Une approche réactive du problème d'arbre couvrant maximal basé sur l'algorithme de Kruskal

Rémy-Robert Joseph, Laurent Linguet

Une approche réactive du problème d'arbre couvrant maximal basé sur l'algorithme de Kruskal *

Rémy-Robert Joseph[1] et Laurent Linguet[1]

Université des Antilles et de la Guyane, Institut d'Enseignement Supérieur de la Guyane
Campus universitaire Saint-Denis, Avenue d'Estrées, BP 792, 97337 Cayenne cedex, Guyane, France

[1] *GRER (Groupe de Recherche sur les Énergies Renouvelables)*

{remy-robert.joseph, laurent.linguet}@guyane.univ-ag.fr

Abstract

Dans un contexte de préférences non classiques (pouvant présenter des intransitivités et incomplétudes, voire des circuits), les problèmes combinatoires de recherche d'une solution préférée sont semi-structurés. Leur résolution s'effectue généralement grâce à des interactions entre le Système Interactif d'Aide à la Décision (SIAD) et les décideurs selon un processus de décision. En proposant des algorithmes de résolution de problèmes connexes pertinents entièrement automatisables, le SIAD aide à obtenir des éléments de réponses aux questions que se pose un intervenant dans un processus de décision.

Nous introduisons un tout nouveau type de SIAD, proposant des problèmes automatisables pertinents basés sur la déduction et le choix : les problèmes de *consistance qualitative*. Ainsi, nous présentons un algorithme glouton de consistance qualitative dans le cadre du problème d'arbre couvrant maximal basé sur une version ordinale de l'algorithme de Kruskal. Utilisés dans un processus itératif interactif, ce type de problèmes ne permet pas toujours d'aboutir à une solution maximale pour l'instance initiale. Le processus décisionnel est dit non *pleinement rationnel*. Pour y remédier, nous identifions algorithmiquement les arêtes *critiques* (présentes dans toute solution du sous-ensemble maximal courant). Enfin, nous fournissons des conditions nécessaires sur les préférences, en nous aidant des travaux en théorie du choix rationnel, pour que le processus conserve cette rationalité procédurale.

1 Introduction

La plupart des problèmes combinatoires issus du monde réel nécessitent de modéliser les incertitudes, les avis divergents et d'arbitrer des conflits, pour évaluer globalement les solutions et identifier celles de meilleur compromis. Ces caractéristiques requièrent une modélisation plus complexe des préférences [21, 23]. Dans cet article, nous nous plaçons à un niveau très général, qu'est celui des préférences globales modélisées sous forme de relation binaire sur l'ensemble des solutions possibles, ni transitive, ni même acyclique ou complète, mais admettant un ensemble maximal (il n'existe aucune solution qui leur sont strictement préférées).

Depuis quelques années, nous assistons à la floraison d'un grand nombre de publications traitant des problèmes combinatoires multiobjectifs (voir [9] pour une synthèse). Cependant, très peu d'articles ont franchi le pas des problèmes combinatoires à information préférentielle purement ordinale et/ou non transitive. Citons les travaux récents en théorie de la décision d'Olivier Spanjaard et de Patrice Perny [19] dans le cadre des arbres couvrants et des chemins d'un graphe, les travaux en théorie des jeux (voir [20] pour une synthèse), dans le cadre des couplages stables, les travaux en optimisation combinatoire algébrique [3], ou encore ceux en intelligence artificielle sur des problèmes de configuration [2, 15] et de parcours de graphes d'états [18, 15]. Ces travaux restent cependant soit cantonnés à la théorie, soit dédiés à un problème particulier. Pour le moment, les réflexions portant sur la manière d'aborder concrètement ce type de problème sont à leur début [15]. Nous avons décidé d'apporter une pierre supplémentaire à l'édifice.

Avec de telles préférences, la problématique décisionnelle de la recherche d'une solution de meilleure qualité est semi-structurée (on dira aussi semi-automatisable) : Dans le cas général (en dehors des préordres totaux), une solution préférée adéquate pour l'utilisateur ne peut être identifiée uniquement à partir des informations préférentielles. Dans la pratique, on ne peut pas limiter la résolution de ces problèmes en fournissant automatiquement une solution de meilleure qualité. Car les solutions de meilleure qualité ne sont pas toutes préférentiellement équivalentes, certaines peuvent être partiellement ou non comparables, et à certains moments, il peut même n'exister aucune solution optimale ou maximale.

A notre connaissance, le seul processus de résolution de problèmes combinatoires purement ordinaux

* **Cet article est une version française sans démonstration de celle soumise (10/2006) à la revue *Discrete Optimization* avec le titre "A deductive interactive approach of the Kruskal-based maximal spanning tree sub-problem".**

proposé [19, 29] est basé sur l'énumération des solutions préférées, le décideur étant en charge d'identifier la solution adéquate parmi les préférées. Malheureusement, dans le pire cas, le nombre de solutions préférées peut atteindre une taille exponentielle en la taille de l'instance. Phénomène qui limite l'application de cette procédure aux instances de petite taille [19, 29]. Dans cet article, nous présentons une autre manière de résoudre les problèmes combinatoires préférés.

Pour investiguer ces problèmes semi-automatisables, nous avons décidé d'utiliser le concept de Système Interactif d'Aide à la Décision (SIAD) – et la théorie du traitement de l'information qui le supporte – pour explorer l'ensemble des solutions préférées. Cette exploration ne se fait pas en construisant itérativement de nouvelles solutions préférées – comme il est fait en programmation multicritère –, mais en décrivant cet ensemble :

- par les valeurs présentes dans au moins une solution préférée, et
- par les valeurs communes à chaque solution préférée.

La notion de consistance définie en PPC, forme l'environnement théorique de cette approche descriptive des ensembles donnés en compréhension.

Ainsi, après un rappel sur les relations de préférence (§ 2.1), nous donnons quelques caractéristiques de l'approche SIAD des problèmes complexes (§ 2.2). Au § 3 nous effectuons une présentation sommaire de la consistance en PPC (§ 3.1), et nous introduisons ensuite (§ 3.2) la consistance qualitative, en d'autres termes, comment redéfinir la notion de consistance afin de tenir compte des particularités préférentielles de nos problèmes combinatoires.

Afin de montrer toute la puissance que renferme cette approche des problèmes combinatoires exploitant des préférences non conventionnelles, nous étudions le cas particulier des arbres couvrants maximaux (§ 4). Dans le cas général, l'essentiel des problèmes automatisables connexes pertinents au problème décisionnel initial sont difficiles (NP-complets). Par conséquent, nous nous focalisons sur un sous-problème[1] dont la recherche automatique d'un seul arbre couvrant maximal est facile (§ 4.2) – problème basé sur une version ordinale de l'algorithme de Kruskal [19, 29] – pour lequel nous élaborons un algorithme de consistance qualitative (§ 4.3). Nous abordons ensuite l'aspect procédural de la résolution, en montrant comment cet algorithme peut être utilisé dans un processus décisionnel et en se focalisant sur les limites interactives de son utilisation (§ 5.1). L'une des limites étant la négligence de la pertinence des arêtes critiques – i.e. arêtes communes à tous les arbres couvrants maximaux –, nous présentons au § 5.2 un algorithme permettant de résoudre ce problème. L'autre limite étant la possibilité d'aboutir en fin de processus décisionnel à un arbre couvrant non maximal, nous proposons au § 5.3 des conditions nécessaires pour aboutir à une solution maximale.

Nous concluons cet article (§ 6) avec une synthèse des résultats obtenus et des perspectives liées à cette approche des problèmes combinatoires exploitant des préférences non conventionnelles.

[1] Un **sous-problème** Π_1 d'un problème Π donné, est un problème qui répond à la même question que Π, mais seulement sur un sous-ensemble des instances de Π.

2 Préférences, processus décisionnels et consistance

2.1. Relation de préférence

Étant donné un ensemble fini S, une **relation (binaire) de préférence** (**rbp**) [25, 21, 26, 23] $\succcurlyeq$ d'un individu sur S est une relation binaire réflexive sur S ($\Leftrightarrow \succcurlyeq \subseteq S \times S$ et $\forall\, x \in S$, $(x, x) \in \succcurlyeq$) traduisant les jugements de cet individu concernant ses préférences entre les paires d'alternatives. Quelque soit le couple d'alternatives x et y de S, l'assertion « $x \succcurlyeq y$ » est équivalent à « $(x, y) \in \succcurlyeq$ » et signifie que « x est au moins d'aussi bonne qualité qu' y pour l'individu considéré ». Une relation binaire de préférence $\succcurlyeq$ effectue une partition de $S \times S$ en quatre relations fondamentales : Pour tout $x, y \in S$

(**indifférence**) $x \simeq y \Leftrightarrow (\ x \succcurlyeq y$ et $y \succcurlyeq x\ )$
(**préférence stricte**) $x \succ y \Leftrightarrow (\ x \succcurlyeq y$ et non$(y \succcurlyeq x)\ )$
(**aversion stricte**) $x \prec y \Leftrightarrow y \succ x$
(**incomparabilité**) $x \parallel y \Leftrightarrow (\ non(x \succcurlyeq y)$ et non$(y \succcurlyeq x)\ )$

