans lequel :

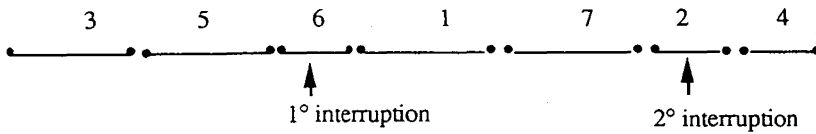
- X est l'ensemble des tâches ;
- Pour chaque machine m , $e(m)$ désigne l'ensemble des tâches nécessitant m ; E est donc l'ensemble de tous les $e(m)$, pour m parcourant l'ensemble des machines ;
- $<$ est la relation d'antériorité à laquelle sont soumis certains couples de tâches de X .

Un planning des tâches X se définit alors comme une permutation (ordre linéaire) σ sur X ; à un tel planning correspond pour chaque arête $e = e(m)$ dans E les quantités :

- $I(e, \sigma)$ = Nombre d'interruptions de l'activité de la machine m pour le planning σ ;
- $T(e, \sigma)$ = Temps écoulé entre le début de la première utilisation et la fin de la dernière utilisation de la machine m pour le planning σ .

Exemple: Si $X = \{1 \dots 7\}$, $\sigma = (3, 5, 6, 1, 7, 2, 4)$, $d(x) = 1$ si x est pair et $d(x) = 2$ si x est impair, $e = \{5, 1, 7, 4\}$ alors :

$$I(e, \sigma) = 2; T(e, \sigma) = 9;$$



Notre problème revient alors à déterminer σ de façon à minimiser une certaine somme pondérée (en fonction des importances relatives des tâches de X) des quantités $I(\sigma, x)$ ou $T(\sigma, x)$.

Nous noterons REP-HYP(INTERUPT) le problème défini par le premier de ces deux critères et REP-HYP(TEMPS) le problème défini par le deuxième critère.

2.2. Un problème de décomposition en programmation linéaire entière

Considérons à présent un programme linéaire entier de la forme :

$$(P) : \{ \text{Trouver } x \text{ dans } Z^n, \text{ tel que } x \geq 0, Ax = b \};$$

à l'intérieur duquel A est supposée être une matrice $n \cdot m$ à coefficients en $\{0, 1\}$ et b un vecteur entier positif.

On peut songer à écrire chaque vecteur ligne A^i , i dans $1 \dots m$, de la façon suivante :

$$A^i = \sum A_j^i, \text{ dans } 1 \dots k(i), \text{ où les } A_j^i \text{ sont formés de } 0 \\ \text{et d'un bloc de } 1 \text{ consécutifs}$$

et cela en s'efforçant de rendre les coefficients $k(i)$ les plus petits possibles ; on note alors A^* la matrice dont les lignes sont les A_j^i , i dans $1 \dots m$, j dans $1 \dots k(i)$.

Le programme P se réécrit alors :

(P1) :

$$\left\{ \begin{array}{l} \text{Trouver } x \text{ dans } Z^n, t = (t_{i,j}, i \text{ dans } 1 \dots m, j \text{ dans } 1 \dots k(i)) \\ \text{à coefficients entiers,} \end{array} \right.$$

tels que :

$$x, t \leq 0$$

Pour tout i dans $1 \dots m$, j and $1 \dots k(i)$, $A_j^i \cdot x = t_{i,j}$;

Pour tout i dans $1 \dots m$, $\sum_{j=1 \dots k(i)} t_{i,j} = b_i$. (E1)

Exemple :

Le programme :

$$\begin{aligned} x_1, x_2, x_3, x_4, x_5, x_6 &\geq 0 \\ x_1 + x_2 + x_5 + x_6 &= 7 \\ x_2 + x_3 + x_4 &= 8 \\ x_1 + x_2 + x_4 + x_5 + x_6 &= 10 \end{aligned}$$

se réécrit :

$$\begin{aligned} x_1 + x_2 &= t_1; \\ x_5 + x_6 &= 7 - t_1; \\ x_2 + x_3 + x_4 &= 8; \\ x_1 + x_2 &= t_2; \\ x_4 + x_5 + x_6 &= 10 - t_2; \\ x_1 \dots x_6 &\geq 0; t_1, t_2 \geq 0. \end{aligned}$$

L'intérêt de cette reformulation provient du fait que toute matrice en $\{0, 1\}$, à l'intérieur de laquelle les 1 figurant dans chaque ligne sont consécutifs, est totalement unimodulaire, et donc que tout programme linéaire entier dont la matrice de contraintes possède cette propriété peut se résoudre directement par l'algorithme du simplexe sans tenir compte de la contrainte d'intégrité (voir [8]). Pour t fixé, le système (P1), ci-dessus se situe dans un tel cadre. Un schéma de résolution de (P) pourra se concevoir alors comme suit :

```

Début
Fixer  $t$  entier,  $\geq 0$ , et satisfaisant (E1); Not Stop;
Tant que not Stop faire
  Début
  Résoudre (P1) à  $t$  fixé;
  Si Succès alors Stop sinon
    Début
    Trouver  $y$  (dualité) =  $(y_{i,j}, i \text{ dans } 1 \dots m, j \text{ dans } 1 \dots k(i))$  tel que :  $y \cdot A^* \geq 0$ 
    et  $y \cdot t < 0$ ;
    Insérer dans (E1) la contrainte  $y \cdot t \geq 0$ ;
    Déterminer  $t \geq 0$ , entier et satisfaisant (E1);
    Si Echec alors Stop;
    Fin
  Fin
Fin.
  
```

Afin qu'un tel processus ait des chances d'être opérationnel, il importe que le nombre des variables de contrôle $t_{i,j}$, sur lesquelles ont été transférées la complexité due à la contrainte d'intégrité, soit le plus petit possible (et petit en comparaison avec le nombre n des coordonnées de x). Cela signifie qu'une opération préalable à la mise en œuvre d'un tel processus consistera à renuméroter (permuter) les colonnes de A de façon à minimiser le nombre d'interruptions des séquences de 1 qui figurent sur chaque ligne. Ce problème est exactement le problème introduit au paragraphe précédent avec :

- X = Ensemble des colonnes de la matrice A ;
- E = Ensemble des lignes de A ;
- $<$ est la relation d'ordre vide ;
- Le critère de performance d'une permutation σ sur X est associé à la mesure $I(\sigma, x)$ du nombre des interruptions et à des poids sur les sommets de X tous égaux à 1.

3. LES HYPERGRAPHES ORDONNÉS INTERVALLE-REPRÉSENTABLES : CARACTÉRISATION

Nous venons de voir deux problèmes, assez différents au départ, et qui se ramènent tous deux à permuter les sommets d'un hypergraphe (ordonné), de façon à rendre les éléments de chaque arête les plus « consécutifs » possible.

Nous pouvons observer que le problème noté REP-HYP(INTERRUPT) est NP-Complet : il contient en effet le problème de l'existence dans un graphe d'un chemin hamiltonien (il suffit pour s'en rendre compte de considérer le cas où chaque arête de notre hypergraphe est de cardinalité 2 et où l'on se pose la question de savoir si la somme $\sum_{e \text{ dans } E} I(\sigma, e)$ peut être rendue $\leq |E| - [X] + 1$).

Nous allons commencer par étudier le cas où cette somme $\sum_{e \text{ dans } E} I(\sigma, e)$ peut être rendue nulle. La permutation σ est alors solution optimale aussi bien de REP-HYP(INTERRUPT) que de REP-HYP(TEMPS). Une telle situation correspondra à la possibilité de permuter les sommets de l'hypergraphe partiellement ordonné $H = (X, E, <)$ d'une façon qui soit compatible avec la relation $<$ et qui fasse que chaque arête e dans E soit formée d'éléments consécutifs.

3.1. Rappel : hypergraphes intervalle-représentables [6, 7]

Dans le cas où $<$ est la relation vide, la réponse à ce problème est déjà connue. Les hypergraphes $H = (X, E)$ dont on peut permuter les sommets

de façon à faire apparaître chaque arête e dans E comme formée d'éléments consécutifs (formant un « intervalle ») sont dit « intervalles-représentables » et peuvent être identifiés comme suit :

THÉORÈME (Duchet [6]) : Soit $H = (X, E)$ un hypergraphe ; notons E^* la plus petite (au sens de l'inclusion) famille de parties de X contenant E et telle que si e_1 et e_2 sont dans E^* et d'intersection non vide alors l'intersection de e_1 et e_2 est aussi dans E^* .

