Contraintes d'optimisation pour les problèmes d'arbres couvrants sous restrictions

Jérôme Brongniart, Dhaenens Clarisse, El-Ghazali Talbi

Contraintes d'optimisation pour les problèmes d'arbres couvrants sous restrictions

Brongniart Jérôme, Dhaenens Clarisse et El-Ghazali Talbi

LIFL INRIA/Université de Lille 1, Bat.M3 59655, Villeneuve d'Ascq Cedex, France
{Jerome.Brongniart, Clarisse.Dhaenens, El-Ghazali.Talbi}@lifl.fr

Résumé

Bien que des méthodes de filtrage basées sur les coûts existent depuis peu pour les problèmes d'arbres couvrants sous incertitudes [2, 5], elles ne permettent pas la prise en compte de restrictions additionnelles. Nous présentons dans cet article une contrainte d'optimisation pour les problèmes d'arbres couvrants de poids optimal sous restrictions fondée sur les travaux de [13].

Nous améliorons tout d'abord la contrainte d'optimisation proposée par [2, 5] pour les problèmes d'arbres couvrants en mettant en évidence l'analogie entre recherche de cycles, recherche de ponts et calcul de remplaçants. Puis nous décrivons les problèmes posés par l'ajout de restrictions lors de l'utilisation de ce type d'algorithme pour proposer ensuite, grâce aux travaux de Sellmann, une nouvelle contrainte d'optimisation basée sur une relaxation Lagrangienne.

Nous fournissons une première expérimentation de cette contrainte sur un problème d'arbre couvrant sous restriction de degré.

Abstract

Cost-based filtering algorithms already exist [2, 5] for robust spanning tree problems, but they can not handle additionnal restrictions. This article describes a cost-based filtering algorithm, founded on the work of [13], for optimal constrained spanning tree problems.

In a first step, we improve the algorithm of [2, 5] for the optimal spanning tree problem using the similarity between cycles detection, bridges detection and replacements computation. After that, we use the work of Sellmann to define the difficulty of tacking into account additionnal restrictions for this kind of algorithms. Due to this definition, we propose a Lagrangian-based filtering algorithms for constrained spanning tree problem.

We conclude by testing this algorithm on the degree-constrained spanning tree problem.

1 Introduction

L'ajout de restrictions additionnelles[1] à un problème d'arbre couvrant rend la recherche de l'arbre de coût minimal $\mathcal{NP}$-difficile. De plus, les contraintes d'optimisation existantes ne peuvent prendre directement en compte les restrictions sur la forme de l'arbre.

Nous adapterons dans cet article la méthode proposée par Sellmann pour résoudre des problèmes d'arbres couvrants sous restrictions grâce à une méthode hybridant programmation par contraintes et relaxation Lagrangienne.

1.1 Contexte

Le succès de la résolution de problèmes d'optimisation par programmation par contraintes vient en grande partie de l'hybridation de ces méthodes avec des algorithmes provenant de la recherche opérationnelle.

En effet, on peut observer une complémentarité entre la programmation par contraintes, raisonnant en terme de faisabilité, et les méthodes de recherche opérationnelle, se basant sur la notion d'optimalité. Ne considérer que la faisabilité d'une affectation ne permet pas de capturer les variations de coût comme le font les algorithmes de recherche opérationnelle. Cependant, l'utilisation des informations fournies par les algorithmes de recherche opérationnelle, associée à des méthodes de filtrage complexes, permet de réduire considérablement le temps d'exécution en améliorant l'élagage de l'arbre de recherche.

La définition de contraintes globales incluant des algorithmes de recherche opérationnelle est la réponse la

[1] Par soucis de clarté, nous appellerons les contraintes, au sens programmation linéaire, des "restrictions". Le terme "contrainte" désignera les algorithmes de filtrage.

plus courante à ce problème d'intégration. L'existence de structures simples et très étudiées, comme le plus court chemin ou l'arbre couvrant de poids optimal, permet de calculer en temps polynomial des informations de variation de coût. Cependant, l'inconvénient majeur de ces méthodes est qu'elles ne permettent pas de prendre en compte l'ajout de restrictions additionnelles et ne fournissent donc que des informations partielles en cas de résolution de problèmes avec restrictions.

L'utilisation de la relaxation Lagrangienne [13] a permis de répondre à cette difficulté en offrant la possibilité d'améliorer les algorithmes de filtrage grâce aux informations duales fournies par les coefficients de Lagrange. Cette méthode a été appliquée avec succès à des problèmes tels que le plus court chemin avec restrictions de capacité [8] et nous allons, dans cet article, l'appliquer aux problèmes d'arbres couvrants de poids optimal sous restrictions.

1.2 Démarche

Nous présenterons dans la première partie de cet article (section 2) les deux types d'information de variation des coûts traditionnellement utilisées lors de la création d'une contrainte d'optimisation. En effet, bien que les méthodes dites "duales", comme le simplexe, sont très souvent employées, l'existence d'algorithmes de calcul de déviations [10] permet d'exploiter des informations structurelles, dite "primales".

Cela nous permettra d'introduire (section 3) la contrainte d'optimisation primale existante pour le problème de l'arbre couvrant de poids optimal, définies dans [2, 5]. Après avoir rappelé le concept de calcul des remplaçants, nous proposerons une simplification de l'algorithme de filtrage grâce à la correspondance entre recherche de cycles, recherche de ponts et calcul des remplaçants.