Les relations de préférence définies sur un ensemble fini correspondent formellement à celle de graphes orientés simples. Par conséquent, la représentation graphique de ces derniers nous permet d'illustrer nos propos. Pour tout $A \subseteq S$, la **restriction** de $\succcurlyeq$ à A est la relation de préférence $\succcurlyeq_{|A}$ définie comme suit : $\succcurlyeq_{|A} = \{(x, y) \in A \times A$, tel que : $x \succcurlyeq y\}$. Par abus, nous ne précisons pas la restriction, le contexte permettant d'identifier le sous-ensemble de S auquel nous nous rapportons. Une relation de préférence $\succcurlyeq$ est dite :

- **transitive** ssi $[x \succcurlyeq y$ et $y \succcurlyeq z] \Rightarrow x \succcurlyeq z$, $\forall\, x, y, z \in S$
- **quasi-transitive** ssi $[x \succ y$ et $y \succ z] \Rightarrow x \succ z$, pour tout $x, y, z \in S$
  ssi la préférence stricte est transitive
- **P-acyclique** ssi $\forall\, t > 2$ et $\forall\, x_1, x_2, \ldots, x_t \in S$, $[x_1 \succ x_2 \succ \ldots \succ x_t] \Rightarrow$ non$(x_t \succ x_1)$
  ssi $\succcurlyeq$ est sans circuit de préférences strictes.
- une **relation d'équivalence** ssi elle est réflexive[2], symétrique et transitive
- un **préordre partiel** ssi elle est réflexive et transitive
- un **préordre complet** ssi elle est réflexive, transitive et complète

Étant donné un ensemble S sur lequel est défini une relation de préférence $\succcurlyeq$, l'**ensemble maximal** (ou

[2] Mentionnons qu'une relation binaire $\succcurlyeq$ est **symétrique** ssi $x \succcurlyeq y \Rightarrow y \succcurlyeq x$, pour tout $x, y \in S$; **asymétrique** ssi $x \succcurlyeq y \Rightarrow$ non$(y \succcurlyeq x)$, pour tout $x, y \in S$; et **complète** (ou **totale**) ssi $x \succcurlyeq y$ ou $y \succcurlyeq x$, pour tout $x, y \in S$ et $x \neq y$.

ensemble efficace) de S selon $\succcurlyeq$, noté $M(S, \succcurlyeq)$, est le sous-ensemble de S vérifiant : $M(S, \succcurlyeq) = \{x \in S \mid \forall y \in S, \text{non}(y \succ x)\}$; Alors que l'**ensemble optimal** de S selon $\succcurlyeq$, noté $B(S, \succcurlyeq)$, est le sous-ensemble de S vérifiant : $B(S, \succcurlyeq) = \{x \in S \mid \forall y \in S, x \succcurlyeq y\}$. Pour des approfondissements sur les axiomes de choix de solutions de meilleure qualité à partir d'une relation de préférence, nous renvoyons à [10, 31, 14].

Dans les applications combinatoires, les solutions sont décrites par un ensemble de composantes élémentaires d'un ensemble E ($\Leftrightarrow S \subseteq \mathcal{P}(E)$). Il est donc nécessaire de concevoir une **représentation compacte des préférences**, la saisie et le traitement des couples $x, y \in S$ tels que $x \succcurlyeq y$ étant dans la pratique inenvisageable. Ainsi, en optimisation combinatoire classique, les préférences sont représentées par une fonction analytique f de $\mathcal{P}(E)$ dans $\mathbb{R}$ à minimiser : $x \succcurlyeq y \Leftrightarrow f(x) \leq f(y)$; En optimisation multicritère basée sur la dominance de Pareto, les préférences sont représentées par un vecteur de fonctions analytiques $(f_1, \ldots, f_k)$, appelées (vrai-)critères, agrégées par la dominance de Pareto : $x \succcurlyeq y \Leftrightarrow [\forall i \in \{1, \ldots, k\}, f_i(x) \leq f_i(y)]$. En intelligence artificielle, plusieurs représentations compactes des préférences ont fait leur apparition : des CP-nets [2, 15] aux contraintes de (description de) voisinage préférentiel (appelées contraintes préférentielles dans [13]), en passant par les contraintes souples et les CSP dynamiques (pour une synthèse bibliographique, voir [34]). Par la suite, une représentation compacte quelconque d'une relation de préférence $\succcurlyeq$ est notée $I(\succcurlyeq)$. Nous présenterons au § 4 la représentation compacte que l'on utilisera pour les arbres couvrants maximaux.

2.2. Systèmes interactifs d'aide à la décision

Depuis les travaux pionniers d'H. A. Simon [27, 17, 28], l'intelligence artificielle donne un cadre très général pour aborder les problèmes complexes organisationnels au moyen d'un système d'information. Il distingue deux types de problèmes décisionnels :

- les problèmes **automatisables** dont la résolution peut se faire entièrement avec une procédure normalisée (généralement un programme informatique). Exemple : les problèmes classiques de RO.
- les problèmes **semi-automatisables** (ou *semi-structurés*), qui ne sont pas automatiquement solvables, à cause du plus ou moins grand manque de structure.

Ce manque structurel peut se trouver au niveau de la définition des solutions (un modèle réalisable pas complètement identifié et formalisé par des contraintes), ou au niveau de l'évaluation des solutions (modèle préférentiel ne permettant pas de porter un jugement global clair identifiant une solution préférée adéquate). Dans la suite, nous nous focalisons sur les problèmes semi-structurés comportant un modèle réalisable formalisé, comme pour le problème des arbres couvrants maximaux, où les solutions réalisables – les arbres couvrants d'un graphe – sont clairement identifiées.

Pour résoudre de tels problèmes semi-structurés [16], les procédures entièrement pré-programmées par le concepteur sont généralement inacceptables. Il est nécessaire d'utiliser des outils interactifs pour aider le décideur dans sa démarche de recherche de solutions de meilleur compromis **adéquate**. Le concept de **système interactif d'aide à la décision** (SIAD) a été défini dans ce but : aborder les problèmes complexes au moyen d'un ordinateur. L'un des principes des SIAD est le suivant [5] : « Quand vous posez une question à un système d'information, vous n'attendez pas qu'il choisisse pour vous. Vous attendez qu'il fasse une déduction, un raisonnement plausible, mais pas que le système d'information prenne une décision ».

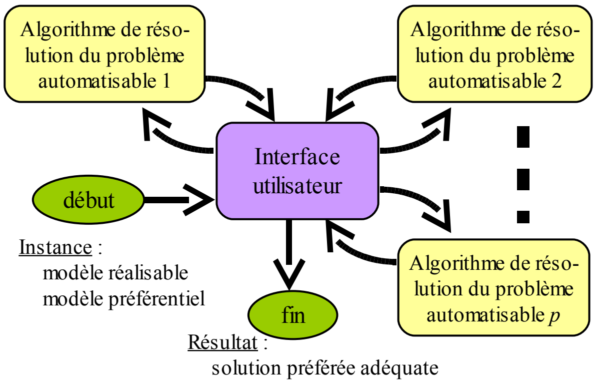

Fig. 1 : Représentation schématique du fonctionnement d'un SIAD

Le **processus de résolution de problèmes** d'un SIAD est basé sur la **recherche heuristique**. Celle-ci consiste à avancer par petits pas, et à chaque pas l'utilisateur doit décider s'il est dans la bonne direction, avec la possibilité de rebrousser chemin ou de changer de chemin dans la cas contraire. « Il est plus important de faire des petits pas dans la bonne direction que de faire des grands pas sans savoir où l'on va » : extrait de [5]. Pour pouvoir décider à chaque pas de la direction à prendre, le SIAD propose à l'utilisateur une batterie de problèmes automatisables (voir Fig. 1) que le système peut résoudre, problèmes qui doivent apporter une réelle valeur ajoutée au décideur dans la compréhension du problème et dans sa recherche de solution *adéquate*.

Les problèmes automatisables ne prennent en compte qu'un aspect des choses ; ils ne peuvent pas tout faire. L'utilisateur et le système doivent être complémentaires ; d'où l'importance de proposer des outils pertinents. L'interactivité doit être considérée comme un outil de conception d'une solution adéquate.

En optimisation combinatoire multicritère, un certain nombre de problèmes automatisables liés à un problème semi-structuré, ont été proposés. Ceux-ci sont inspirés des problèmes combinatoires de l'optimisation classique (voir [23] pour une liste plus complète) :

- Rechercher une solution de meilleure qualité, satisfaisant des niveaux d'exigence minimale sur certains critères, si une telle solution existe ;
- Rechercher les solutions de meilleure qualité, le plus proche possible d'un point de référence ;
- Lister les solutions de meilleure qualité ;
- Prélever un échantillon (représentatif ?) de solutions de meilleure qualité ;
- etc.

Ces SIAD se placent dans la logique formulé par Bernard Roy [5, p.16] : « Les calculs d'optimisation restent, à mon sens, fondamentaux pour éclairer des décisions. […] Éclairer des décisions, c'est accepter des hypothèses de travail pour trouver des éléments de réponse à des questions que l'on se pose mais cela n'est pas, là encore, découvrir la vérité, trouver la décision optimale pour dire au responsable: voilà, Madame, Monsieur, ce que vous devez faire ». Leurs défauts ? En voici quelques uns (pour d'autres critères d'évaluation des SIAD, nous renvoyons à [30]) :

(*a*) Ils requièrent pour la plupart l'usage de l'espace des critères pour guider la recherche de la solution préférée adéquate. Ils sont par conséquent inadaptés aux problèmes combinatoires purement ordinaux.