On a alors :

H est intervalle-représentable

ssi

E^* est sans triangle, c'est-à-dire qu'il n'existe pas 3 arêtes dans E^* , qui sont 2 à 2 d'intersection non vide et qui sont d'intersection globale vide.

La reconnaissance de ces hypergraphes intervalle-représentables peut se faire en temps polynomial (voir [6]).

Soit à présent un hypergraphe $H = (X, E, <)$ dont les sommets sont partiellement ordonnés (nous parler d'hypergraphe ordonné). Par analogie avec ce qui précède, nous dirons que H est intervalle-représentable si la permutation (ordre linéaire) σ qui fait apparaître chaque arête de H comme formée d'éléments consécutifs peut être choisie de façon à respecter la relation d'ordre $<$. La permutation σ est alors nommée intervalle-représentation de H .

Pour un tel hypergraphe $H = (X, E, <)$ nous pouvons noter $<^*$ la plus petite (au sens de l'inclusion) relation contenant $<$ et telle que :

Pour tout e dans E , x, y dans e , z hors de e , on a les implications :

$$\begin{aligned} x <^* z &\Rightarrow y <^* z ; \\ z <^* x &\Rightarrow z <^* y . \end{aligned}$$

On obtient alors :

THÉORÈME 1: L'hypergraphe ordonné $H = (X, E, <)$ est intervalle-représentable si et seulement si (notations ci-dessus) :

- E^* est sans triangle ;
- La relation $<^*$ est sans circuit.

Schéma général de la démonstration : Le côté « seulement si » de l'équivalence ci-dessus est trivial ; démontrer la réciproque signifie partir d'une structure $H = (X, E, <)$ telle que E^* est sans triangle et $<^*$ est sans circuit et parvenir à construire sur X une relation d'ordre linéaire σ ,

compatible avec $<$ et telle que pour toute arête e de H , les sommets de e soient consécutifs pour σ . Nous procéderons pour cela en considérant un couple (A, σ) , où A est un sous-ensemble de X et σ un ordre linéaire sur A , compatible avec $<$ et tel que pour tout e dans E^* , les sommets de e soient consécutifs pour σ , et en prouvant que si A est choisi de cardinalité maximale, alors il est égal à X .

Démonstration du théorème 1 : La partie « seulement si » de l'équivalence ci-dessus étant triviale, nous partons donc d'une structure $H = (X, E, <)$ telle que E^* soit sans triangle et $<^*$ sans circuit, et nous cherchons à prouver que X peut être linéairement ordonné de façon compatible avec $<$ et de telle sorte que les sommets d'une même arête de H apparaissent comme consécutifs pour cet ordre linéaire.

Pour cela nous remarquons :

- qu'il est possible, sans changer le fond du problème, de rajouter à X deux éléments DÉBUT et FIN qui respectivement précèdent et succèdent chaque élément de X pour la relation $<$ et qui ne sont dans aucune arête ;
- qu'il est possible de supposer que $<$ coïncide avec $<^*$;
- qu'il est possible de supposer $E = E^*$. Il suffit en effet de constater pour cela que si e et f sont 2 arêtes de H , d'intersection non vide, alors le rajout à E de $e \cup f$ ne modifie pas $<^*$.

Considérons à présent un couple (A, σ) où :

- A est un sous-ensemble de X ;
- σ est une relation d'ordre linéaire sur A , compatible avec $<$, et telle que pour toute arête e de E , les éléments de $A \cap e$ sont consécutifs pour σ ;
- La cardinalité k de A est la plus grande possible avec ces propriétés.

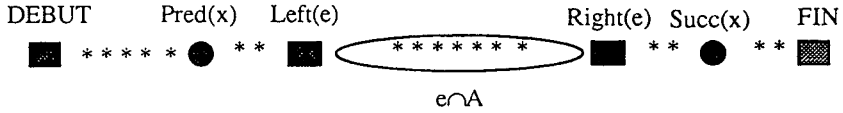
Si $k = n$ alors nous avons fini, sinon nous considérons x dans $X - A$ et nous cherchons à prolonger σ sur $A + \{x\}$.

Pour cela :

- Nous notons $\text{Pred}(x)$ et $\text{Succ}(x)$, respectivement le sommet de A le plus grand (pour σ) qui est inférieur à x (pour $<$) et le sommet de A le plus petit (pour σ) qui est supérieur à x (pour $<$).
- Pour chaque arête e de H qui contient x , nous notons $\text{Left}(e)$ le plus grand (pour σ) élément de A qui est plus petit (pour σ) que tous les éléments de $A \cap e$; si $A \cap e$ est vide, nous posons $\text{Left}(e) = \text{DÉBUT}$;

Nous notons de même $\text{Right}(e)$ le plus petit élément de A (pour σ) qui est plus grand (pour σ) que chaque élément de $A \cap e$; si $A \cap e$ est vide alors nous posons $\text{Right}(e) = \text{FIN}$;

La signification de $\text{Left}(e)$ et $\text{Right}(e)$ est que prolonger σ sur $A + \{x\}$ doit se faire de telle sorte que $\text{Left}(e) \sigma \times \sigma \text{Right}(e)$.



Puisque $E = E^*$ et que les éléments de l'intersection de toute arête de H avec A sont consécutifs pour σ , nous voyons que :

Pour tout e, e' dans E , $\text{Left}(e) \sigma \text{Right}(e')$;

De même, puisque $<$ coïncide avec $<^*$, nous voyons que pour tout e dans E :

$\text{Pred}(x) \sigma \text{Right}(e)$

et

$\text{Left}(e) \sigma \text{Succ}(x)$.

Notons alors L l'élément de A le plus grand (pour σ) parmi les éléments $\text{Pred}(x)$ et $\text{Left}(e)$, e dans E , et R l'élément de A le plus petit (pour σ) parmi les éléments $\text{Succ}(x)$ et $\text{Right}(e)$, e dans E .

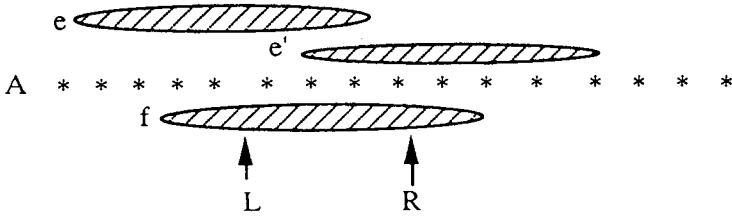
Nous avons alors $L \sigma R$ et nous allons chercher à insérer x quelque part entre L et R .

Raisonnons par l'absurde et supposons que cela ne soit pas possible. Si nous notons $L = a_0, \dots, a_{p+1} = R$, la suite des éléments de A situés entre L et R (pour σ), alors nous voyons que cela signifie pour tout i dans $0 \dots p$, l'existence d'une arête f_i de H qui contient a_i et a_{i+1} et ne contient pas x .

Notons f l'union de toutes les arêtes $f_i, i = 0 \dots p$; puisque $E = E^*$, f est une arête de E . Trois cas peuvent alors se présenter :

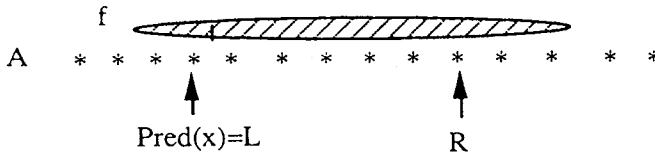
1° cas : R et L s'écrivent respectivement $\text{Right}(e)$ et $\text{Left}(e')$, avec e et e' dans E .

Nous obtenons alors une contradiction en notant que e, e' et f forment un triangle, s'appuyant sur les sommets L, R et x .



2° cas : $L = \text{Pred}(x)$;

Nous obtenons alors le résultat en notant que $\text{Pred}(x)$ étant dans f ainsi que R , on devrait, puisque $<$ et $<^*$ coïncident, avoir en fait $x > R$, ce qui induit une contradiction sur la définition de $\text{Pred}(x)$.