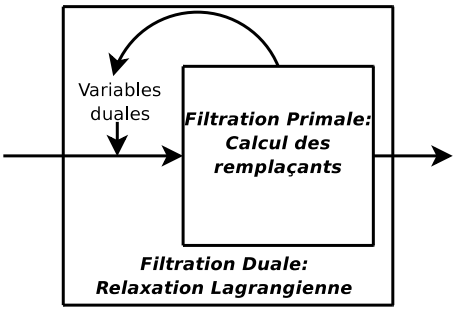

FIG. 1 – Schéma général de l'algorithme de filtrage.

Nous verrons par la suite (section 4) les problèmes que pose l'ajout de restrictions supplémentaires au problème initial lors de l'utilisation d'un algorithme primal. Nous rappellerons la solution apportée par Sellmann [13], appliquée avec succès sur un problème de plus court chemin sous restrictions [12], et l'appliquerons aux problèmes d'arbres couvrants de poids optimal en décrivant la procédure de filtrage retenue, schématisée figure 1. Nous justifierons cette approche par une première expérimentation (section 5), appliquée aux problèmes des arbres couvrants sous restriction de degré.

2 Valuation des coûts de changement en programmation par contraintes

Selon la terminologie de [9] et l'appellation usuelle de la recherche opérationnelle, les informations évaluant la variation des coûts recensées ici utilisent des relaxations, dans le sens où elles se fondent sur une simplification du problème original, souvent $\mathcal{NP}$-difficile. Ces techniques de recherche opérationnelle fournissent une estimation de la borne de la fonction objectif et de l'évolution de celle-ci en cas de modification d'une variable. Ces informations sont minorantes pour pouvoir garantir la complètude de la recherche.

Il est possible de classer les relaxations utilisées pour la création de contraintes globales d'optimisation en deux types dépendant des informations utilisées pour filtrer le domaine de ces variables.

2.1 Relaxations duales

Les relaxations les plus utilisées pour mesurer la variation des coûts se fondent sur la théorie de la dualité en utilisant des algorithmes tels que le simplexe, les méthodes de points intérieurs ou les méthodes de décompositions. Ce genre de méthodes simplifie en général le problème original grâce au relâchement des restrictions d'égalité ou grâce à une résolution itérative de sous-problèmes polynomiaux.

Elles évaluent, avec l'aide des variables du problème dual, les variations du coût de la solution dûes aux restrictions du problème. La théorie de la dualité certifie que ces méthodes offrent des informations minorantes, permettant ainsi de garantir la complètude.

2.2 Relaxations primales

Il est possible pour quelques problèmes, tels que le plus court chemin ou l'arbre de poids optimal, de calculer en temps polynomial le coût minimum nécessaire pour rétablir la structure en cas de modification des données du problème, appelé coût de remplacement par [10].

En prenant l'exemple d'un problème de plus court chemin, la suppression d'une arête appartenant à la solution optimale nécessite le re-calcul du nouveau plus court chemin. La différence entre l'ancienne et la nouvelle solution donne le coût minimal nécessaire pour retirer cette arête de l'ensemble des arêtes potentielles.

Pour les relaxations utilisant ce type d'information, la complètude est garantie par le fait que les coûts de remplacement sont calculés grâce à une solution minorante.

3 Relaxation primale pour les arbres couvrants pondérés

Le problème de l'arbre couvrant de poids optimal consiste à trouver un arbre[2] reliant tous les nœuds du graphe et dont la somme du coût des arêtes est minimale ou maximale. Ce problème, très étudié, peut être résolu en temps polynomial grâce à des algorithmes tels que celui de Kruskal. La structure simple de ce problème permet le calcul d'informations primales à partir desquelles ont été définies des contraintes d'optimisation [2, 5].

Dans cette section, nous présenterons puis simplifierons ces contraintes grâce au principe de calcul des remplaçants, à l'origine des informations primales.

Nous utiliserons dans la suite de cet article, les conventions suivantes :

$G = \{V, E, cost\}$ est le graphe pour lequel on cherche à trouver un arbre de poids optimal. V désigne l'ensemble des nœuds, E l'ensemble des arêtes et $cost : E \rightarrow R^+$ la fonction de coût. T est l'arbre couvrant non contraint de poids minimal de G et $x_{(i,j)}$ est la variable binaire désignant si l'arête (i, j) est présente ou non dans la solution optimale. Nous ne nous intéresserons qu'à un problème de minimisation.

3.1 Calcul des remplaçants

Le calcul des remplaçants a été défini pour la première fois par [6] pour calculer les k plus petits arbres couvrants. Ce calcul peut être effectué en temps pseudo-linéaire grâce à des algorithmes tels que [14].

Ce calcul se base sur la définition structurelle d'un arbre, à savoir un graphe ne possédant qu'un seul chemin entre chaque paire de nœuds. Gabow introduit dans son article la notion d'échange admissible (e, f) où $e \in T$ et $f = (i, j) \notin T$ tels que e appartienne au cycle formé dans T par f. Pour toute arête e, les échanges admissibles (e, f) définissent l'ensemble des arêtes f permettant la re-connections de l'arbre en cas de retrait de e. De même, pour toute arête f, les échange admissibles (e, f) définissent l'ensemble des arêtes e permettant d'éliminer le cycle formé par l'ajout de f. Cela a permis à Gabow de définir deux théorèmes permettant l'évaluation du coût minimal de remplacement d'une arête.