(*b*) Un choix doit être fait en **temps réel** dans un contexte interactif. Par conséquent, il est nécessaire d'éviter durant le processus de décision toute information (fournie par le SIAD) dont la quantité ne peut être traitée en un temps raisonnable, à la fois par la machine et par l'homme [28]. C'est-à-dire, en un temps polynomial en la taille de l'information d'entrée pour la machine, <u>et</u> en un temps proche du linéaire, voire moins pour l'homme. Certains problèmes automatisables (énumération notamment) ne satisfont pas ces critères de qualité pratique nécessaire d'un SI.

(*c*) Certains problèmes nécessitent une **approche réactive**, c'est-à-dire que la solution à adopter n'est connue qu'au dernier moment. Par opposition, d'autres se satisfont d'une **approche proactive**, c'est-à-dire que la solution est calculée dès le début, et le but est d'en dévier le moins possible et ce jusqu'à sa réalisation. Dans ce dernier cas, une analyse de robustesse du résultat est indispensable [23, 5, 22]. Il existe bien sûr des approches intermédiaires. L'essentiel des problèmes automatisables utilisés actuellement dans le multicritère sont adaptés à une approche proactive. Or, proposer au décideur un outil qui lui permette d'exploiter l'aspect combinatoire des problèmes, notamment en morcelant sa décision (l'identification de la solution *adéquate*) et en l'étalant si possible dans le temps, serait un atout indéniable.

Par contre, ces problèmes automatisables satisfont généralement une propriété indispensable à un SIAD efficace :

(*d*) Le processus de décision doit donner la possibilité d'atteindre n'importe quelle solution préférée en un nombre d'étapes calcul-dialogue qui est polynomial en la taille de l'information en entrée.

Nous présentons ici un autre type de problèmes automatisables, apportant une réponse positive à toutes ses propriétés (hormis la complexité algorithmique) pour résoudre les problèmes combinatoires purement ordinaux. Ce type de problèmes s'inspire directement de la problématique de la consistance de l'IA (PPC), mieux adaptée à la problématique mixte combinatoire + préférences non conventionnelles. Nous faisons une présentation succincte :

3 PPC et consistance qualitative

3.1. Consistance et déduction

En programmation par contraintes, la **consistance** fait partie d'une problématique plus générale qu'est la description. L'objectif de la consistance est de décrire un ensemble – en l'occurrence l'ensemble des instanciations réalisables pour les CSP – par les valeurs ou des combinaisons de valeurs contenues dans ses éléments. De telles valeurs et combinaisons de valeurs sont dites **consistantes**. Dans le cas contraire elles sont appelées **inconsistantes**.

Il existe plusieurs problèmes de consistance, selon que l'on retourne des valeurs ou des combinaisons, mais aussi selon que l'on retourne <u>toutes</u> les valeurs consistantes ou un sur-ensemble (une approximation). Nous nous limitons aux formes de consistance retournant les valeurs consistantes ou un sur-ensemble, de manière à respecter l'aspect humain de la propriété (*b*) d'un bon SIAD.

Les problèmes de consistance sont généralement couplés à un processus décisionnel, l'utilisateur étant un être humain ou un agent algorithmique préprogrammé par le concepteur. Dans le cas d'un être humain, la consistance permet de mettre en œuvre une *approche réactive*, basée sur la **déduction**. Dans le cas contraire, nous revenons à une approche proactive. Utilisé dans un processus décisionnel, la consistance est liée à un ensemble d'actions que l'utilisateur peut effectuer :

- supprimer des valeurs ;
- instancier (affecter une valeur à) des variables ;
- remettre en question des suppressions ou des affectations (retour arrière).

Le système d'information réagit (**approche déductive**) en effectuant un certain nombre de traitements :

- vérifier les valeurs consistantes
- supprimer les valeurs inconsistantes
- instancier (affecter) les variables possédant une seule valeur consistante.

Remarquons que les approches déductives ont été très tôt utilisées en RO, notamment dans les problèmes d'ordonnancement : Les méthodes PERT (Program Evaluation Review Technique) et CPM (Critical

Path Method) résolvent les problèmes d'ordonnancement et de gestion de projets en proposant uniquement comme problèmes automatiques dans le SIAD de restreindre les intervalles de dates de début et de fin possibles des différentes tâches, et ce de manière dynamique en fonction des nouvelles informations parvenant sur l'état d'avancement des projets, avec identification de chemins critiques. Nos travaux vont aussi dans ce sens : restriction du domaine d'investigation en fonction des actions de l'utilisateur lors de l'étape de dialogue du SIAD, et identification des valeurs critiques = présentes dans toutes les solutions. Nous y reviendrons.

Une telle résolution réactive et déductive semble très bien adaptée aux problèmes admettant des intransitivités et des incomplétudes préférentielles. Ainsi en un nombre polynomial d'actions de l'utilisateur (suppressions et instanciations), on aboutit à une solution de meilleur compromis (propriété (*d*) d'un SIAD opérationnel).

3.2. Consistance qualitative

La consistance peut être modifiée pour tenir compte des informations préférentielles, dans le cadre des problèmes combinatoires exploitant des préférences non classiques. Simplement, la consistance ne se rapportera plus à la faisabilité mais à la meilleure qualité, au meilleur compromis. Ainsi, nous ne supprimerons pas les valeurs inconsistantes dans le sens qu'elles n'appartiennent à aucune solution réalisable, mais plutôt, nous les supprimerons parce qu'elles n'appartiennent à aucune solution préférée. Dans ce cadre, nous parlons de **consistance qualitative**.

Comme en programmation par contraintes, plusieurs niveaux de consistance qualitative peuvent être définis, selon que l'on supprime tout où une partie de l'information inconsistante. On parlera de **consistance qualitative globale**, lorsque l'on s'intéressera à la suppression de toute l'information inconsistante d'un point de vue préférentiel.

Pour mieux concrétiser tout ce qui vient d'être présenté, dans ce qui suit, nous étudions en détail la consistance qualitative globale dans le cadre des arbres couvrants maximaux.

Plusieurs problèmes de consistance ont été récemment étudiés sur les arbres couvrants. Notons les travaux de [33] sur la consistance associée aux arbres couvrants d'un graphe. D'autres travaux ont porté sur la consistance associée aux arbres couvrants pondérés [7], et aux arbres couvrants de poids minimum [8]. Mais pour le moment, aucune étude de consistance ne concerne les problèmes d'arbres couvrants exploitant des préférences non-classiques.

4 Consistance qualitative globale et arbres couvrants maximaux

4.1. Introduction

Tout d'abord, faisons un petit rappel : Étant donné un graphe non orienté $G = (V, E)$ avec V les sommets et E les arêtes, un **arbre couvrant** x de G est un graphe partiel de G connexe et acyclique. x est donc toujours composé de $|V| - 1$ arêtes. Notons $S_{ST}(G)$ l'ensemble des arbres couvrants de G. Par abus, nous écrirons : $e \in x$, avec $e \in E$, pour dire que : e est une arête de l'arbre couvrant x.

Maintenant, considérons le problème de la recherche d'un arbre couvrant maximal <u>et</u> adéquat pour le décideur. Noté DECISION SUPPORT(ST/BPR/MAX), il se formule de la manière suivante :

DECISION SUPPORT(ST/BPR/MAX) : Étant donnés un graphe non orienté $G = (V, E)$ et une représentation compacte $I(\succcurlyeq)$ d'une relation binaire de préférence $\succcurlyeq$ sur $\mathcal{P}(E)$, retourner un arbre couvrant :

(i) maximal pour $\succcurlyeq$, s'il en existe, et
(ii) en adéquation avec le système de valeurs de l'utilisateur.

Sinon retourner 'non'.

Remarquons que cette définition de problème implique que la solution adéquate, doit aussi être maximale pour $(S_{ST}(G), \succcurlyeq)$. En d'autres termes le seul degré de liberté laissé à l'utilisateur du SIAD est le choix d'une solution adéquate <u>parmi</u> les solutions maximales. Cette définition renvoie par exemple aux contextes où des préférences ont été formulées par les différents acteurs du problème décisionnel, puis agrégées en des préférences globales $\succcurlyeq$ sur les solutions $\mathcal{P}(E)$ via une représentation compacte $I(\succcurlyeq)$; Maintenant, un individu : l'utilisateur, capable d'apporter efficacement l'information préférentielle manquante, est chargé de rechercher la solution adéquate reflétant au mieux les préférences globales.

A ce problème semi-structuré est associé le problème automatisable de recherche d'un arbre couvrant maximal, noté ST/BPR/MAX, dont la définition correspond à celle de DECISION SUPPORT(ST/BPR/MAX) en supprimant la propriété (ii). Dans un cadre aussi général, ces problèmes sont très difficiles. Pour s'en convaincre, il suffit de prendre le cas particulier où la représentation compacte des préférences utilisée est celui du multicritère. Ainsi, le problème d'appartenance associé aux arbres couvrants multi-objectif est NP-complet [4, 11].

Considérons maintenant le problème technique général suivant, de consistance qualitative sur les arbres couvrants maximaux d'un graphe :

GC(ST/BPR/MAX) : Étant donnés un graphe non orienté $G = (V, E)$ et une représentation compacte $I(\succcurlyeq)$ d'une relation binaire de préférence $\succcurlyeq$ sur $\mathcal{P}(E)$, retourner toutes les arêtes de E qui font partie d'un arbre couvrant maximal pour $\succcurlyeq$, s'il en existe. Sinon retourner 'non'.