3° cas : $R = \text{Succ}(x)$; nous concluons selon le même processus que pour le 2° cas.

Dans tous les cas, nous obtenons une contradiction, ce qui conclut la démonstration. FIN.

La démonstration ci-dessus est constructive. Le lecteur pourra vérifier qu'elle s'interprète de façon algorithmique, jusqu'à permettre la reconnaissance en $O(|X| \cdot |E|)$ d'un hypergraphe ordonné intervalle-représentable $H = (X, E, <)$.

4. HEURISTIQUES POUR LES PROBLÈMES REP-HYP(INTERRUPT) ET REP-HYP(TEMPS)

Nous allons à présent montrer comment les résultats théoriques précédents peuvent s'interpréter de façon à induire un traitement pour les deux problèmes REP-HYP(INTERRUPT) et REP-HYP(TEMPS) décrits en introduction. La première de ces heuristiques, conçue pour traiter le problème REP-HYP(INTERRUPT) de la recherche d'une permutation σ sur les sommets de l'hypergraphe de départ qui minimise le nombre de ruptures dans la représentation des arêtes, est gloutonne et directement conçue comme un prolongement de la démonstration du théorème 1 du paragraphe précédent. La seconde traite du problème REP-HYP(TEMPS) qui suppose que l'on

cherche à minimiser l'ampleur des intervalles discrets définis par σ et recouvrant complètement les arêtes de H . Elle repose sur une approche par Branch/Bound et présente la particularité d'inclure un mécanisme de filtrage des sommets de l'arbre d'exploration qui permet de ne générer que les permutations satisfaisant un certain critère d'optimalité locale.

4.1. Une heuristique gloutonne pour le problème REP-HYP(INTERRUPT)

Rappelons l'énoncé du problème REP-HYP(INTERRUPT). Un hypergraphe ordonné $H = (X, E, <)$ étant donné, dont les arêtes sont munies de poids k_e , e dans E , on souhaite trouver une permutation σ , compatible avec la relation $<$, et minimisant la quantité

$$\text{BREAK}(H, \sigma) = \sum_{e \text{ dans } E} k_e \cdot I(\sigma, e)$$

où $I(\sigma, e)$ représente le nombre d'interruptions de e selon σ , c'est-à-dire le nombre de couples x, y dans X tels que :

- x et y sont dans e , x est avant y selon σ ;
- les éléments entre x et y selon l'ordre σ sont tous hors de e et définissent un ensemble non vide ;

La procédure que nous allons présenter ici s'appuie sur le fait qu'une permutation (ordre linéaire) de l'ensemble X peut être aussi considérée comme un plongement des éléments de X sur la droite réelle D . Afin d'en fournir une description complète, nous avons besoin de quelques notations :

Soient $H = (X, E, <)$ notre hypergraphe ordonné, A un sous-ensemble de X et σ une permutation de A qui est compatible avec $<$, (une telle situation sera celle induite à chaque entrée dans la boucle principale de notre algorithme). Nous pouvons calculer la quantité $\text{BREAK}(H_A, \sigma)$, où H_A est le sous-hypergraphe ordonné induit par A .

Si x_0 est dans $X - A$, si x et y dans A sont consécutifs pour σ , et si pour tout z dans A tel que $z < (>) x_0$ on a que z est avant x pour σ ou égal à x (est après y pour σ ou égal à y), alors on peut envisager de prolonger σ à $A + \{x_0\}$ en insérant x_0 entre x et y . Notant σ^* la permutation ainsi définie sur $A + \{x_0\}$, nous pouvons poser :

$$\text{GAP}(H, A, \sigma, x_0, x, y) = \text{BREAK}(H_{A \cup \{x_0\}}, \sigma^*) - \text{BREAK}(H_A, \sigma).$$

Explication : La quantité $\text{GAP}(H, A, \sigma, x_0, x, y)$ mesure la perte de performance induite par l'insertion de x_0 entre x et y au cours de la

construction de σ . La procédure PLONGE va alors travailler de façon gloutonne en construisant la permutation σ sommet par sommet, et en réalisant chaque insertion d'un nouveau sommet dans σ de façon à minimiser la quantité GAP.

Algorithme PLONGE

Input : L'hypergraphe ordonné $H = (X, E, <)$ et les poids k_e, e dans E ;

Output : Une permutation σ de X compatible avec $<$ et fournissant une solution approchée du problème REP-HYP(INTERRUPT);

Début

$\sigma :=$ Indéfini; $A :=$ Nil;

Tant que $A \neq X$ faire

 Début

 Choisir x_0 dans $X - A$; (*)

 Trouver x, y consécutifs dans A pour σ , tels que l'insertion de x_0 entre x et y est réalisable et tels que la quantité $\text{GAP}(H, A, \sigma, x_0, x, y)$ soit minimale; (**)

$\sigma :=$ Permutation sur $A + \{x_0\}$ obtenue à partir de σ par insertion de x_0 entre x et y ;

$A := A + \{x_0\}$;

 Fin

Fin.

Remarque : Il découle du théorème de Dushnik et Miller relatifs aux plongements d'ensembles ordonnés sur la droite réelle que l'instruction (**) est toujours exécutable et donc que l'algorithme s'achève bien avec la construction d'une permutation de X compatible avec $<$. Par ailleurs on voit aisément que dans le cas où E est stable par union connexe ($E = E^*$ au sens du théorème 1), alors PLONGE reconnaît le fait que l'hypergraphe ordonné $H = (X, E, <)$ soit intervalle-représentable.

Performances : PLONGE peut être implémenté de façon à tourner en $O(|X|^2 \cdot |E|)$. Pratiquement il ne lui faut pas plus d'une minute pour travailler sur un hypergraphe ordonné comportant n sommets et m arêtes, n, m étant tels que $10 \leq m \leq n \leq 1000$, sur une station SUN4. Les résultats qu'il fournit dépendent de façon substantielle de la manière dont l'instruction (*) est spécifiée. Nous avons obtenu nos meilleurs résultats en reprenant les principes de la programmation par contraintes et en adoptant le principe du "First Fail", c'est-à-dire en choisissant x_0 comme étant le sommet de plus haut degré dans l'hypergraphe défini par les arêtes de H dont l'intersection avec A est non vide. Cette stratégie fait alors apparaître un écart par rapport à l'optimum théorique (calculé par Branch and Bound pour des exemples mettant en jeu entre 20 et 40 sommets) d'environ 8 %,

les résultats les moins bons apparaissent bien sûr quand la relation $<$ est vide (environ 11 %).

4.2. Une méthode par Branch/Bound pour le problème REP-HYP(TEMPS)

Rappelons l'énoncé du problème ; un hypergraphe $H = (X, E, <)$ étant donné, ainsi que des poids k_e , e dans E , sur les arêtes de E , et des durées d_x , x dans X , on cherche une permutation σ de X qui minimise la quantité :

$$T^*(\sigma) = \sum_{e \text{ dans } E} k_e \cdot T(\sigma, e)$$

où $T(\sigma, e) = \sum_{x \text{ dans } cl(\sigma, e) = \{x \text{ tels que } First(e, \sigma) \leq_\sigma x \leq_\sigma Last(e, \sigma)\}} d_x$.

$First(e, \sigma)$, $Last(e, \sigma)$ et \leq_σ étant eux-mêmes définis par :

- \leq_σ = ordre linéaire défini par σ sur X ;
- $First(e, \sigma)$ = le premier (plus petit) élément de e pour \leq_σ ;
- $Last(e, \sigma)$ = le dernier (plus grand) élément de e pour \leq_σ .

Explication heuristique : Si l'on pense aux objets de X comme à des actions à exécuter et aux objets de E comme à des ressources, $cl(e, \sigma)$ (cl comme « clôture ») apparaît comme l'ensemble des actions planifiées selon σ entre la première ($First(e, \sigma)$) et la dernière ($Last(e, \sigma)$) utilisation de la ressource e . $T(\sigma, e)$ est alors le temps requis pour l'exécution des actions de $cl(e, \sigma)$.

Une démarche naturelle consiste à explorer l'arbre dont les sommets sont définis par les débuts d'ordonnancement de X qui sont compatibles avec la relation $<$.