Théoreme 3.1.1 *Pour toute arête $e = (i, j) \in T$, si $x_{(i,j)} = 0$, alors le coût minimal d'un arbre de $G - e$ est de $cost(T) - cost(e) + cost(f)$, où f est l'arête de coût minimal telle que (e, f) soit un échange admissible.*

Théoreme 3.1.2 *Pour toute arête $f = (i, j) \notin T$, si $x_{(i,j)} = 1$, le coût minimal d'un arbre de G contenant f est de $cost(T) - cost(e) + cost(f)$, où e est l'arête de coût maximal telle que (e, f) soit un échange admissible.*

Le théorème 3.1.1 peut être illustré par la figure 2. Lors du retrait d'une arête e, en pointillé noir, les seules arêtes f permettant de reconnecter l'arbre appartiennent à la coupe formée par $T - e$ et définissent un échange admissible (e, f). La différence entre le coût de e et celui de la plus petite arête de cette coupe définie le coût de remplacement de l'arête e. Pour cette figure, ce coût est de 1.

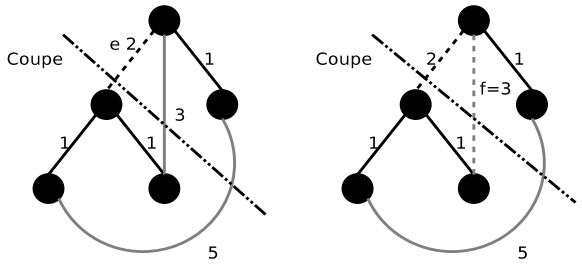

FIG. 2 – Illustration du théorème 3.1.1.

De même, le théorème 3.1.2, illustré figure 3, indique que, lors de l'ajout dans l'arbre de l'arête $f = (i, j)$ en pointillé gris, l'élévation minimum du coût passe par le retrait de l'arête $e \neq f$ de coût maximum appartenant au cycle formé par l'ajout de f, en pointillé noir. Pour cette figure, ce coût est de 3.

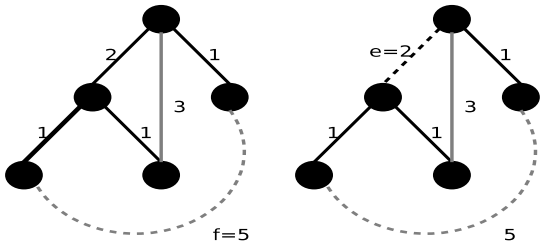

FIG. 3 – Illustration du théorème 3.1.2.

[2] C'est à dire un sous-graphe acyclique et connexe.

3.2 Travaux existants

Comme spécifié en introduction, deux travaux se sont basés sur cette notion pour définir des méthodes de filtrage pour un problème d'arbre couvrant sous incertitude. Ils traitent de problèmes de robustesse, où le coût d'une arête est définie par un intervalle mesurant les coûts probables de cette arête. Dans [5], les graphes à partir desquels sont calculés les arbres peuvent également être variables. La méthode proposée dans cet article offre la possibilité de définir des nœuds variables. Nous ne considérerons pas ce cas particulier par la suite, les auteurs ayant démontré qu'assurer la consistance de borne est $\mathcal{NP}$-difficile, par réduction à un problème d'arbre de Steiner.

Ces algorithmes [2, 5] utilisent des bornes majorantes pour filtrer les variables des domaines. Étant donnés un arbre minorant T et une solution majorante T', si le coût d'échange d'une arête $e \in T$ entraîne un arbre minorant $T-e$ de coût supérieur à T', alors cette arête appartient à la solution optimale[3].

Ces deux travaux appliquent, avant d'effectuer le filtrage par coût de remplacement, un algorithme de recherche de cycles et de ponts.

Un pont est une arête obligatoirement présente dans toute solution car son retrait entraîne une partition du graphe. De plus, une arête $f = (i, j) \notin T$ est obligatoirement absente de toute solution si elle forme un cycle, c'est à dire si, pour toutes les arêtes e du chemin entre i et j de T, $x_{(i,j)} = 1$. La recherche de ces cycles et ces ponts peut s'effectuer en temps linéaire grâce notamment à des algorithmes de recherche en profondeur d'abord.

On peut résumer l'algorithme de filtrage de ces articles de la manière suivante :

1. Recherche des cycles et des ponts.
2. Retirer de l'ensemble des arêtes potentielles chaque arête formant un cycle.
3. Ajouter aux arêtes obligatoires chaque arête formant un pont.
4. Calculer les coûts de remplacement.
5. Affecter les arêtes dont le coût de remplacement entraîne un dépassement de borne.

3.3 Amélioration proposée

3.3.1 Cycles, ponts et remplaçants

La notion de ponts et de cycles est très semblable à la notion d'échanges admissibles utilisée pour le calcul des coûts de remplacement et il est possible de simplifier l'algorithme de filtrage grâce aux deux remarques que nous formulons ci-dessous :

Remarque 3.3.1 *Si, pour toute arête (i, j) telle que $x_{(i,j)} = 1$, $cost(i, j) = -\infty$, alors une arête f forme un cycle si et seulement si elle a pour remplaçant une arête de coût $-\infty$.*

Remarque 3.3.2 *Si, pour toute arête (i, j) telle que $x_{(i,j)} = 0$, $cost(i, j) = \infty$, alors une arête e est un pont si et seulement si elle a pour remplaçant une arête de coût ∞.*

Ces remarques, illustrées figure 4, se démontrent assez naturellement :

Si, pour une arête $f = (i, j) \notin T$, il existe sur le chemin entre i et j de T une arête non affectée, cette arête aura un coût supérieur au coût des arêtes affectées et sera donc désignée comme remplaçant pour f. Par contre, si toutes les arêtes sur ce chemin sont affectées, le remplaçant de f aura un coût de $-\infty$, ce qui démontre 3.3.1.

De même, si une arête e de T ne forme pas un pont, il existera obligatoirement une arête de sa coupe de coût inférieur à ∞. À l'inverse, si elle forme un pont, son remplaçant aura un coût de ∞, ce qui démontre la remarque 3.3.2.