Une arête e est dite **consistante** pour $(G, I(\succcurlyeq))$ s'il existe au moins un arbre couvrant maximal pour $(S_{ST}(G), \succcurlyeq)$ contenant e. Dans le cas contraire, elle est dite **inconsistante** pour $(G, I(\succcurlyeq))$.

Nous ne nous sommes pas penchés sur la complexité de ce problème. Mais il y a de grandes chances qu'il soit encore plus difficile que ST/BPR/MAX, au vue des résultats obtenus en PPC. Afin pourtant de mieux apprécier l'utilisation de ce type de problèmes dans un SIAD, tournons-nous vers un sous-problème de ST/BPR/MAX que l'on peut résoudre efficacement.

4.2. ST/KRUSKAL(MAX, MAX)/MAX

Notons la définition suivante : Considérons un graphe non orienté $G = (V, E)$ et une relation binaire de préférence P-acyclique $\succcurlyeq_E$ sur E. Alors une relation binaire de préférence $\succcurlyeq$ sur $\mathcal{P}(E)$ est dite **KRUSKAL(MAX, MAX)** pour le couple $(G, \succcurlyeq_E)$ ssi ses éléments maximaux sur $S_{ST}(G) \subseteq \mathcal{P}(E)$ peuvent être obtenus en appliquant la version ordinale de l'algorithme de Kruskal présenté en Fig. 2, sur l'instance $(G, \succcurlyeq_E)$.

KruskalOrdinalMax1($G = (V, E)$: graphe non orienté, $\succcurlyeq_E$: rbp P-acyclique sur E) :
retourner {ensemble d'arêtes, non}
début
(1) **si** (nbConnectedComponents(G) > 1) **alors**
retourner non **fin si**
(2) $A \subseteq E \leftarrow \varnothing$
(3) $B \subseteq E \leftarrow E$
(4) **tant que** ($B \neq \varnothing$) **faire**
% invariants de boucle : A ne contient pas de cycle, $|A| \leq |V| - 1$ et $B \cap A = \varnothing$
(5) $e \leftarrow$ choisir($M(B, \succcurlyeq_E)$)
% choisit un élément maximal dans B
(6) $B \leftarrow B \setminus \{e\}$
(7) **si** (nbConnectedComponents(V, A) > nbConnectedComponents($V, A \cup \{e\}$))
alors $A \leftarrow A \cup \{e\}$ **fin si**
(8) fin tant que
(9) **retourner** A
fin KruskalOrdinalMax1

Fig. 2 Un algorithme résolvant le problème ST/KRUSKAL(MAX, MAX)/MAX.

L'algorithme ci-dessus résout le problème automatisable de recherche d'un arbre couvrant maximal pour une relation de préférence KRUSKAL(MAX, MAX). Ce sous-problème de ST/RBP/MAX est noté ST/KRUSKAL(MAX, MAX)/MAX. Une version énumérative de cet algorithme, résolvant donc ENUM(ST/KRUSKAL(MAX, MAX)/MAX), a été présenté dans [19, 29] [3].

Cet algorithme suppose connu :
- Un autre algorithme nbConnectedComponents résolvant le problème du dénombrement des composantes connexes d'un graphe non orienté. Ce problème est connu pour être solvable en un temps linéaire (par une recherche en profondeur d'abord) quelque soit le graphe non orienté donné (voir par exemple [24, § 6.3 p. 90]).
- Une stratégie de choix choisir qui retourne un élément de l'ensemble en extension donné en argument si celui-ci est non vide. Sinon, retourne 'non'.

Exemple 1. La Fig. 3 illustre le cas d'un graphe non orienté complet G formé de 3 sommets, d'une relation binaire $\succcurlyeq$ sur l'ensemble des parties de E et d'une de ses représentations compactes $\succcurlyeq_E$. Pour désigner un ensemble d'arêtes, par exemple $\{a, b\}$, nous adoptons comme notation ab. La version ordinale de l'algorithme de Kruskal permet d'identifier l'unique solution maximale de $(S_{ST}(G), \succcurlyeq)$: l'arbre couvrant ab.

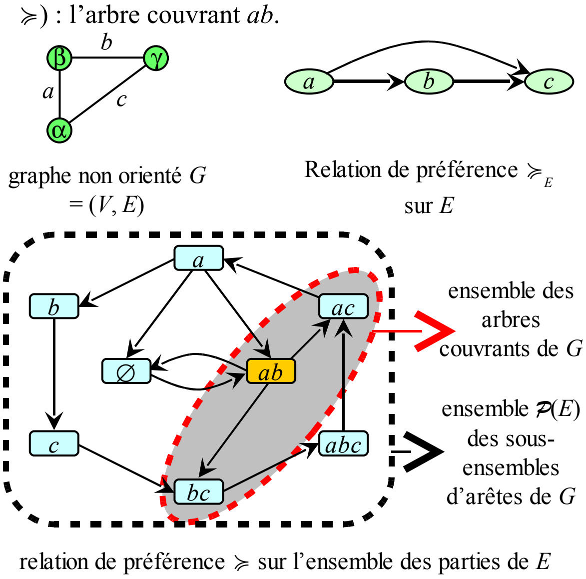

Fig. 3 : Illustration des arbres couvrants maximaux [4].

Vous remarquerez que la seule information disponible sur $(S_{ST}(G), \succcurlyeq)$ est l'existence d'un ensemble maximal qui est obtenu via un algorithme polynomial à partir du couple $(G, \succcurlyeq_E)$. Par conséquent, la relation $\succcurlyeq$ peut comporter des circuits de préférences strictes, cela n'a aucune incidence sur l'existence de l'ensemble maximal. Bien entendu si l'algorithme retourne 'non', cela signifie qu'il n'existe pas de solution maximale dans $(S_{ST}(G), \succcurlyeq)$; ce qui est équivalent à dire que $S_{ST}(G) = \varnothing$ pour notre sous-problème d'arbres couvrants. Le couple ($\succcurlyeq_E$, KruskalOrdinalMax1) forme la *représentation compacte* de la relation $\succcurlyeq$, car si on utilise un autre

[3] Notre interprétation du lien entre la relation $\succcurlyeq_E$ et la relation $\succcurlyeq$ est cependant différente de celle de [19, 29], lequel suppose en plus que la *règle d'extension* [1] est satisfaite : $\forall\, e_1, e_2 \in E,\ e_1 \succcurlyeq_E e_2 \Leftrightarrow \{e_1\} \succcurlyeq \{e_2\}$. Mais cette information ne fait que limiter inutilement la portée des algorithmes de résolution.

[4] Les relations de préférence utilisées (sur E ou l'ensemble de ses parties) étant réflexives, pour éviter les surcharges graphiques, les arcs réflexifs ne sont pas représentés.

algorithme, par exemple une version ordinale de Prim [19], on identifie un autre ensemble maximal, et donc on modélise une autre relation.

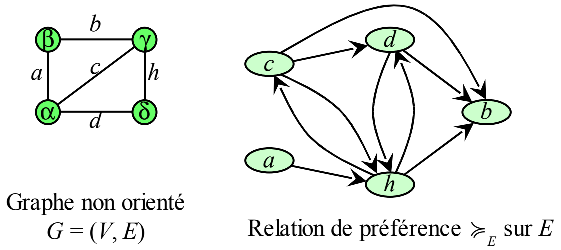

Fig. 4 Exemple[6] d'instance du problème ST/KRUSKAL(MAX, MAX)/MAX.

Exemple 2 : Considérons l'instance $(G, \succcurlyeq_E)$, illustrée à la Fig. 4, du problème ST/KRUSKAL(MAX, MAX)/MAX :

- un graphe non orienté $G = (V, E)$, avec $V = \{\alpha, \beta, \gamma, \delta\}$ et $E = \{a, b, c, d, h\}$;
- une relation binaire P-acyclique $\succcurlyeq_E$ sur E qui, en plus des arcs réflexifs, vérifie :

$a \succ_E h$, $c \succ_E b$, $c \succ_E d$, $d \succ_E b$, $h \succ_E b$, $c \simeq_E h$, $d \simeq_E h$.

Alors l'ensemble des arbres couvrants réalisables est $S_{\mathrm{ST}}(G) = \{abd, abh, acd, ach, adh, bcd, bch\}$; Notons $S_{\mathrm{ST/K/MAX}}(G, \succcurlyeq_E)$ l'ensemble des arbres maximaux obtenu en appliquant l'algorithme de Kruskal ordinal : $S_{\mathrm{ST/K/MAX}}(G, \succcurlyeq_E) = \{acd, ach\}$.

Remarquons que pour fonctionner, cet algorithme suppose qu'il existe à toute itération un ensemble d'arêtes maximales non vides (ligne 5) ; propriété qui est assurée pour tout sous-ensemble non vide d'arêtes si et seulement si la relation de préférence est P-acyclique (voir [25] pour une démonstration).

Dans ce qui suit, nous proposons un algorithme permettant de résoudre le problème de consistance globale associé à ce sous-problème, nommément GC(ST/KRUSKAL(MAX, MAX)/MAX).