Formellement un tel ordonnancement partiel σ^* sera défini comme une suite d'éléments de X , à l'intérieur de laquelle chaque élément de X figure au plus une fois, et telle que si x et y figurent dans cette suite et sont tels que $x < y$, alors x est avant y dans σ^* .

Une borne inférieure de $T(\sigma)$, pour σ solution admissible de REP-HYP(TEMPS) et admettant σ^* comme début de liste, sera alors donnée par la formule :

$$EVAL(\sigma^*) = \sum_{e \text{ dans } E} k_e \star \sum_{x \text{ dans } X - \sigma^*} + (\sum_{x \text{ dans } \sigma^* \text{ et } \geq_\sigma First(e, \sigma^*)} d_x).$$

Nous allons montrer à présent qu'il est possible d'améliorer l'efficacité du filtrage induit par le recours à l'évaluation EVAL en ne générant

que des plannings σ optimums par rapport au critère (TRANSLATE) qui exprime qu'une permutation σ est améliorable si il existe une arête et un sommet x_0 placé de telle sorte qu'il « brise » e , tels que le déplacement de x_0 immédiatement à droite (ou à gauche) de e ne brise aucune arête. Ce critère se formalise comme suit :

Description du critère (TRANSLATE) : Supposons qu'une permutation σ de X , compatible avec $<$, soit telle qu'il existe e dans E , x_0 dans $X - e$, satisfaisant $\text{First}(e, \sigma) < x_0 < \text{Last}(e, \sigma)$ et permettant à une des deux suites d'assertions AM-GAUCHE ou AM-DROITE ci-dessous d'être vraie :

AM-GAUCHE :

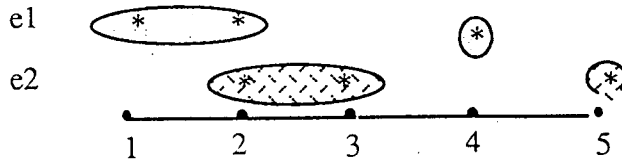
- pour tout f dans E qui contient x_0 on a que $\text{First}(e, \sigma)$ est dans $cl(f, \sigma)$;
- pour tout f dans E tel que $cl(f, \sigma)$ ne contient pas x_0 , on a que $\text{Pred}(\text{First}(e, \sigma))$ et $\text{First}(e, \sigma)$ ne sont pas tous deux dans $cl(f, \sigma)$;
- il n'existe pas x dans X tel que $\text{First}(e, \sigma) \leq x \leq x_0$;

AM-DROITE : pour tout f dans E qui contient x_0 , on a que $\text{Last}(e, \sigma)$ est dans $cl(f, \sigma)$; pour tout f dans E tel que $cl(f, \sigma)$ ne contient pas x_0 , on a que $\text{Succ}(\text{Last}(e, \sigma))$ et $\text{Last}(e, \sigma)$ ne sont pas tous deux dans $cl(f, \sigma)$; il n'existe pas x dans X tel que $x_0 < x \leq \text{Last}(e, \sigma)$.

L'hypothèse AM-GAUCHE permet d'améliorer σ , au sens du problème REP-HYP-(TEMPS), en déplaçant x_0 entre $\text{First}(e, \sigma)$ et le prédécesseur de ce dernier pour σ ; l'hypothèse AM-DROITE permettra une amélioration similaire.

Remarque : Le critère TRANSLATE fournit des conditions d'amélioration de σ qui sont *seulement suffisantes*. En effet, la transformation induite implique l'amélioration de la représentation d'une arête particulière de l'hypergraphe, sans que les représentations des autres arêtes puissent se trouver dégradées. On pourrait bien sûr imaginer de pouvoir améliorer globalement σ tout en dégradant la représentation de certaines arêtes.

Exemple : Si $\sigma = \{1, 2, 3, 4, 5\}$, $e_1 = \{1, 2, 4\}$, $e_2 = \{2, 3, 5\}$, on voit que l'on peut améliorer σ en utilisant $x_0 = 3$ et $e = e_1$ et en plaçant 3 immédiatement à la droite de 4 (assertion AM-DROITE) :



Nous dirons que σ satisfait le *critère d'optimalité partielle (TRANSLATE)* si aucune des deux configurations d'amélioration locale AM-GAUCHE et AM-DROITE ne peut être appliquée à σ .

Nous allons montrer comment il est possible, au cours d'un procédé de résolution du problème REP-HYP-(TEMPS) par séparation et évaluation du type résumé plus haut, de ne générer que des planning satisfaisant le critère (TRANSLATE).

Supposons que nous ayons formé un début de planning σ^* conformément à ce qui a été exposé au début de ce paragraphe, et construisons les trois sous-ensembles R-DROITE, R-GAUCHE, R-ORDRE de X suivant :

- R-DROITE = $\{x$ hors de σ^* tels qu'il existe e dans E , contenant x et inclus dans $\sigma^* + \{x\}$, et y dans $\sigma^* - e$ vérifiant :

$$\text{First}(e, \sigma^*) < y;$$

ainsi que les implications :

- Si f est dans E et y est dans f alors f n'est pas inclus dans la liste σ^* ;
- Si f est dans E , ne contient pas y et contient des éléments hors de σ^* et des éléments dans σ , alors f contient des éléments situés avant y dans σ^* ;
- Si z est dans X et $z > y$ alors z n'est pas dans la liste σ^* .

Exemple : Si l'on se réfère à l'exemple précédent et à la figure associée et que l'on suppose que $\sigma^* = \{1, 2, 3\}$, on voit que R-DROITE = $\{4\}$; pour $x = 4$, la définition de R-DROITE s'applique en faisant $e = e1$ et $y = 3$.

- R-GAUCHE = $\{x$ hors de σ^* tels qu'il existe e dans E , ne contenant pas x , d'intersections non vides avec σ^* et $X - \sigma^*$ et satisfaisant les implications :

- Si f est dans E et x dans f , alors $\text{First}(e, \sigma^*) \geq \text{First}(f, \sigma^*)$;
- Si f est dans E et contenue dans la liste σ^* , alors f ne contient pas simultanément $\text{First}(e, \sigma^*)$ et $\text{Pred}(\text{First}(e, \sigma^*))$;
- Si y est dans σ^* et $y < x$ alors $y < \text{First}(e, \sigma^*)$;

Exemple : Si l'on reprend à nouveau l'exemple précédent, on voit que si l'on suppose $\sigma^* = \{1, 2, 3\}$, alors R-GAUCHE = $\{4\}$. Pour $x = 4$, la définition de R-GAUCHE s'applique en faisant $e = e2$.

– R-ORDRE = $\{x \text{ hors de } \sigma^* \text{ tels qu'il existe } y \text{ hors de } \sigma^* \text{ et satisfaisant } y < x\}$.

Alors :

THÉORÈME 3: *On génère les permutations optimales pour le critère (TRANSLATE) en procédant de la façon suivante :*

Procédure GEN-TRANSLATE :

Début

Not Stop; $\sigma^* := \text{Nil}$;

Tant que Not Stop faire

 Début

 Former R-GAUCHE, R-DROITE, R-ORDRE;

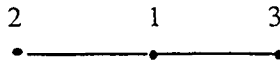
 Choisir x dans $X - \sigma^* - \text{R-GAUCHE} - \text{R-DROITE} - \text{R-ORDRE}$;

 Si x n'existe pas alors Stop sinon prolonger σ^* avec x ;

 Fin;

Fin.

Exemple : Si $X = \{1 \dots 5\}$, $\sigma^* = (2, 1, 3)$, $E = \{(1, 3, 5), (2, 1, 5), (4, 3)\}$



alors $x_0 = 4$ ne pourra être choisi pour prolonger σ^* du fait de l'appartenance de 4 à R-GAUCHE (on fait $e = (1, 3, 5)$).

Démonstration : Supposons qu'une permutation σ de X ait été obtenue selon la procédure ci-dessus; si σ ne respecte pas la relation $<$, c'est qu'un certain x était dans R-ORDRE au moment où il a été incorporé à la liste partielle σ^* au cours de l'exécution de GEN-TRANSLATE.