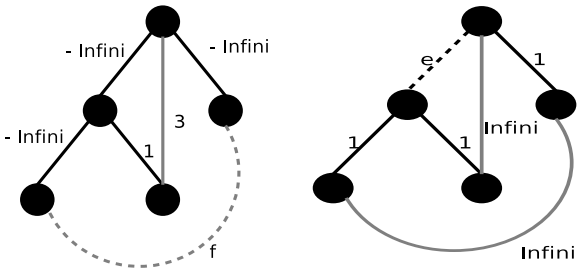

FIG. 4 – Illustration des remarques 3.3.1 et 3.3.2.

3.3.2 Filtrage des cycles et des ponts par les coûts

Ces deux remarques rendent inutile la recherche de cycles et de ponts, car elles permettent leur détection grâce à leur coût de remplacement.

En effet, l'affectation par les coûts de remplacement pour les arêtes $e \in T$ et $f \notin T$ se base sur l'équation suivante, où T' est la borne supérieure sur la solution optimale[4] :

$$cost(T) + cost(f) - cost(e) > cost(T')$$

[3] Le raisonnement est symétrique pour les arêtes n'appartenant pas à T.

[4] À noter qu'une telle borne peut toujours être calculée, en prenant par exemple l'arbre de coût maximum.

Si le coût de remplacement $cost(f) - cost(e)$ d'un pont e ou d'un cycle f est supérieur à $cost(T') - cost(T)$, il est possible de les affecter correctement par le filtrage primal. Cela permet de garantir la consistance de borne de la contrainte [5] sans nécessiter aucun calcul complémentaire.

4 Restrictions additionnelles

Jusqu'à maintenant, la contrainte primale d'arbre couvrant n'a été appliquée qu'à des problèmes de robustesse. Cependant, de nombreuses applications nécessitent l'ajout de restrictions supplémentaires, comme par exemple des restrictions sur le degré des nœuds ou sur la longueur du plus court chemin. Ces restrictions rendent $\mathcal{NP}$-difficile la recherche d'un arbre de poids optimal et gênent la création d'un filtrage primal prenant en compte leur effet. De plus, cet ajout pose des problèmes lors de l'application de contraintes primales tels que celle présentée précédemment.

Nous allons, dans cette section, appliquer les travaux de Sellmann à la contrainte d'arbre couvrant pour faire face à ces difficultés.

4.1 Problématique

Les particularités structurelles sur lesquelles se fondent les méthodes primales rendent difficile la prise en compte de restrictions. En effet, bien que les informations fournies par la recherche de remplaçants restent exploitables, car minorantes, le fait qu'elles ne prennent pas en compte les interactions entre la structure arborescente et les restrictions supplémentaires nuit à l'efficience du filtrage.

Il est possible d'ajouter une deuxième méthode de filtrage dual, prenant en compte la variation des coûts dûe à la violation des restrictions mais la coopération de ces deux méthodes n'est possible que si la borne fournie par la méthode duale respecte la structure sur laquelle est établie la méthode primale.

Les méthodes duales de décomposition peuvent offrir une solution à ce problème à la condition qu'elles permettent l'exploitation des coûts duaux. Il est également souhaitable de pouvoir utiliser la vaste littérature de recherche opérationnelle traitant des problèmes d'arbres couvrants sous restrictions.

4.2 Filtrage par relaxation Lagrangienne

Sellmann, dans [13], propose une réponse possible à ces exigences grâce à l'utilisation de la relaxation Lagrangienne.

En effet, cette méthode offre suffisamment de souplesse pour permettre la combinaison d'un filtrage primal et de l'utilisation d'informations duales.

4.2.1 Relaxation Lagrangienne

La relaxation Lagrangienne est une méthode consistant à dualiser la partie compliquante des restrictions dans la fonction objectif. Le but est de tirer profit de la structure principale du problème grâce à une décomposition se basant sur la résolution itérative d'un problème plus simple, et en général resoluble en temps polynomial.

À l'instar de la relaxation linéaire, cette méthode fournit à chaque itération des informations duales pour les restrictions transférées dans la fonction objectif. Cependant son intérêt réside dans le fait que, contrairement à la relaxation linéaire, la borne inférieure qu'elle fournie respecte la structure du sous-problème.

De manière plus concrète, posons le problème suivant :

$$\begin{aligned} min \quad & CX && (1)\\ & AX \leq A' && (2)\\ & BX \leq B' && (3)\\ \forall x \in X \quad & x \in \{0, \ldots, k\} && (4) \end{aligned}$$

Relaxer les restrictions (2) par relaxation Lagrangienne revient à les pénaliser, au niveau de la fonction objectif grâce aux coefficients de Lagrange λ :

$$\begin{aligned} min \quad & CX + \lambda(AX - A') && (5)\\ & BX \leq B' && (6)\\ \forall x \in X \quad & x \in \{0, \ldots, k\} && (7) \end{aligned}$$

Trouver la meilleure borne inférieure consiste à maximiser les coefficients λ. La théorie de la dualité de Lagrange stipule que la solution optimale du problème dual est une borne inférieure pour tout vecteur $\lambda \geq 0$. De plus, ces vecteurs définissent une information de valuation duale valide permettant le filtrage de X.

4.2.2 Méthodes primales et coefficients de Lagrange

Sellmann s'est appuyé sur ces propriétés pour définir une base théorique permettant l'utilisation des coefficients de Lagrange pour la prise en compte de restrictions additionnelles lors du filtrage primal.