4.3. GC(ST/KRUSKAL(MAX,MAX)/MAX)

Au lieu d'énumérer tous les arbres couvrants maximaux, ou de rechercher un tel arbre, on peut s'intéresser à supprimer les arêtes qui ne font partie d'aucun arbre couvrant maximal. En d'autres termes, notre intérêt porte sur le problème de consistance qualitative associé à l'algorithme de Kruskal ordinal. Plus formellement on écrit :

GC(ST/KRUSKAL(MAX, MAX)/MAX) : Étant donnés un graphe non orienté $G = (V, E)$ et une relation de préférence P-acyclique $\succcurlyeq_E$ sur E induisant une relation binaire de préférence $\succcurlyeq$ sur $\mathcal{P}(E)$ qui est KRUSKAL(MAX, MAX) pour $(G, \succcurlyeq_E)$, retourner toutes les arêtes de E qui font partie d'un arbre couvrant maximal pour $\succcurlyeq$, s'il en existe. Sinon retourner 'non'.

Notons $S_{\mathrm{GC(ST/K/MAX)}}(G, \succcurlyeq_E) \subseteq E$, l'ensemble des arêtes retourné par un algorithme résolvant ce problème. Alors, par définition, nous avons l'égalité suivante :

$$S_{\mathrm{GC(ST/K/MAX)}}(G, \succcurlyeq_E) = \bigcup_{x \in S_{\mathrm{ST/K/MAX}}(G, \succcurlyeq_E)} x \,. \qquad (1)$$

Égalité qui est équivalente aux deux assertions suivantes :

(a) Pour tout $e \in S_{\mathrm{GC(ST/K/MAX)}}(G, \succcurlyeq_E) \subseteq E$, il existe $x \in S_{\mathrm{ST/K/MAX}}(G, \succcurlyeq_E) \subseteq \mathcal{P}(E)$, tel que : $e \in x$.

(b) Pour tout $x \in S_{\mathrm{ST/K/MAX}}(G, \succcurlyeq_E) \subseteq \mathcal{P}(E)$, $x \subseteq S_{\mathrm{GC(ST/K/MAX)}}(G, \succcurlyeq_E)$.

Pour résoudre ce problème de consistance remarquons que si une arête (par ex. l'arête d de notre exemple précédent) fait partie d'un arbre couvrant maximal, alors toutes les arêtes qui lui sont préférées sont dans cet arbre couvrant (arête c), et ceci récursivement. Nous obtenons donc l'algorithme suivant :

GCKRUSKALORDINALMAX1($G = (V, E)$: graphe non orienté, $\succcurlyeq_E$: rbp P-acyclique sur E) : **retourner** {ensemble d'arêtes, non}

début

(1) **si** (NBCONNECTEDCOMPONENTS(G) > 1) **alors** **retourner** non **fin si**

(2) $A \subseteq E \leftarrow \varnothing$

(3) $B \subseteq E \leftarrow E$

(4) $C(e) \subseteq E \leftarrow \varnothing$, pour tout $e \in E$

(5) **tant que** ($B \neq \varnothing$) **faire** % invariants de boucle : $A \cap B = \varnothing$ et $B \cap C(e) = \varnothing$

(6) $e \leftarrow$ CHOISIR($M(B, \succcurlyeq_E)$)

(7) $B \leftarrow B \setminus \{e\}$

(8) $C(e) \leftarrow \left(\bigcup_{e' \succ_E e} C(e') \cup \{e'\} \right)$

(9) **si** (NBCONNECTEDCOMPONENTS(V, $C(e) \cup \{e\}$) $<$ NBCONNECTEDCOMPONENTS(V, $C(e)$)) **alors** $A \leftarrow A \cup \{e\}$ **fin si**

(10) **fin tant que**

(11) **retourner** A

fin GCKRUSKALORDINALMAX1

Fig. 5 Un algorithme résolvant le problème GC(ST/KRUSKAL(MAX, MAX)/MAX).

Comme pour KRUSKALORDINALMAX1, cet algorithme (GCKRUSKALORDINALMAX1) utilise NBCONNECTEDCOMPONENTS un algorithme de dénombrement des composantes connexes d'un graphe non orienté, et CHOISIR une stratégie de choix dans un ensemble en extension.

Exemple 3 : En faisant marcher GCKRUSKALORDINAL-MAX1 sur l'instance de l'Exemple 2 illustré à la Fig. 4, nous obtenons comme résultat l'ensemble d'arêtes $E \setminus \{b\}$.

Nous avons les résultats suivants :

Proposition 1. L'algorithme GCKRUSKALORDINALMAX1 :

(1) a une complexité en temps dans le pire cas qui est linéaire en la taille de son entrée $(G, \succcurlyeq_E)$.

(2) retourne uniquement les arêtes qui font partie d'un arbre couvrant maximal pour le problème ST/KRUSKAL(MAX, MAX)/MAX à partir d'une instance $((V, E), \succcurlyeq_E)$, si de telles solutions existent. Sinon retourne 'non'.

La logique sous-jacente à cet algorithme est de se placer dans le meilleur scénario de choix de l'arête e, quelque soit $e \in E$, pour l'algorithme Kruskal ordinal. Le meilleur scénario consiste à choisir e le plus tôt possible, en supprimant uniquement les arêtes qui l'empêchent d'être maximale dans un sous-ensemble de E. Or, pour que e soit maximale, il faut choisir et supprimer toutes les arêtes qui lui sont préférées ; et pour choisir chacune de ces arêtes, il faut qu'elles soient maximales, et ainsi de suite. Si dans ce meilleur cas, l'algorithme n'arrive pas à trouver un arbre couvrant maximal, et que e diminue le nombre de composantes connexes, alors e peut être choisie pour faire partie d'un arbre couvrant maximal pour une relation de préférence KRUSKAL(MAX, MAX). En effet, ce meilleur scénario peut alors être complété en choisissant itérativement n'importe quelle arête restante et maximale pour les arêtes non encore choisies de E.

5 Interactivité et Arbres couvrants maximaux

5.1. Utilisation interactive de la consistance qualitative globale

La consistance qualitative globale peut être utilisée dans un processus décisionnel interactif, où l'utilisateur fait des choix locaux, et où l'ordinateur se borne à supprimer les arêtes ne faisant pas partie d'un arbre couvrant maximal. Les choix locaux que nous allons permettre ici sont d'un seul type : la suppression d'arêtes. La consistance qualitative vue précédemment est parfaitement adaptée à ce type de choix local. Par contre, les choix locaux qui consistent à sélectionner une arête nécessitent la résolution d'un autre problème de consistance qualitative que nous n'abordons pas dans cet article.

Comme tout processus décisionnel, celui-ci peut admettre des retours arrières – des remises en question de suppression d'arêtes – pour effectuer une *recherche heuristique* de la solution de meilleure qualité. Pour cela, il est nécessaire d'effectuer un suivi des opérations effectuées pour un traitement adapté, comme il est d'ailleurs fait en programmation par contraintes. Nous ne nous étalerons pas sur la mise en œuvre des retours arrières dans un processus décisionnel.

Avant d'aller plus loin, voici une définition qui nous sera utile pour la suite : Étant donnée une instance $(G, \succcurlyeq_E)$ de ST/KRUSKAL(MAX, MAX)/MAX, une arête e de G est dite **critique** (ou **indispensable**) si elle est consistante pour $(G, \succcurlyeq_E)$ et qu'elle est présente dans <u>tout</u> arbre couvrant maximal.

Étudions les conséquences de suppressions d'arêtes sur des exemples :

Exemple 4 : Reprenons l'Exemple 2 et présentons deux scénarios possibles (Fig. 6), sachant que l'ensemble maximal $S_{ST/K/MAX}(G, \succcurlyeq_E) = \{acd, ach\}$. Le graphe G possède donc deux arêtes critiques a et c.

L'application de la consistance qualitative globale permet de supprimer l'arête b. Pour le premier scénario, l'utilisateur choisit de supprimer une arête non critique, h par exemple. Le graphe partiel obtenu est

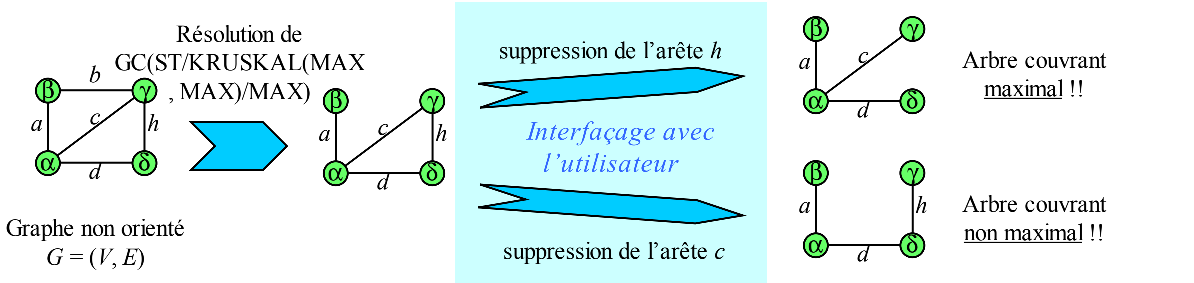

Fig. 6 Recherche de solutions pour DECISION SUPPORT(ST/KRUSKAL(MAX, MAX)/MAX) avec maintien de consistance qualitative globale

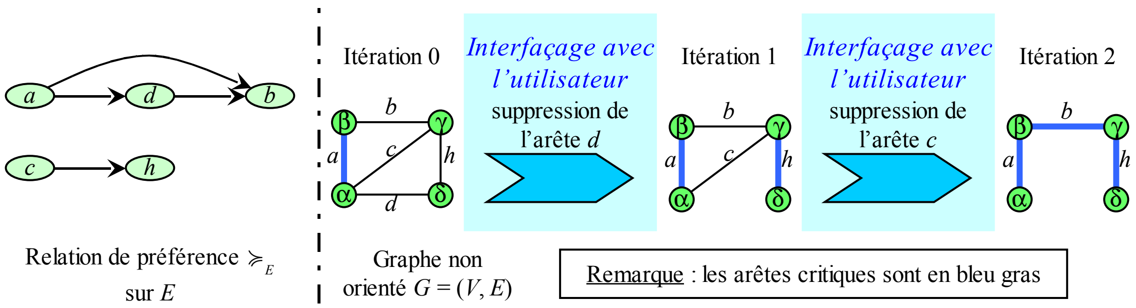

Fig. 7 Recherche[6] de solutions pour DECISION SUPPORT(ST/KRUSKAL(MAX, MAX)/MAX) avec identification d'arêtes critiques et maintien de consistance qualitative globale.

un arbre couvrant maximal de ($S_{\mathrm{ST}}(G)$, $\succcurlyeq$). Maintenant, si l'utilisateur choisit une arête critique (scénario 2), l'arbre couvrant obtenu n'est plus maximal dans ($S_{\mathrm{ST}}(G)$, $\succcurlyeq$) !