Si x_0 dans X , e dans E permettent de rendre vraie l'assertion (AM-GAUCHE), c'est que x_0 était dans R-GAUCHE au moment de son incorporation dans σ au cours du processus GEN-TRANSLATE.

Si x_0 dans X , e dans E permettent de rendre vraie l'assertion (AM-DROITE), c'est que x_0 était dans R-DROITE au moment de son incorporation dans σ au cours de l'exécution de GEN-TRANSLATE. Réciproquement, si σ satisfait le critère local (TRANSLATE), il suffit de vérifier que pour chaque x_0 dans X , la sous-liste σ^* formée des éléments

de σ situés avant x_0 est telle que (avec les notations précédentes), x_0 n'est ni dans R-GAUCHE, ni dans R-DROITE, ni dans R-ORDRE (vérification laissée au lecteur). FIN.

V. REPRÉSENTATION DE STRUCTURES D'INTERVALLES

Comme nous l'avons expliqué en introduction, nous adoptons à présent un point de vue dual, qui consiste à exploiter le fait que les graphes d'intersections de familles d'intervalles fermés de la droite réelle sont exactement les graphes d'intersection de familles d'arêtes définissant des hypergraphes d'intervalles. Si l'on se réfère à la sémantique associée aux problèmes d'ordonnancements, on peut dire que nous nous intéressons dès lors non plus aux instants, mais aux phases, et que ce que nous avons en tête en terme d'applications est non plus d'ordonner les instants de façon à faire apparaître le plus possible les phases comme des intervalles (familles d'instant consécutifs), mais de placer des phases intervalles dans l'espace-temps de façon à ce que les relations que l'on impose sur elles coïncident le plus possible avec les relations de positionnement relatifs définies par les intervalles de l'espace-temps. Ce point de vue est celui qui prévaut généralement en robotique quand il est question de génération et validation de plans [1], un plan se construisant généralement à partir d'intentions, que l'on spécifie par des actions élémentaires et des contraintes de positionnement relatifs entre ces actions, et que l'on valide en cherchant à placer les actions ainsi contraintes dans l'espace-temps continu. Nous considérerons donc au cours de ce paragraphe les graphes définis par des familles d'intervalles (de la droite discrète ou continue) reliés entre eux par différents types de relation (inclusion, recouvrement, antériorité, disjonction) et nous les caractériserons, puis nous poserons, sans le résoudre, le problème de représentation approchée associé.

Soit donc un graphe simple $G = (X, E)$; nous notons $G^c = (X, E^c)$ son complémentaire et $A(E)$ l'ensemble des arcs dont le support est dans E . Nous notons $[a, b]^*$ un arc d'origine a et d'extrémité b tel que $[a, b]$ est dans E . Si $u = [a, b]^*$ est dans $A(E)$, nous posons $u^{-1} = [b, a]^*$.

Si A est inclus dans $A(E)$, nous disons que le support de A est l'ensemble des arêtes de E qui peuvent être considérées comme le support d'un arc de $A(E)$.

Rappels : Graphes d'intervalle, graphes d'ordre, relations de forçage

Un graphe simple $G = (X, E)$ est dit d'ordre si ses arcs peuvent être orientés de façon à définir une relation d'ordre (transitive et antisymétrique).

Un graphe $G = (X, E)$ est dit d'intervalles si il peut être considéré comme le graphe d'intersection d'une famille d'intervalles bornés de la droite réelle.

La caractérisation de ces graphes d'ordre et d'intervalles s'effectue à l'aide de la relation suivante, dite de forçage :

– Sur $A(E)$, nous définissons la relation $R(E)_0$ par : $[a, b]^* R_0 [a, c]^*$ et $[b, a]^* R_0 [c, a]^*$ ssi $[b, c]$ n'est pas dans E ;

– Nous notons $R(E)$ la fermeture transitive de $R(E)_0$; $R(E)$ est alors une relation d'équivalence sur $A(E)$, et pour tout u dans $A(E)$ nous notons $\text{Eq}(R(E), u)$ la classe d'équivalence de u pour $R(E)$;

Nous pouvons alors énoncer :

THÉORÈME (Golumbic [9], Pnueli [10]) : *Un graphe $G = (X, E)$ est un graphe d'ordre si et seulement si pour tout u dans $A(E)$ on a : $\text{Eq}(R(E), u) = \text{Eq}(R(E), u^{-1})$.*

THÉORÈME (Gilmore et Hoffman [8]) : *Un graphe $G = (X, E)$ est un graphe d'intervalles si et seulement si il ne possède pas de cycle sans corde de longueur 4 et si son complémentaire est un graphe d'ordre.*

Nous nous proposons ici d'étendre ces résultats en considérant une structure $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$ formée d'un graphe simple $G = (X, E)$, de deux familles Ov (Ov comme « Overlap » et Inc (Inc comme « Inclusion ») d'arcs de $A(E)$ et d'une famille Ant (Ant comme « Antériorité ») d'arcs de $A(E^c)$. Nous souhaitons caractériser ce qui doit satisfaire cette structure pour qu'il soit possible d'associer à chaque sommet x de X un intervalle borné fermé $r(x)$ de la droite réelle de telle sorte que l'on ait :

– $[x, y]$ est dans E si et seulement si $r(x)$ et $r(y)$ sont d'intersection non vide; (A1)

– Si $[x, y]^*$ est dans Ant alors $r(x)$ est situé avant $r(y)$ sur la droite réelle; (A2)

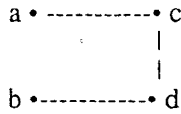
– Si $[x, y]^*$ est dans Ov alors on a :

$$\text{Début}(r(x)) \leq \text{Début}(r(y)) \leq \text{Fin}(r(x)) \leq \text{Fin}(r(y)); \text{ (A3)}$$

— Si $[x, y]^*$ est dans Inc alors $r(x)$ est inclus dans $r(y)$. (A4)

Si tel est le cas, nous dirons que la structure $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$ est *intervalle-représentable*.

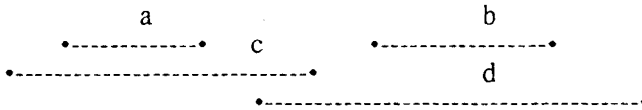
Exemple : à la structure



$$E = \{[a, c], [b, d], [c, d]\}; \quad \text{Ov} = \{[c, d]^*\};$$

$$\text{Ant} = \{[a, b]^*\}; \quad \text{Inc} = \{[a, c]^*; \{[b, d]^*\}$$

peut être associée la représentation :



Le résultat que nous allons prouver ici est le suivant : (Nil désigne ici l'ensemble vide).

THÉORÈME 4 : (a) : La structure $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$ ci-dessus est intervalle-représentable si et seulement si la structure $(X, E, \text{Ov}, \text{Nil}, \text{Ant})$ l'est aussi et si pour tout $[x, y]^*$ dans Inc et tout sommet z adjacent à x dans G , on a que z est aussi adjacent à y ;

(b) : La structure $(X, E, \text{Ov}, \text{Nil}, \text{Ant})$ est intervalle-représentable si et seulement si la structure $(X, E, \text{Nil}, \text{Nil}, \text{Co}(\text{Ant}, \text{Ov}))$ l'est aussi, $\text{Co}(\text{Ant}, \text{Ov})$ étant la plus petite famille B d'arcs de $A(E^c)$ qui contient Ant et satisfait :

- si $[x, y]^*$ ($[y, x]^*$) est dans Ov, si $[x, z]$ est dans E et si $[z, y]$ est dans E^c , alors $[z, y]^*$ ($[y, z]^*$) est dans B ;

(c) : La structure $(X, E, \text{Nil}, \text{Nil}, \text{Ant})$ est intervalle-représentable si et seulement si le graphe $G = (X, E)$ est un graphe d'intervalles, et si la famille $\text{EXT}(\text{Ant}, E^c)$ est sans circuit, $\text{EXT}(\text{Ant}, E^c)$ étant la plus petite famille B d'arcs de $A(E^c)$ qui contient Ant et est telle que si u est dans B alors toute la classe $\text{Eq}(R(E^c), u)$ est aussi incluse dans B .