Supposons que la dualisation de la restriction compliquante $AX \leq A'$ dans le problème précédent aboutit à un sous-problème pour lequel il existe un algorithme primal de filtrage par les coûts. Il est alors possible d'utiliser, pour tout $\lambda \geq 0$, cet algorithme avec

les coûts modifiés $C + \lambda A$ sans perdre l'exactitude de la recherche à condition de soustraire au coût global de la solution duale la partie constante $-\lambda A'$.

Pour une utilisation efficace de ces informations, il est nécessaire de rappeler quelques remarques de Sellmann :

Premièrement, les coefficients de Lagrange optimaux ne sont pas nécessairement ceux permettant le plus de filtrage. Bien que la borne inférieure augmente au fur et à mesure de la résolution itérative, les coefficients de Lagrange évoluent de manière non monotone. Comme, de plus, la borne majorante s'améliore tout au long de la recherche, il est préférable d'utiliser les informations inférées[5] sous la forme de contraintes locales valides pour un sous-arbre de l'arbre de recherche.

Deuxièmement, les méthodes incrémentales de calcul des coefficients de Lagrange ne garantissent pas toutes la validité de ces coefficients en cas d'affectation de variables au cours de la résolution. Il est donc important d'éviter d'affecter les variables sans précaution préalable.

4.2.3 Calcul des remplaçants

Différentes méthodes permettent le calcul des remplaçants. Soit le calcul est effectué dynamiquement. À chaque fois que le coût d'une arête est modifié ou qu'une arête est affectée, les arêtes dont le remplaçant a potentiellement changé sont recherchées et mises à jour. Le calcul des remplaçants peut également être calculé de manière statique grâce à des algorithmes calculant, pour un arbre et un graphe donnés, l'ensemble des remplaçants.

Il peut à première vue sembler intéressant d'utiliser une méthode dynamique évitant les mises à jours inutiles. Cependant, l'utilisation d'une relaxation Lagrangienne entraîne une forte variation des coûts, et il s'avère préférable d'utiliser un algorithme statique. En effet, les algorithmes dynamiques ont une complexité théorique de $O(|E| \log |V|)$ pour les arêtes de l'arbre, alors que celle des algorithmes statiques est de $O(|E|\alpha(|E|, |V|))$ pour le calcul de l'ensemble des remplaçants de toutes les arêtes du graphe[6].

Tarjan [14] fut le premier à proposer un algorithme statique permettant le calcul de l'ensemble des remplaçants, pour toutes les arêtes du graphe, grâce à la recherche du plus petit ancêtre commun. Comme l'illustre la figure 5, le plus petit ancêtre commun a de deux nœuds i et j est le premier nœud commun des chemins définis par i ou j et la racine. Seules les arêtes comprises entre i, j et a appartiennent au chemin entre i et j.

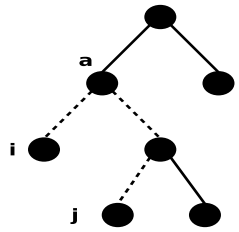

FIG. 5 – Plus petit ancêtre commun.

La solution proposée par Tarjan est de constituer, en $O(|E|\alpha(|E|, |V|))$, un graphe orienté acyclique possédant $O(|E|\alpha(|E|, |V|))$ nœuds représentant les liens de remplacement entre $e \in T$ et $f \notin T$, comme illustré figure 6.

Dans ce graphe, chaque arête f est représentée par un nœud n'ayant pas d'arc rentrant et chaque arête e par un nœud n'ayant pas d'arc sortant. Des nœuds intermédiaires, de degré deux, permettent de relier une arête f à une arête e si (e, f) définie un échange admissible. Il suffit, une fois ce graphe calculé, de le parcourir pour trouver l'ensemble des échanges admissibles et ainsi, calculer les remplaçants.

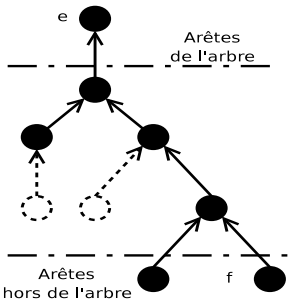

FIG. 6 – Graphe de lien de remplacement.

4.3 Algorithme de filtrage proposé

4.3.1 Algorithme dual

En nous basant sur les remarques de Sellmann, nous proposons l'algorithme 1, décrivant l'hybridation d'une contrainte primale par relaxation Lagrangienne.

Les données fournies à cette procédure sont :

- Les dernières informations calculées, représentées par la borne inférieure Sol et les coûts duaux λ_x.
- La borne de la solution optimale, pour laquelle $DSol.min$ et $DSol.max$ représentent respectivement la meilleure borne inférieure et la meilleure

[5] Ces informations sont les coûts de remplacement et la borne minorante pour laquelle ces coût sont valides.

[6] $\alpha(|E|, |V|)$ est la fonction inverse de Ackermann, de croissance inférieure à un logarithme.

Algorithme 1 : Contrainte duale par relaxation Lagrangienne pour le problème d'arbre couvrant.

```
Data :
  Sol : Dernier arbre couvrant calculé
  λ_x : Coûts duaux pour x
  DSol : Bornes de la solution optimale
1  pour chaque arête (i, j) faire
2  |  si x_(i,j) == 1 alors
3  |  |  cost(i, j) = −∞;
4  |  sinon si x_(i,j) == 0 alors
5  |  |  cost(i, j) = ∞;
6  |  fin
7  fin
8  Initialisation();
9  tant que Sol s'améliore faire
10 |  pour chaque arête (i, j) faire
11 |  |  si x_(i,j) ≠ 0 et x_(i,j) ≠ 1 alors
12 |  |  |  cost(i, j) = cost(i, j) + Σ_i λ_i a_i;
13 |  |  fin
14 |  fin
15 |  Sol = Kruskal();
16 |  cost(Sol) = cost(Sol) − Σ_i λ_i b_i;
17 |  PropagationPrimale(Sol,DSol,λ_x) ;
18 |  MiseAJourCoefficients(Sol,DSol,λ_x) ;
19 fin
```

borne supérieure trouvées. $DSol.min$ est une information locale. Par contre, $DSol.max$ est une information globale, représentant le coût de la meilleure solution trouvée jusqu'à maintenant dans tout l'arbre de recherche.