Cet exemple montre l'importance d'indiquer à l'utilisateur la présence d'arêtes indispensables pour qu'il ne cherche pas à les supprimer, et afin que ses choix mènent plus certainement à une solution maximale.

Exemple 5 : Considérons le même graphe non orienté de l'exemple précédent, ainsi que la relation de préférence illustrée à gauche de la Fig. 7.

Notons $G^{(k)}$ le graphe partiel de G obtenu à l'itération k. Itération 0 : Le graphe initial $G^{(0)} = G$ possède 3 arbres couvrants maximaux $S_{\mathrm{ST/K/MAX}}(G^{(0)}, \succcurlyeq_E) = \{abd, acd, ach\}$, et une arête critique a. Le maintien de la consistance qualitative ne supprime aucune arête, car chaque arête fait partie d'au moins un arbre couvrant maximal. Itération 1 : Après suppression de l'arête d par l'utilisateur, l'ensemble des arbres couvrants maximaux est modifié et non inclus dans celui du graphe précédent : $S_{\mathrm{ST/K/MAX}}(G^{(1)}, \succcurlyeq_E) = \{abh, ach\} \nsubseteq S_{\mathrm{ST/K/MAX}}(G^{(0)}, \succcurlyeq_E)$!!! Par contre, l'ensemble des arêtes critiques de l'itération 2 contient celui de l'itération 1 !!

En résumé, cet exemple montre que même lorsque l'utilisateur supprime une arête non critique pour le graphe partiel courant, la conservation de l'ensemble maximal n'est pas garantie, et on peut donc arriver au choix d'un arbre couvrant non maximal (itération 2) pour l'instance de départ ! Ce manque de *rationalité procédurale* arrive même dans le cas transitif, la relation $\succcurlyeq_E$ de notre exemple étant un préordre partiel !

En définitive, un processus de décision réactif et déductif, faisant usage de la suppression d'arêtes comme type de choix local par l'utilisateur, implique la mise en place de deux procédures techniques :

- maintenir la consistance qualitative globale : suppression des arêtes qui appartiennent à un arbre couvrant maximal qui ne peut plus être choisi car l'une de ces arêtes a été supprimée, mais aussi
- identifier les arêtes critiques, c'est-à-dire, celles présentes dans tous les arbres couvrants maximaux qui peuvent encore être choisis.

Le maintien de la consistance qualitative globale a été étudié au § 4, nous proposons un algorithme résolvant le problème d'arêtes critiques associé à Kruskal ordinal au § 5.2 suivant.

De plus, la suppression d'arêtes est une opération qui ne garantit pas, même avec un maintien de la consistance qualitative globale, l'obtention d'un arbre couvrant maximal pour l'instance de départ, ni d'identifier un sous-ensemble maximal tout au long du processus. Par conséquent, au § 5.3, nous étudions quelques propriétés que doivent satisfaire la relation $\succcurlyeq_E$ pour que le processus décisionnel aboutisse inévitablement à un arbre couvrant maximal pour l'instance de départ.

5.2. Identification des arêtes critiques

Le problème de l'identification des arêtes critiques pour les arbres couvrants maximaux d'un graphe G se formule comme suit :

CRITICAL EDGES(ST/KRUSKAL(MAX, MAX)/MAX) : Étant donnés un graphe non orienté $G = (V, E)$ et une relation de préférence P-acyclique $\succcurlyeq_E$ sur E induisant une relation binaire de préférence $\succcurlyeq$ sur $\mathcal{P}(E)$ qui est KRUSKAL(MAX, MAX) pour $(G, \succcurlyeq_E)$, retourner les arêtes de E qui font partie de <u>tout</u> arbre couvrant maximal pour $\succcurlyeq$, s'il en existe. Sinon retourner 'non'.

Notons $S_{\mathrm{CE(ST/K/MAX)}}(G, \succcurlyeq_E) \subseteq E$, l'ensemble des arêtes retourné par un algorithme résolvant le problème CRITICAL EDGES(ST/KRUSKAL(MAX, MAX)/MAX). Alors, par définition, nous avons l'égalité et l'inclusion suivantes :

$$S_{\mathrm{CE(ST/K/MAX)}}(G, \succcurlyeq_E) = \bigcap_{x \in S_{\mathrm{ST/G/MAX}}(G, \succcurlyeq_E)} x \subseteq S_{\mathrm{GC(ST/K/MAX)}}(G, \succcurlyeq_E). \quad (2)$$

Qu'en est-il de la résolution de ce problème ? Notons $\succcurlyeq_E^{1}$ l'inverse de $\succcurlyeq_E$, c'est-à-dire la relation binaire obtenue en inversant le sens des préférences de $\succcurlyeq_E$: $x \succcurlyeq_E^{1} y \Leftrightarrow y \succcurlyeq_E x$. En d'autres termes, la partie symétrique est inchangée , alors que la partie asymétrique change de sens :

$x \simeq_E^{1} y \Leftrightarrow y \simeq_E x$ et $x \succ_E^{1} y \Leftrightarrow y \succ_E x$.

L'algorithme de la Fig. 8 résout le problème précédent, et suppose l'existence d'au moins un arbre couvrant réalisable (⇔ existence d'au moins un arbre couvrant maximal d'après § 4.2).

Les différences fondamentales entre cet algorithme et son homologue dédié à la consistance qualitative globale (Fig. 5) portent sur deux points :

- l'utilisation de l'inverse de la relation de préférence sur E,
- le test en ligne 8 de ceKruskalOrdinalMax1 utilisant l'ensemble complémentaire du test en ligne 9 de gcKruskalOrdinalMax1.

Proposition 2. L'algorithme ceKruskalOrdinalMax1 :

(1) a une complexité en temps dans le pire cas qui est linéaire en la taille de son entrée ;

(2) résout le problème CRITICAL EDGES(ST/KRUSKAL(MAX, MAX)/MAX).

La logique sous-jacente à l'algorithme d'identification des arêtes critiques est de se placer dans le pire scénario de choix de l'arête e, pour tout $e \in E$. Le pire scénario pour l'algorithme Kruskal ordinal, est de choisir e au dernier moment, lorsqu'il n'y a vraiment plus aucun autre choix, c'est-à-dire la choisir après avoir choisi toutes les arêtes qui ne lui sont pas

inférieures directement ou indirectement (arêtes f pour lesquelles il n'existe pas de chemin formé de préférences strictes de e vers f).
Si dans ce pire cas, l'algorithme Kruskal ordinal n'arrive pas à trouver un arbre couvrant maximal, et que e diminue le nombre de composantes connexes, alors e est un passage obligatoire pour obtenir une solution maximale.

---

ceKruskalOrdinalMax1(G = (V, E) : graphe non orienté, $\succcurlyeq_E$: rbp P-acyclique sur E) : **retourner** {ensemble d'arêtes, non}

% Précondition : $S_{ST}(G) \neq \varnothing$.

début

(1) $A \subseteq E \leftarrow \varnothing$

(2) $B \subseteq E \leftarrow E$

(3) $\underline{C}(e) \subseteq E \leftarrow \varnothing$, pour tout $e \in E$

(4) **tant que** ($B \neq \varnothing$) **faire**

% invariants de boucle : (V, A) est acyclique, $A \cap B = \varnothing$ et $B \cap \underline{C}(e) = \varnothing$

(5) $e \leftarrow$ choisir($M(B, \succcurlyeq_E^1)$)

(6) $B \leftarrow B \setminus \{e\}$

(7) $\underline{C}(e) \leftarrow \left(\bigcup_{e' \succ_E^1 e} \underline{C}(e') \right)$

(8) **si** (nbConnectedComponents(V, $E \setminus \underline{C}(e)$) > nbConnectedComponents(V, $E \setminus (\underline{C}(e) \cup \{e\})$)) **alors** $A \leftarrow A \cup \{e\}$ **fin si**

(9) fin tant que

(10) **si** ($A = \varnothing$) **alors** **retourner** non **sinon** **retourner** A **fin si**

fin ceKruskalOrdinalMax1

Fig. 8 Un algorithme résolvant le problème CRITICAL EDGES(ST/KRUSKAL(MAX, MAX)/MAX).