Schéma heuristique de démonstration : Nous commençons par procéder en cascade, et nous ramenons au cas où Inc est vide, puis au cas où Inc et Ov sont vides ; dans ce dernier cas, nous vérifions que notre problème consiste à

orienter les arcs du complémentaire de G de façon à obtenir une orientation transitive et acyclique qui prolonge Ant. Nous résolvons ce dernier problème en reprenant la technique utilisée par Golumbic pour caractériser des graphes d'ordre et définissons un graphe quotient pour la relation de forçage, qui s'avère être un graphe complet. Sur ce graphe quotient, nous vérifions que Ant se projette en une orientation acyclique, qui peut alors se prolonger en une relation d'ordre linéaire. Le relèvement de cette relation sur la structure de départ fournit l'orientation cherchée.

Détail de la démonstration :

Partie (a).

Un des deux sens de l'équivalence proposée est trivial (le sens \Rightarrow); afin d'obtenir la réciproque, considérons une intervalle-représentation r de la structure (X, E, Ov, Nil, Ant) , c'est-à-dire une fonction qui à tout x dans X , associe un intervalle fermé borné $r(x)$ de la droite réelle, de façon à ce que (A1), ..., (A4) soient vérifiés. Supposons cette représentation minimale en ce sens qu'il n'existe pas d'autre intervalle-représentation r' de la structure telle que pour tout x dans X , $r'(x)$ soit inclus dans $r(x)$, l'une au moins de ces relations d'inclusion étant stricte. Supposons aussi que pour tout arc $[x, y]^*$ de Inc, il n'existe aucun X , adjacent à x et non adjacent à y , et prouvons que dans ces conditions, r est aussi une intervalle-représentation de (X, E, Ov, Inc, Ant) .

Si tel n'est pas le cas, il existe $[x, y]^*$ dans Inc tel que par exemple le Début($r(y)$) > Début($r(x)$). Essayons de diminuer $r(x)$ (qui n'est pas réduit à un point) en augmentant $d(r(x))$. La minimalité de r fournit alors l'existence de z tel que :

$$[x, z] \text{ est dans } E \text{ et } \text{Fin}(r(z)) = \text{Début}(r(x)); \text{ (B1)}$$

ou bien

$$[x, z]^* \text{ est dans } Ov \text{ et } \text{Début}(r(x)) = \text{Début}(r(z)). \text{ (B2)}$$

(B1) fournit que z est adjacent à x sans être adjacent à y et une contradiction;

Dans le cas (B2), on constate que $r(z)$ ne peut être réduit à un point et on contredit la minimalité de r en augmentant simultanément $d(r(x))$ et $d(r(z))$.

Partie (b).

A nouveau le sens \Rightarrow de l'équivalence proposée est trivial.

Afin d'obtenir la réciproque, considérons une intervalle-représentation minimale (au sens ci-dessus) r de $(X, E, \text{Nil}, \text{Nil}, \text{Co}(\text{Ant}, \text{Ov}))$ et prouvons qu'il s'agit d'une intervalle-représentation de $(X, E, \text{Ov}, \text{Nil}, \text{Ant})$.

Si tel n'est pas le cas, $[x, z]^*$ existe dans Ov tel que par exemple $\text{Début}(r(y)) < \text{Début}(r(x))$. L'intervalle $r(y)$ n'est alors pas réduit à un point. Comme précédemment, nous essayons de diminuer $r(y)$ en augmentant simultanément $\text{Début}(r(y))$ ainsi que tous les $\text{Début}(r(z))$ tels que $\text{Début}(r(z)) = \text{Début}(r(y))$ et $[y, z]^*$ est dans Ov . Nous constatons une impossibilité (minimalité de r) et donc l'existence de t dans X tel que $[t, y]$ est dans E et $\text{Fin}(r(t)) = \text{Début}(r(y))$. Mais l'arc $[x, t]^*$ devrait alors être dans $\text{Co}(\text{Ant}, \text{Ov})$, ce qui induit une contradiction sur les positions relatives de $r(x)$ et $r(t)$ sur la droite réelle.

Partie (c).

Il est connu (voir [11]), que dès lors que $G = (X, E)$ est sans cycle de longueur 4 et sans corde, et que $G^c = (X, E^c)$ est orienté de façon à représenter une relation d'ordre $<_A$, alors une intervalle-représentation de G peut être trouvée telle que si $<_{Ay}$ dans X , alors $\text{Fin}(r(x)) < \text{Début}(r(y))$.

Nous obtiendrons alors notre résultat en prouvant le lemme suivant, dit Lemme d'Extension :

LEMME D'EXTENSION : Soit $G = (X, E)$ un graphe simple, A une famille d'arcs de $A(E)$.

Il est alors possible de prolonger $A(E)$ en une orientation transitive et antisymétrique des arêtes de G si et seulement si la famille $\text{EXT}(A, E)$ est sans circuit, $\text{EXT}(A, E)$ étant la plus petite famille B d'arcs de $A(E)$, qui contient A et qui est telle que pour tout u dans $A(E)$ contenu dans B , alors $\text{Eq}(R(E), u)$ est aussi incluse dans B .

Démonstration du Lemme d'Extension : Pour simplifier, les notations, nous conviendrons que tout au long de cette démonstration, la notation R désignera la relation $R(E)$.

Il est clair là encore que seul le sens (\Leftrightarrow) de l'équivalence ci-dessus est à considérer.

Rappelons tout d'abord l'énoncé du lemme dit du « Triangle », utilisé en [11] afin de caractériser les graphes d'ordre :

Si a, b, c sont dans X , forment un triangle, et sont tels que :

$$\begin{aligned} \text{Eq}(R, [a, b]^*) &\neq \text{Eq}(R, [a, c]^*); \\ \text{Eq}(R, [a, b]^*) &\neq \text{Eq}(R, [b, c]^*). \end{aligned}$$

Alors pour tout $[x, y]$ dans E tel que $\text{Eq}(R, [x, y]^*) = \text{Eq}(R, [a, b]^*)$, on a $[a, c]^* R [x, c]^*$ et $[b, c]^* R [y, c]^*$.

Il s'en déduit que si l'on suppose que deux arcs u et u^{-1} ne sont jamais dans une même classe de $A(E)$ pour R , alors que chaque classe $\text{Eq}(R, v)$, v dans $A(E)$, est transitive.

Introduisons ensuite deux notations et définitions additionnelles :

– Graphes quotients

Soit un graphe simple (ou orienté) quelconque $H = (Z, F)$, dont l'ensemble des sommets est muni d'une relation d'équivalence notée \sim ; nous définissons le *graphe quotient* $H/\sim = (Z/\sim, E/\sim)$ par :

$Z/\sim =$ Ensemble des classes d'équivalence de X par \sim .

Deux de ces classes sont adjacentes si et seulement si il existe deux représentants de chacune des classes qui sont adjacents dans H .

– Complétion absolue d'une classe $\text{Eq}(R, e)$

Pour e dans $A(E)$, nous posons $|\text{Eq}(R, e)| =$ Complétion absolue de $\text{Eq}(R, e) =$ l'union de $\text{Eq}(R, e)$ et de $\text{Eq}(R, e^{-1})$.

Supposons donc que notre graphe de départ $G = (X, E)$, et A incluse dans $A(E)$ sont tels que $\text{EXT}(A, E)$ est sans circuit, nous considérons le quotient $A(E)/R$. Pour chaque classe $c = \text{Eq}(R, u)$, nous posons $c^{-1} = \text{Eq}(R, u^{-1})$. A chaque c dans $A(E)/R$, nous associons deux objets $\text{Or}(c)$ et $\text{Dest}(c)$, que nous relient par un arc. Nous considérons que pour une même classe c , *les objets $\text{Or}(c)$ et $\text{Dest}(c^{-1})$ sont en fait les mêmes*. Nous posons enfin :

$X_1 =$ Union de tous les $\text{Or}(c), \text{Dest}(c)$, pour c dans $A(E)/R$.

Ce faisant, nous avons construit un graphe G_1 sur X_1 , symétrique, et que nous pouvons donc considérer comme un graphe non orienté, formé d'arêtes deux à deux disjointes.