Cet algorithme commence, lignes (1) à (6), par fixer "à l'infini" les coûts des arêtes affectées. Cette opération permet la recherche des cycles et des ponts. Ensuite, les informations nécessaires aux calculs des λ sont re-initialisées. Cela permet d'assurer que les affectations d'arêtes effectuées depuis le dernier appel soient correctement prises en compte. Pour garantir la complètude, il est suffisant de réinitialiser les coefficients λ à "zéro" pour recommencer une nouvelle relaxation Lagrangienne.

Commence alors, tant que la borne inférieure converge suffisamment rapidement, l'itération de la relaxation. Les coûts des arêtes sont modifiées, ligne (12), en fonctions des λ. Cela permet, lignes (15) et (16), le re-calcul d'une solution minorante. Les coûts duaux étant pris en compte par la modification du coût des arêtes et de la borne inférieure, l'appel à l'algorithme primal, ligne (17), permet un filtrage hybride. Enfin, les coefficients sont mis à jour ligne (18) grâce à la nouvelle solution minorante.

Algorithme 2 : Nouvelle contrainte primale pour l'arbre couvrant pondéré.

```
Data :
  Sol : Dernier arbre couvrant calculé
  DSol : Bornes de la solution optimale
  δ_x^v : Coût de remplacement pour x = v
1  nbLibres = 0;
2  DSol.min ≥ cost(Sol);
3  Rep = CalculRemplaçant(Sol);
4  pour chaque arête (i, j) faire
5  |  si (i, j) ∈ Sol alors
6  |  |  δ_x^0 = max(δ_x^0, cost(Sol) − cost(i, j) +
   |  |  cost(Rep(i, j));
7  |  |  nbLibres+1;
8  |  |  mise à jour de la contrainte
   |  |  δ_x^0 ≤ DSol.max ⋁ x = 1 ;
9  |  sinon
10 |  |  δ_x^1 = max(δ_x^1, cost(Sol) + cost(i, j) −
   |  |  cost(Rep(i, j));
11 |  |  si x_(i,j) ≠ 0 alors
12 |  |  |  nbLibres+1;
13 |  |  fin
14 |  |  Mise à jour de la contrainte
   |  |  δ_x^1 ≤ DSol.max ⋁ x = 0 ;
15 |  fin
16 fin
17 si nbLibres < |V| − 1 alors
18 |  Echec();
19 fin
```

4.3.2 Algorithme primal

L'algorithme primal que nous proposons, dont le pseudo-code est illustré par l'algorithme 2, prend en entrée :

- La borne inférieure Sol et la borne optimale $DSol$.
- Les meilleurs coûts de remplacement trouvés δ_x^v.

Cette méthode commence, ligne (1), par initialiser la variables $nbLibres$ comptant le nombre d'arêtes non affectées à "zéro". Le test, ligne (17), effectué grâce à cette variables offre la possibilité de détecter plus rapidement l'infaisabilité dû à un nombre d'arêtes insuffisant pour former un arbre. Cette condition est nécessaire car, si elle n'est pas vérifiée, le calcul des ponts et des cycles devient erroné à cause d'arêtes affectées à "zéro" mais, malgré tout, présentes dans la borne inférieure.

Enfin, les informations inférées sont mises à jour et postées de manière locale dans le sous-problème considéré, lignes (8) et (14). Cela permet de prendre en compte l'évolution non-monotone des coûts de rem-

placement. Il faut cependant à ce niveau s'assurer que les affectations découlant de ces informations ne seront effectuées qu'à la fin de la relaxation Lagrangienne.

5 Expérimentations

Cette section présente les premiers résultats expérimentaux de la contrainte présentée dans cet article.

Pour ce faire, nous avons choisi d'utiliser un problème d'arbre couvrant sous restriction de degré, où le nombre d'arêtes connectées à un nœud doit être inférieur à une limite D.

Nous vérifions l'utilité de la coopération entre informations duales et primales en comparant la contrainte "Primale/Duale" par rapport à la contrainte primale seule et à une contrainte duale simple, n'utilisant pas la structure arborescente de la solution.

5.1 Description du prototype

Le prototype utilisé pour effectuer les tests a été implémenté sous Gecode [7] version 1.3.1. La relaxation Lagrangienne et l'algorithme de Tarjan ont été développés en C++ et le programme a été compilé sous Debian 4.0 grâce à $g++$ version 4.1.2 avec le drapeau d'optimisation $-O3$ activé. Nous avons choisi d'effectuer nos tests sur une restriction de degré fixée à 3.

5.1.1 Instances

Les graphes de test ont été générés aléatoirement comme indiqué dans [3] grâce au générateur Erdöos-Renyi de la librairie Boost version 1.33.1.

Pour chaque instance de n nœuds, un graphe a été généré en fonction d'un paramètre p indiquant, pour chaque pair de nœuds, la probabilité qu'une arête existe pour ces nœuds. Ce paramètre détermine la connexité du graphe résultant. À chaque arête est associé un coût tiré aléatoirement de manière uniforme entre 1 et 1000.

Tous les résultats fournis dans cette section ont été générés sur des graphes dont le nombre de nœuds varie de 100 en 100 entre 100 et 700 et dont la probabilité p varie de $0,5$ à 1(graphe complet) de $0,25$ en $0,25$.