5.3. Choix rationnel et arbres couvrants maximaux

Nous allons maintenant essayer de mieux comprendre pourquoi les processus décisionnels basés sur la consistance qualitative globale, et la suppression d'arêtes non critiques, n'aboutissent pas toujours à une solution maximale pour l'instance de départ (illustré par l'Exemple 5). Pour cela, nous nous sommes tournés vers les travaux effectués en théorie du choix social [26], et plus spécifiquement en théorie du choix rationnel et des fonctions de choix [32, 26, 6].

Quatre axiomes de choix rationnel vont particulièrement canaliser notre attention. Formulons-les en utilisant la maximalité. Pour cela, considérons un ensemble fini S muni d'une relation binaire de préférence $\succcurlyeq$. Alors,

- **Axiome de choix d'Arrow** [32, p. 25] (ACA): Pour tout $A, B \in \mathcal{P}(S)$ et $A \subseteq B$, si $M(B, \succcurlyeq) \cap A \neq \varnothing$ alors $M(A, \succcurlyeq) = M(B, \succcurlyeq) \cap A$. (Si certains éléments maximaux d'un plus grand ensemble sont disponibles, alors seuls ces éléments seront maximaux dans le plus petit ensemble)
- **Axiome de Chernoff** (AC): Pour tout A, $B \in \mathcal{P}(S)$ et $A \subseteq B$, si $M(B, \succcurlyeq) \cap A \neq \varnothing$ alors $M(B, \succcurlyeq) \cap A \subseteq M(A, \succcurlyeq)$. (Tout ce qui n'est pas maximal dans un plus petit ensemble ne sera pas aussi maximal dans le plus grand)
- **Axiome de dominance forte** [6] ou **axiome dual de Chernoff** [32] (ADC): Pour tout A, $B \in \mathcal{P}(S)$ et $A \subseteq B$, si $M(B, \succcurlyeq) \cap A \neq \varnothing$ alors $M(A, \succcurlyeq) \subseteq M(B, \succcurlyeq) \cap A$. (Si des éléments maximaux d'un plus grand ensemble sont disponibles, alors tout ce qui n'est pas maximal dans le plus grand ensemble ne sera pas maximal dans le plus petit)
- **Axiome du sur-ensemble** (ASE): Pour tout A, $B \in \mathcal{P}(S)$ et $A \subseteq B$, si $M(B, \succcurlyeq) \subseteq M(A, \succcurlyeq)$ alors $M(A, \succcurlyeq) = M(B, \succcurlyeq)$. (Si tous les éléments maximaux d'un plus grand ensemble sont disponibles, alors eux et eux seuls sont maximaux dans le plus petit)

L'axiome ACA, correspond à une forme de monotonie procédurale. Nous dirons qu'un processus satisfaisant ACA est **pleinement rationnel**. Ces axiomes sont reliés par les relations logiques suivantes :

Théorème 1. (a) ACA $\Leftrightarrow$ AC et ADC
(b) Transitivité et Complétude $\Leftrightarrow$ ACA
(c) Quasi-transitivité $\Rightarrow$ AC et ASE
(d) P-acyclicité $\Rightarrow$ AC

Que devraient satisfaire les instances de notre problème pour qu'un processus décisionnel aboutisse toujours à un arbre couvrant maximal ? En s'inspirant des axiomes ci-dessus, nous voudrions que la procédure interactive satisfasse l'axiome ACA. Un processus qui aboutirait toujours à une solution maximale pour l'instance de départ peut vérifier un axiome moins contraignant comme ADC.
Réécrivons ces axiomes dans notre contexte particulier d'arbres couvrants, en fixant sans perte de généralité[5] le plus grand ensemble B à $S_{ST}(G) = S$, son ensemble maximal étant $S_{ST/K/MAX}(G, \succcurlyeq_E)$ dans le cadre du problème ST/KRUSKAL(MAX, MAX)/MAX. Remarquons que dans le type de processus décisionnels étudié, le plus petit ensemble A ne correspond pas à n'importe quel sous-ensemble de $S_{ST}(G)$, mais aux sous-ensembles obtenus en supprimant les arbres contenant un ensemble d'arêtes donné. Ainsi, nous obtenons :

Considérons une instance $(G, \succcurlyeq_E)$ de ST/KRUSKAL(MAX, MAX)/MAX ; et pour $\varnothing \neq A \subseteq E$ notons : $\mathcal{A} = \{x \in S_{ST}(G)$ et $x \subseteq A\}$. Alors un processus interactif vérifie :

(ACA) $\Leftrightarrow$ $\forall A \subseteq E,\ \mathcal{A} \cap S_{ST/K/MAX}(G, \succcurlyeq_E) \neq \varnothing \Rightarrow \mathcal{A} \cap S_{ST/K/MAX}(G, \succcurlyeq_E) = S_{ST/K/MAX}((V, A), \succcurlyeq_E)$.

(ACG) $\Leftrightarrow$ $\forall A \subseteq E,\ \mathcal{A} \cap S_{ST/K/MAX}(G, \succcurlyeq_E) \neq \varnothing \Rightarrow \mathcal{A} \cap S_{ST/K/MAX}(G, \succcurlyeq_E) \subseteq S_{ST/K/MAX}((V, A), \succcurlyeq_E)$.

(ADCG) $\Leftrightarrow$ $\forall A \subseteq E,\ \mathcal{A} \cap S_{ST/K/MAX}(G, \succcurlyeq_E) \neq \varnothing$

---

[5] Remarquons cependant qu'en fixant $B = S$, les axiomes AC et ADC réécrits sont plus généraux que les originaux ! C'est pourquoi nous ajoutons la lettre G mis pour Généralisé.

$\Rightarrow\ S_{\text{ST/K/MAX}}((V, A), \succcurlyeq_E) \subseteq \mathcal{A} \cap S_{\text{ST/K/MAX}}(G, \succcurlyeq_E)$.

(ASE) $\Leftrightarrow\ \forall A, B \subseteq E$ et $A \subseteq B$,

$S_{\text{ST/K/MAX}}((V, B), \succcurlyeq_E) \subseteq S_{\text{ST/K/MAX}}((V, A), \succcurlyeq_E)$

$\Rightarrow\ S_{\text{ST/K/MAX}}((V, B), \succcurlyeq_E) = S_{\text{ST/K/MAX}}((V, A), \succcurlyeq_E)$.

Maintenant, intéressons-nous aux conséquences des propriétés de la relation de préférence $\succcurlyeq_E$ sur les dépendances entre ensembles maximaux intervenant dans le processus décisionnel. Les trois dernières assertions du Théorème 1 permettent d'obtenir la proposition suivante :

Proposition 3. Étant donnée une instance $(G, \succcurlyeq_E)$ de ST/KRUSKAL(MAX, MAX)/MAX, alors les assertions suivantes sont vraies :

(*a*) Tout processus décisionnel vérifie l'axiome de Chernoff généralisé (ACG)

(*b*) Si $\succcurlyeq_E$ est un préordre complet, alors le processus est pleinement rationnel ($\Leftrightarrow$ vérifie ACA)

Étudions maintenant les conséquences du type de processus de décision utilisé – basé sur la suppression d'arêtes et la consistance globale –, sur l'identification d'un arbre couvrant maximal. Pour cela, introduisons la notion de fermeture transitive forte d'une relation binaire [26, 14] :

Étant donnée une relation binaire quelconque $\succcurlyeq$ sur un ensemble S, la **fermeture transitive forte** associée à $(S, \succcurlyeq)$ est la relation binaire $\tau_{\text{better}}(\succcurlyeq)$ définie comme suit :

Pour tout $x, y \in S$ et $x \neq y$, $\quad x\ \tau_{\text{better}}(\succcurlyeq)\ y \Leftrightarrow$

$$\begin{cases} \exists \text{ une sequence } x \succ z_1, \cdots, z_k \succ y \\ \text{avec } z_i \in S, \quad \text{pour tout } 1 \le i \le k \end{cases} . \tag{3}$$

En d'autres termes, c'est la fermeture transitive de la partie asymétrique de $\succcurlyeq$. La proposition suivante résume l'une des propriétés de ST/KRUSKAL(MAX, MAX)/MAX :

Proposition 4. Étant donnés un graphe non orienté $G = (V, E)$ et une relation de préférence P-acyclique $\succcurlyeq_E$ sur E, les deux instances $(G, \succcurlyeq_E)$ et $(G, \tau_{\text{better}}(\succcurlyeq_E))$ ont le même ensemble de solutions pour le problème ST/KRUSKAL(MAX, MAX)/MAX :

$$S_{\text{ST/K/MAX}}(G, \succcurlyeq_E) = S_{\text{ST/K/MAX}}(G, \tau_{\text{better}}(\succcurlyeq_E))$$

La consistance globale qualitative associée au problème ST/KRUSKAL(MAX, MAX)/MAX conserve l'ensemble maximal. Par contre, la suppression d'arêtes, ne garantit pas le respect de l'axiome d'Arrow, mais juste l'axiome de Chernoff. La première affirmation est confirmée par la proposition suivante, tandis que la seconde a été illustrée par l'Exemple 5 et la Proposition 3 (*a*) :

Proposition 5. Étant donnés un graphe non orienté $G = (V, E)$ et une relation binaire $\succcurlyeq_E$ P-acyclique sur E, nous avons l'égalité suivante :

$S_{\text{ST/K/MAX}}(G, \succcurlyeq_E) = S_{\text{ST/K/MAX}}((V, S_{\text{GC(ST/K/MAX)}}(G, \succcurlyeq_E)), \succcurlyeq_E)$

La Proposition 5 peut se généraliser sans difficulté à tout niveau de consistance effectuant une approximation de l'ensemble des arêtes contenues dans au moins un arbre couvrant maximal. Nous obtenons alors la proposition suivante, dont la démonstration est similaire à celle de la Proposition 4 :

Corollaire 1. Étant donnés un graphe non orienté $G = (V, E)$ et une relation binaire $\succcurlyeq_E$ P-acyclique sur E, nous avons l'égalité suivante :

$S_{\text{ST/K/MAX}}(G, \succcurlyeq_E) = S_{\text{ST/K/MAX}}((V, A), \succcurlyeq_E)$, pour tout $A \subseteq E$ et $S_{\text{GC(ST/K/MAX)}}(G, \succcurlyeq_E) \subseteq A$

En résumé, ce sont les opérations de restriction de l'ensemble maximal (suppression d'arêtes consistantes pour $(G, \succcurlyeq_E)$ et non critiques) qui empêchent à ces processus de respecter l'axiome de choix d'Arrow. Ces opérations étant réalisées par l'utilisateur, il serait intéressant de redéfinir la notion de consistance qualitative et d'identification d'arêtes critiques en y intégrant des éléments du processus décisionnel, de manière à garantir une pleine rationalité, et ce quelque soit l'instance du problème initial.