Sur X_1 , nous définissons alors la relation suivante \sim :

s et r dans X_1 sont tels que $s \sim r$

ssi

$s = r$

ou bien si il existe x, y, z dans X vérifiant :

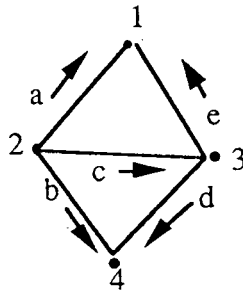
$s = \text{Or}(\text{Eq}(R, [x, y]^*)); r = \text{Or}(\text{Eq}(R, [x, z]^*));$

$|\text{Eq}(R, [x, y]^*)| \neq |\text{Eq}(R, [y, z]^*)| \neq |\text{Eq}(R, [x, z]^*)|$; (C1)

On dira dans ce dernier cas que le triangle $\{x, y, z\}$ supporte la relation $s \sim r$ et qu'il constitue un triangle fort.

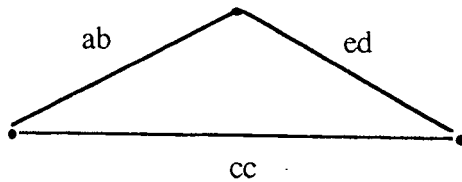
Explication : Un triangle fort est donc un triangle du graphe de départ G , dont les arêtes sont en quelque sorte indépendantes du point de vue de la relation de forçage R . La structure quotient définie à partir de la relation \sim aura donc pour objet de réduire une classe de R à un seul arc et de ramener dès lors notre problème de départ à un problème de prolongement d'un ordre partiel en un ordre linéaire.

Exemple : Soit le graphe simple $G = (X, E)$



a désigne l'arc $[2, 1]^*$, et b désigne l'arc $[2, 4]^*$ et l'on a Rb ;
 e désigne l'arc $[3, 1]^*$ et d désigne l'arc $[3, 2]^*$ et l'on a Rb ;
 c désigne l'arc $[2, 3]^*$

Les 2 triangles $\{2, 3, 4\}$ et $\{2, 3, 1\}$ sont des triangles forts ; la structure quotient définie par la relation \sim devient :



Les arêtes ab , ed et cc de la structure quotient renvoient aux classes $\{a, b\}$, $\{e, d\}$, et $\{c\}$ définies par la relation R .

LEMME 1: La relation \sim est une relation d'équivalence, dès lors que l'on suppose bien qu'il n'existe pas u dans $A(E)$ tel que $u R u^{-1}$.

Démonstration du lemme 1 : Il suffit de vérifier que \sim est transitive. Considérons pour cela s, r, p dans X_1 et 2 triangles forts $\{x, y, z\}$ et $\{x', t, z'\}$ qui supportent les relations $s \sim r$ et $r \sim p$.

La relation $[x, z]^* R [x', z']^*$ et le lemme du Triangle appliqué à partir de cette relation et du triangle $\{x', t, z'\}$ permet de dire que le triangle $\{x, z, t\}$ est fort et supporte aussi la relation $r \sim p$. Nous désirons dès lors prouver $s \sim p$. Il nous suffit de prouver que $\{x, y, t\}$ est un triangle fort. Mettant de côté le cas trivial $[x, y]^* R [x, t]^*$, on constate que l'arête $[y, t]$ est dans E . Supposons $[x, y]^* R [t, x]^*$ (et par transitivité $\text{Eq}(R, x, y), [x, y]^* R [t, y]^*$).

Le lemme du Triangle appliqué au triangle $\{x, y, z\}$ et à la relation $[t, x]^* R [x, y]^*$ produit $[z, t]^* R [z, x]^*$, ce qui contredit le fait que $\{x, z, t\}$ soit un triangle fort.

Donc $\text{Eq}(R, [t, x]^*) \neq \text{Eq}(R, [x, y]^*)$. (β)

Supposons maintenant que le triangle x, y, t ne soit pas fort, parce que par exemple $[t, y]^* R [x, y]^*$ ou $[t, y] R [x, y]^*$.

Le deuxième cas fournit $[y, x]^* R [t, x]^*$ (par transitivité de $\text{Eq}(R, [x, y]^*)$) et donc une contradiction avec (β).

Le premier cas fournit, par application du lemme du Triangle à $\{x, y, z\}$ et à la relation $[t, y]^* R [x, y]^*$, la nouvelle relation $[z, x]^* R [z, t]^*$ et une contradiction sur le fait que $\{x, z, t\}$ soit un triangle fort. FIN-LEMME1.

LEMME 2: Si s, r sont dans X_1 , c dans $A(E)/R$ tels que $s = \text{Or}(c)$, $r = \text{Or}(c^{-1}) = \text{Dest}(c)$, alors on ne peut avoir $s \sim r$. (On suppose toujours que u n'existe pas dans $A(E)$ tel que $u R u^{-1}$.)

Démonstration du lemme 2 : C'est évident compte tenu de la définition même de la relation \sim . Un triangle $\{x, y, z\}$ supportant une relation $s \sim r$ telle que dans l'énoncé du lemme ne pourrait être fort. FIN-LEMME2.

LEMME 3: Chacune des composantes connexes du graphe quotient G_1/\sim est une clique, sans boucle.

Démonstration du lemme 3 : L'absence de boucle dans G_1 découle du lemme 2. Supposons s, r, p, p' dans X_1 , c, c' dans $A(E)/R$ tels que :

$$s \sim r; \text{Or}(c) = s; \text{Or}(c') = r; \text{Dest}(c) = p; \text{Dest}(c') = p'.$$

Il s'agit pour nous de prouver que p et p' sont adjacents ou confondus dans G_1/\sim . La relation $s \sim r$ est supportée par un triangle fort $\{x, y, z\}$ qui supporte aussi les relations :

$$p \sim \text{Or}(\text{Eq}(R, [y, z]^*));$$

$$p' \sim \text{Or}(\text{Eq}(R, [z, y]^*)) = \text{Dest}(\text{Eq}(R, [y, z]^*));$$

De ce fait, p et p' sont bien adjacents dans G_1/\sim . FIN-LEMME3.

Notons maintenant que puisque chaque arête de G se projette en une arête de G_1/\sim , (qui n'est pas une boucle), les arcs de notre famille A de départ se projettent en une famille A_1 de $A(E_1/\sim)$.

LEMME 4: Si $c = \text{Eq}(R, [x, y]^*)$ et c' sont 2 éléments de $A(E)/R$ tels que : $c' \neq c^{-1}$ et $\text{Dest}(c) \sim \text{Or}(c')$, alors il existe z dans X , adjacent à y et tel que $c' = \text{Eq}(R, [y, z]^*)$. (On suppose toujours que u n'existe pas dans $A(E)$ tel que $u R u^{-1}$).

Démonstration du lemme 4 : On a bien sûr $c \neq c'$ et donc $|c| \neq |c'|$. La relation $\text{Or}(c') \sim \text{Or}(c^{-1})$ est donc supportée par un triangle fort, et le lemme du Triangle nous a déjà permis de vérifier que le représentant de c dans ce triangle pouvait être choisi comme étant $[y, x]^*$. Ce troisième sommet de ce triangle constitue alors l'élément z recherché dans notre lemme. FIN-LEMME4.

La démonstration du Lemme d'Extension (et donc du théorème 4) s'achève alors aisément. Nous supposons que $\text{EXT}(A, E)$ est sans circuit : le lemme 4 nous permet de voir que la famille A_1 d'arcs de $A(E_1/\sim)$ est aussi sans circuit, puisque si un tel circuit existait, il pourrait se relever en un circuit de $\text{EXT}(A, E)$. A_1 peut donc être prolongée en une orientation de G_1/\sim qui définit une relation d'ordre (puisque G_1/\sim est une union disjointe de cliques). A son tour, cette orientation se relève en une orientation des arêtes de G , forcément transitive, puisque tout circuit induit par cette orientation se projetterait en un circuit sur l'orientation en ordre de G_1/\sim . FIN-LEMME-EXTENSION.

COROLLAIRE : Reconnaître les structures $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$ qui sont intervalle-représentables peut être fait en temps polynomial.

Démonstration : Il suffit pour s'en rendre compte de vérifier que l'algorithme suivant, qui est polynomial, reconnaît le fait qu'une structure $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$ soit intervalle-représentable ou non et, dans le cas positif, construit l'intervalle-représentation associée.