5.1.2 Contrainte supplémentaire

À chaque algorithme testé dans cette expérimentation, nous avons décidé d'ajouter une contrainte *gcc* exprimant la restriction de degré. Cette contrainte, issue de [11], assure une consistance de borne pour la restriction de degré en $O(n^{1,5})$.

5.1.3 Relaxation Lagrangienne et restriction de degré

D'autres limitations que le problème de l'arbre couvrant sous restriction de degré, comme celle portant sur la profondeur de l'arbre [4], peuvent être résolues par ce type de méthode. Cependant, cette restriction a été choisie pour effectuer les expérimentations car elle est naturellement résolue par relaxation Lagrangienne du fait de sa faible complexité.

Nous utiliserons pour calculer les informations duales la relaxation proposée par Andrade [1] et fournissant à ce jour les meilleurs résultats de convergence. La relaxation est re-initialisée à chaque appel et le critère d'arrêt vérifie si la borne inférieure s'est améliorée depuis les n dernières itérations. Dans ce contexte de validation de l'approche, ce paramètre a été fixé à 5. Il pourrait cependant être intéressant, dans un soucis de performance, d'étudier l'impact du critère d'arrêt.

Nous utiliserons également à chaque itération de la relaxation la méthode "KRUSKALX sans amélioration" proposée par [1] dans ce même article pour calculer une borne supérieure.

5.2 Résultats

5.2.1 Méthode duale simple

Pour pouvoir comparer l'apport de la combinaison des informations duales et primales, il a été nécessaire d'utiliser une méthode simple de filtrage par les coûts duaux.

Nous avons choisie pour cela d'utiliser une mesure basée sur le nombre nécessaire d'arêtes d'un arbre couvrant. En effet, tout arbre couvrant d'un graphe de n nœuds possède $n-1$ arêtes. Cela permet d'inférer la présence des arêtes de la manière suivante :

Soient une solution minorante Sol et les coûts c_{max} de l'arête de coût maximum appartenant à l'arbre et c_{min} de l'arête hors de l'arbre de coût minimum. Pour toute arête $e \in T$, le coût de remplacement de e est supérieur ou égal à $cost(Sol) - cost(e) + c_{min}$ car l'arête permettant de reconnecter le graphe ne peut qu'avoir un coût supérieur ou égale à c_{min}. De même, pour toute arête $f \notin T$, le coût de remplacement de f est supérieur ou égal à $cost(Sol) - c_{max} + cost(f)$.

5.2.2 Description des méthodes

Pour effectuer les expérimentations, trois méthodes différentes sont été testées :

CDS pour "Contrainte Duale Seule". Cette contrainte n'utilise que très partiellement la structure arborescente grâce à la méthode d'inférence décrite dans la sous-section précédente.

CPS pour "Contrainte Primale Seule". Cette contrainte n'utilise aucune information duale, seule la contrainte primale basée sur les coûts de remplacement est appelée.

CPD pour "Contrainte Primale/Duale". Cette contrainte utilise les informations fournies par la relaxation dans la méthode primale basée sur les coûts de remplacement.

Les résultats de ces exécutions sont donnés par le tableau 1 page 320. Ce tableau présente, pour chaque méthode et pour chaque taille de graphe, la moyenne du temps de calcul total, la moyenne du nombre échecs total et le pourcentage d'échecs dû à la contrainte *gcc*.

Différents commentaires peuvent être faits à la vue de ce tableau.

5.2.3 Prise en compte des restrictions

La première hypothèse sur laquelle s'est fondée cet article est qu'une méthode primale perd de son efficacité en cas d'ajout de restrictions supplémentaires.

La comparaison entre contrainte duale simple et contrainte primale simple tend à la vérifier. En effet, on observe que le nombre moyen d'échecs pour la contrainte primale simple est légèrement inférieur à la contrainte duale simple. Cela signifie donc que la prise en compte de la structure primale seule ne suffit pas à capturer correctement la variation des coûts, car une méthode duale basique permet de faire légèrement mieux, pour un temps d'exécution légèrement plus rapide, dû à la simplicité du code permettant l'inférence.

5.2.4 Hybridation

Si l'on compare l'apport de la collaboration entre informations duales et primales, on observe qu'elle permet un gain de calcul très important.

La complexité de calcul d'une propagation est supérieure à celle des deux autres contraintes. Cependant on observe un nombre d'échecs beaucoup moins important pour la contrainte hybride. Cela permet à la méthode hybride d'être plus efficace et montre que les informations prises en compte par elle sont beaucoup plus riches.

6 Conclusion

Nous avons présenté dans cet article une nouvelle contrainte d'optimisation pour les problèmes d'arbres couvrants sous restrictions. Cette contrainte s'appuie sur une simplification de la contrainte primale, proposée par [2], et permet d'effectuer la détection des cycles et des ponts grâce à l'inférence sur les coûts de variation donnés par la structure d'arbre couvrant. En nous appuyant sur les travaux de Sellmann, nous avons pu également proposer une méthode hybride permettant la prise en compte de restrictions supplémentaires.

Les premières expérimentations, effectuées sur un problème d'arbre couvrant sous restriction de degré, montrent l'importance de la prise en compte des restrictions additionnelles dans l'algorithme de filtrage. En effet, l'utilisation de ces informations a permis, pour ce problème, de réduire considérablement le temps de calcul par rapport à des méthodes primales ou duales ne prenant en compte qu'une partie des informations disponibles.