6 Conclusions et Perspectives

L'une des limites des processus décisionnels basés sur l'énumération qu'avaient proposé Perny & Spanjaard [19] pour la résolution des problèmes combinatoires ordinaux était l'intractabilité des données de grande taille. Nous avons proposé des problèmes automatisables, la consistance qualitative et l'identification d'information indispensable ($\Leftrightarrow$ critique), dont les résultats peuvent être traités en temps réel par un être humain (i.e. linéaire en la taille de l'instance de départ). Ces problèmes automatisables sont basés à la fois sur la notion de consistance mise en avant par la PPC, et celle de choix étudiée en Aide à la Décision. Dans le cadre des arbres couvrants maximaux basés sur l'algorithme de Kruskal ordinal, nous avons proposé des algorithmes de résolution de ces problèmes automatisables ayant une complexité en temps dans le pire cas linéaire en la taille des données.

En contre partie, utilisés dans un processus décisionnel, ces problèmes automatisables ne permettent pas d'aboutir inévitablement à une solution maximale. Par conséquent, dans le cadre des arbres maximaux, nous avons identifié certaines propriétés, que doivent vérifier la représentation compacte des préférences utilisée, pour y aboutir, et ce en s'aidant de la théorie du choix rationnel. La consistance qualitative et la recherche d'une solution préférée ne sont pas deux problématiques automatisables exclusives dans un SIAD. Les deux peuvent être combinées pour améliorer, enrichir la recherche de solution préférée adéquate pour l'utilisateur.

Comme perspectives, l'un des buts de cet article est de rapprocher les communautés de l'IA, de la AD-RO et du Choix Social (comme appelaient à le faire [5]), pour traiter plus efficacement les problèmes combinatoires exploitant des préférences complexes. Nous pensons que les apports mutuels ouvriront la voie à une nouvelle ère de résolution interactive de problèmes complexes.

De manière plus immédiate, notre objectif est de nous ouvrir sur d'autres problèmes combinatoires exploitant des préférences non conventionnelles ; et de concevoir des SIAD, utilisant des algorithmes de consistance qualitative et d'identification d'information indispensable, à destination des professionnels.

Références

1. S. Barberà, W. Bossert, P. K. Pattanaik (2001), "Ranking sets of objects", Cahiers de recherche n°2001-02, Université de Montréal, Département de Sciences Économiques.
2. C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, D. Poole, "Preference-based constrained optimization with cp-nets", Computational Intelligence 20: 137-157.
3. R.E. Burkard, R.A. Cuninghame-Green, U. Zimmermann (eds.) (1984), "Algebraic and combinatorial methods in operations research", Annals of discrete mathematics 19, North Holland, Amsterdam.
4. P.M. Camerini, G. Galbiati, and F. Maffioli (1984) "The complexity of multi-constrained spanning tree problems", *in* Theory of Algorithms, Colloquium Pecs 1984: 53-101.
5. J. C. Courbon, D. Dubois, J.-C. Pomerol, B. Roy (1994), "Autour de l'aide à la décision et de l'intelligence artificielle", Research report Laforia/IBP 1994/01, Institut Blaise Pascal, Université Paris 6, Paris, France, 25 p.
6. R. Deb (2007), "Non-binary social choice", *in* K. J. Arrow, A. K. Sen, K. Suzumura (eds.), "Handbook of social choice and welfare", Vol. 2, chapter 18, Elsevier, North-Holland, Amsterdam.
7. G. Dooms, I. Katriel (2006), "Graph constraints in constraint programming: Weighted spanning trees", INGI research report 2006-01, UCL, Belgique.
8. G. Dooms, I. Katriel (2006), "The minimum spanning tree constraint", 12th International Conference on Prrinciples and Practice of Constraint Programming, Nantes, France
9. M. Ehrgott, X. Gandibleux (2000), An annotated bibliography of multiobjective combinatorial optimization, OR Spektrum 22, 2000 : 425-460.
10. J. Ghoshal, R. Laskar and D. Pillone, Topics on domination in directed graphs, in: T. Haynes, S. Hedetniemi and P. J. Slater, Domination in graphs. Advanced topics, coll. Monographs and textbooks in pure and applied mathematics, vol. 209 (Marcel Dekker, New York, 1998) chapitre 15: 401-437.
11. H.W. Hamacher and G. Ruhe (1994), "On spanning tree problems with multiple objectives", Annals of Operations Research 52: 209-230.
12. A. Jaszkiewicz (2001), "Multiple objective metaheuristic algorithms for combinatorial optimization", Thèse d'Habilitation n°330, Université de Technologie de Poznań, Poznań, Pologne.
13. R.-R. Joseph, P. Chan, M. Hiroux, G. Weil, Decision-support with preference constraints, European Journal of Operation Research 177 (2007) 1469-1494.
14. R.-R. Joseph (2005), "Choosing with a binary relation: Relative choice axioms and transitive closures", *submitted to* Discrete Applied Mathematics: 26 p.
15. U. Junker (2006), "Preference-based problem solving for constraint programming", *in* G. Bosi, R. I. Brafman, J. Chomicki, W. Kießling (eds.), Preferences 2004: Specification, inference, applications, Dagstuhl seminar proceedings, Schloss Dagstuhl, Allemagne.
16. P. Lévine, J.-C. Pomerol (1989), "Systèmes interactifs d'aide à la décision et systèmes experts" coll. Traité des nouvelles technologies, série décision assistée par ordinateur, Hermès, Paris.
17. A. Newell, H. A. Simon, Human problem solving, Englewood Cliffs, NJ, Prentice-Hall, 1972.
18. P. Perny, O. Spanjaard (2002), Preference-based search in state space graphs, Proceedings of the AAAI'2002 Conference, Edmonton, Canada: 751-756.
19. P. Perny, O. Spanjaard, "A Preference-based Approach to Spanning Trees and Shortest Paths Problems", European Journal of Operational Research 162, (2005) 584–601.
20. A. Roth, M. Sotomayor, "The two-sided matching", Cambridge University Press, Cambridge, 1990.
21. B. Roy (1985), "Méthodologie multicritère d'aide à la décision", Economica, Paris.
22. B. Roy (1998), "A missing link in OR-DA: Robustness analysis", Foundations of Computing and Decision Sciences 23 (3): 141-160.
23. B. Roy, D. Bouyssou (1993), "Aide multicritère à la décision: Méthodes et cas", Economica, coll. « Gestion », Paris.
24. A. Schrijver (2003), "Combinatorial optimization: Polyhedra and efficiency", Series in algorithms and combinatorics vol. 24, Springer Verlag Publication, Chichester.
25. A. K. Sen (1970), "Collective choice and social welfare", coll. Advanced textbooks in economics, vol. 11, Elsevier Science Publishers, Pays-Bas.
26. A. K. Sen (1986), "Social choice theory", chap. 22 in Arrow K. J., Intriligator M. D. (eds.), Handbook of mathematical economics, vol. 3, North-Holland, Elsevier Science Publishers B. V.: 1073-1181.
27. H. A. Simon, Models of man, New York, NY, Wiley, 1957.
28. H. A. Simon (1976), "From substantive to procedural rationality", *in* S.J. Latsis (ed.), *Methods and appraisal in economics*, Cambridge University Press, New-York: 129-148.
29. O. Spanjaard (2003), Exploitation de préférences non-classiques dans les problèmes combinatoires: modèles et algorithmes pour les graphes, PhD Thesis, Université Paris-Dauphine, France, 16 Décembre 2003.
30. R. E. Steuer (1986), "Multiple criteria optimization: Theory, computation and application", Wiley, New-York.
31. B. Subiza and J. E. Peris, Choice functions: Rationality re-examined, Theory and Decision 48 (2000) 287-304.
32. K. Suzumura (1983), "Rational choice, collective decisions and social welfare", Cambridge University Press, Cambridge.
33. C. Unsworth, P. Prosser (2006), "Rooted tree and spanning tree constraints", 17th ECAI Workshop on Modelling and Solving Problems with Constraints.
34. G. Verfaillie, N. Jussien (2005), "Constraint solving in uncertain and dynamic environnements – a survey", Constraints 10 (3): 253-281.