Algorithme INTERVALLE

Input : la structure $(X, E, \text{Ov}, \text{Inc}, \text{Ant})$;

Output : Échec ou Succès, et en cas de succès, l'intervalle-représentation r de la structure ;

Début

(* 1° phase : construction d'une orientation des arêtes de E^c qui définisse un ordre et contienne Ant^*);

Not Echec ; $\text{Ordre} := \text{Nil}$; $V := E^c$;

Tant que Not Echec et V non vide faire

Début

Choisir $[x, y]^*$ dans $A(E^c)$; (Si Ant non vide, alors prendre $[x, y]^*$ dans Ant)

Si $[x, y]^* \in R(V)$ ou s'il existe $[z, t]^*$ dans Ant tel que $[t, z]^*$ est dans $\text{Eq}(R(V), [x, y]^*)$

alors Échec

sinon

Début

$\text{Ordre} := \text{Ordre} + \text{Eq}(R(V), [x, y]^*)$; $\text{Ant} := \text{Ant} - \text{Eq}(R(V), [x, y]^*)$;

$V := V - \text{Support}(\text{Eq}(R(V), [x, y]^*))$;

Fin

Fin

(* 2° phase : construction de l'intervalle-représentation r^*)

Si Not Echec alors

Début

Définir pour chaque sommet x dans X , deux éléments $d(x)$ et $f(x)$;

Sur l'ensemble $X^D = \{d(x), f(x), x \text{ dans } X\}$, définir la relation $<$ suivante :

Si $[x, y]^*$ est dans Ordre alors $f(x) < d(y)$;

Si $[x, y]^*$ est dans Inc alors $d(x) \leq d(y) \leq f(y) \leq f(x)$;

Si $[x, y]^*$ est dans Ov alors $d(x) \leq d(y) \leq f(x) \leq f(y)$;

Si \leq ainsi définie contient un circuit comportant au moins un arc étiqueté $<$, alors

Echec sinon

Début

Succès ;

Plonger les éléments de X^D sur la droite réelle de façon compatible avec la relation $<$;

La représentation r est alors définie par : $r(x) = \text{l'intervalle } [d(x), f(x)]$ pour tout x dans X ;

Fin

Fin

Fin.

Soin est laissé au lecteur de se référer à [9] et de vérifier que la première partie de l'algorithme fournit bien une extension Ordre de A du type cherché dès lors qu'une telle extension existe).

6. CONCLUSION

Nous avons focalisé, au cours de cette étude, sur la caractérisation de certaines structures « parfaites » liées à la géométrie des intervalles, et fait apparaître la possibilité de déduire, à partir de telles caractérisations,

des procédés algorithmiques simples et efficaces pour la représentation de certaines structures combinatoires visant à faire le plus possible ressembler celles-ci à ces structures « parfaites ».

Un certain nombre de prolongements de cette étude sont possibles, qui concernent soit les aspects applicatifs, soit ses aspects fondamentaux.

Mentionnons donc en guise de conclusion quelques-uns des problèmes qu'il pourrait ainsi être intéressant de résoudre.

Le problème de représentation REP-GRAPH

Il est d'une certaine façon le pendant des problèmes REP-HYP(TEMPS) et REP-HYP(INTERRUPT) traités au cours des parties précédentes. Une structure $G = (X, E, \text{Ov}, \text{Inc}, \text{Ant})$ étant proposée, on souhaite représenter les sommets de G comme des intervalles fermés de la droite réelle, de façon à ce que les relations Ov, Inc et Ant traduisent les implications (A2), (A3), (A4) formulées en début de paragraphe 5, et que la relation d'adjacence E traduise le plus possible l'équivalence (A1). Si l'on se réfère à l'exemple proposé en partie 2.1, cela signifie que la structure G va représenter les intervalles de temps couvrant complètement l'utilisation de chaque machine et que l'on souhaite que ces intervalles coïncident le plus possible avec les phases d'utilisation effective de ces machines.

Chaque arête e dans E^c étant munie d'un poids $D(e)$, on va chercher alors à associer à chaque x dans X , un intervalle $p(x)$ de la droite réelle de façon à ce que :

Problème REP-GRAPH défini par G :

- Les implications (A2), (A3), (A4) soient satisfaites ;
- Si $[x, y]$ est dans E , alors $p(x)$ et $p(y)$ sont d'intersection non vide ;
- La somme des poids $D(e)$, considérée pour les arêtes $e = [x, y]$ de E^c telles que $p(x)$ et $p(y)$ sont d'intersection non vide, est minimale.

L'extension du problème d'ordonnancement présenté au paragraphe 2 au cas où certaines tâches peuvent être exécutées en parallèle

On considère les mêmes hypothèses qu'au paragraphe II.1, en omettant celle relative à l'existence d'un même opérateur participant à l'exécution de toutes les actions et en supposant qu'une même ressource ne peut être sollicitée pour deux actions à effectuer simultanément. Un planning σ redevient alors de façon classique l'affectation à chaque action d'une date

de début et d'une date de fin, les quantités $I(e, \sigma)$ et $T(e, \sigma)$ étant définie comme précédemment.

L'évaluation systématique des performances de la méthode de résolution d'un programme linéaire entier ou mixte présentée au paragraphe 2.2

Il serait alors intéressant de voir comment cette méthode s'adapte au cas d'un programme mixte, à celui d'un programme comportant une fonction à optimiser ou encore au cas d'un programme en $\{0, 1, -1\}$, les matrices d'intervalles étant alors remplacées par des matrices plus générales possédant la propriété de totale unimodularité.

La recherche de sous-problèmes de REP-HYP(TEMPS) ou REP-HYP (INTERRUPT) qui soient polynomiaux

On pourra chercher par exemple à spécialiser le type d'hypergraphe sur lequel on travaille ou introduire des contraintes supplémentaires sur les durées d'utilisation des machines. On pourra aussi s'efforcer de dégager des garanties de performances (au sens du plus mauvais cas ou au sens probabiliste) pour les heuristiques présentées à l'intérieur de ce travail.

RÉFÉRENCES

1. J. F. ALLEN, Towards a general theory of action and time, *A.I* 23, 1984, p. 123-154.
2. P. BAPTISTE et S. FAVREL, Résolution de problèmes d'ordonnements par treillis de Galois et graphes d'intervalles, *RAIRO Automatique* 18, 4, 1984.
3. S. BENZER, On the topology of the genetic fine structure, *Proc. Acad. Sci. USA*, 1959, 45, p. 1607-1620.
4. C. BERGE, Graphes et hypergraphes, Dunod, 1975.
5. J. C. BOLAND et G. G. LEKKERKERKER, Representation of a finite graph by a set of intervals of a line, *Fund. Math.*, 1962, 51, p. 45-64.
6. K. S. BOOTH et G. S. LUEKER, Testing for the consecutive one'propety, *Journ. Comput. Sciences*, 1976, 13, p. 35-58.
7. P. DUCHET, Problème de représentation et noyaux, *Thèse d'État*, Paris-VI, 1979.
8. D. R. FULKERSON et J. R. GROSS, Incidence Matrices and interval graphs, *Pacific Journ. Math.*, 1965, 15, p. 835-855.
9. S. P. GHOSH, File organization: the consecutive retrieval property, *Comm. ACM*, 1975, 9, p. 802-808.
10. P. GILMORE et A. HOFFMAN, A characterization of compatibility graphs and interval graphs, *Can. Journ. Math.*, 1964, 16, p. 539-548.
11. M. GOLUBIC, Algorithmic graph theory and perfect graphs, *Acad. Press N. Y.*, chap. 5, 1980.
12. E. HUNTINGTON, A set of completely independent postulates for cyclic orders, *Proc. Nat. Acad. Sciences USA*, 1924, 10, p. 630-631.

13. D. G. KINDALL, Incidence matrices, interval graphs and seriation in archeology, *Pacific Journ. Math.*, 1969, 28, 3, p. 565-570.
14. F. LUCCIO et F. P. PREPARATA, Storage for consecutive retrieval, *Inform Process Letter*, 5, 1976, 3, p. 68-71.
15. A. PNUELI, A. LEMPEL et S. EVEN, Transitive of graphs and identification of permutation graphs, *Can. Journ. Math.*, 1971, 23, p. 160-175.