Cependant, de nombreux points restent à éclaircir : La notion de coût de remplacement semble porteuse d'informations intéressantes qu'il serait utile de partager. Des résultats préliminaires semblent montrer que l'utilisation de ces coûts comme stratégie de branchement permet d'améliorer l'élagage de l'arbre de recherche.

De plus, Sellmann a montré que l'efficacité du filtrage par relaxation Lagrangienne ne dépend pas obligatoirement de la vitesse de convergence ni des coefficients de Lagrange optimaux. Il semble donc intéressant d'explorer les différentes relaxations utilisées à ce jour pour des problèmes d'arbres couvrants sous restrictions pour rechercher celle permettant le meilleur filtrage des variables.

Enfin, le calcul des remplaçants tient une très grande part dans le temps de calcul de la contrainte. Bien que les méthodes de mises à jour dynamiques classiques ont été écartées à cause de leur complexité, l'apparition récente de nouvelles structures de données [15] peuvent permettre de combiner algorithmes statiques et dynamiques.

Références

[1] Rafael Andrade, Abilio Lucena, and Nelson Maculan. Using Lagrangian dual information to generate degree constrained spanning trees. *discrete applied mathematics*, 154 :703–717, 2006.

[2] Ionuţ Aron and Pascal Van Hentenryck. A Constraint Satisfaction Approach to the Robust Spanning Tree Problem with Interval Data. In Adnan Darwiche and Nir Friedman, editors, *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, University of Alberta, Edmonton, Alberta, Canada*, pages 18–25. Morgan Kaufmann, 2002.

[3] Luis Caccetta and Sam Hill. A branch and cut method for the degree-constrained minimum spanning tree problem. *Networks*, 37 :74–83, 2001.

[4] Geir Dahl, Luís Gouveia, and Cristina Requeijo. On Formulations and Methods for the Hop-

n	p	CDS			CPS			CPD		
		Temps	Echecs		Temps	Echecs		Temps	Echecs	
		(seconds)	GCC (%)	Total	(seconds)	GCC (%)	Total	(seconds)	GCC (%)	Total
100	0.50	0,52	17,64	104,30	0,76	20,89	114,53	0,43	21,64	12,93
	0.75	0,77	18,50	103,76	1,06	21,43	117,86	0,40	18,11	8,46
	1.00	1,04	16,60	101,20	1,33	20,08	114,50	0,47	19,33	7,06
200	0.50	5,22	18,22	203,43	7,65	22,59	232,20	1,35	32,64	7,96
	0.75	7,37	16,66	207,20	11,02	19,96	233,20	1,82	22,69	8,66
	1.00	9,51	16,11	203,76	14,07	19,87	234,10	5,00	16,91	18,13
300	0.50	17,85	17,82	316,86	28,73	20,95	354,43	6,44	17,83	18,50
	0.75	24,41	16,19	302,26	36,45	19,94	349,80	10,29	21,59	21,30
	1.00	31,80	17,25	310,93	46,90	20,47	359,66	14,48	24,11	21,56
400	0.50	41,28	16,50	406,23	64,37	19,71	466,83	6,48	28,51	9,00
	0.75	57,02	16,10	399,43	84,55	20,00	466,66	24,06	25,48	26,03
	1.00	74,99	15,91	410,90	106,16	18,86	498,93	22,56	26,49	18,36
500	0.50	79,61	16,47	511,36	125,94	20,22	583,13	11,17	25,84	10,83
	0.75	113,28	15,65	527,36	168,90	18,41	615,83	25,22	19,09	17,63
	1.00	128,51	14,99	509,00	197,31	18,77	628,53	31,62	23,23	17,50
600	0.50	139,60	16,20	615,26	212,23	18,87	732,30	19,73	19,79	12,96
	0.75	178,89	14,56	588,56	260,06	18,73	706,96	44,28	15,64	23,43
	1.00	233,22	14,73	630,13	335,25	16,96	754,23	48,95	24,95	18,96
700	0.50	205,95	15,76	719,70	326,70	18,91	814,06	12,23	22,89	5,53
	0.75	268,65	14,18	720,26	419,77	17,04	806,50	46,11	25,19	16,93
	1.00	350,16	13,58	679,56	466,08	17,12	878,04	36,53	21,21	11,00

TAB. 1 – Temps de calcul moyen et nombre d'échecs moyen. n correspond au nombre de nœuds et p à la probabilité d'existence d'une arête.

Constrained Minimum Spanning Tree Problem. Technical report, CENTRO DE INVESTIGACAO OPERACIONAL, 2004.

[5] Gregoire Dooms and Irit Katriel. Graph constraints in constraint programming : Weighted Spanning Trees. INGI Research Report, 2006.

[6] Harold Gabow. Two Algorithms for Generating Weighted Spanning Trees in Order. *SIAM J. Comput.*, 6 :139–150, 1977.

[7] Gecode. http ://www.gecode.org/. generic constraint development environment.

[8] Thorsten Gellermann, Meinolf Sellmann, and Robert Wright. Shorter Path Constraints for the Resource Constrained Shortest Path Problem. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 3524 :201–216, 2005.

[9] John Hooker. A Search-Infer-and-Relax Framework for Integrating Solution Methods. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 3524 :243–257, 2005.

[10] Diego Olivier Fernandez Pons. Au sujet de certaines contraintes globales. In *JFPC2005*, 2005.

[11] Claude-Guy Quimper, Peter van Beek, Alejandro López-Ortiz, Alexander Golynski, and Sayyed Bashir Sadjad. An efficient bounds consistency algorithm for the global cardinality constraint. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming*, volume 2833, pages 600–614, Kinsale, Ireland, September 2003.

[12] Meinolf Sellmann. Cost-based filtering for shorter path constraints. *Principles and Practice of Constraint Programming - CP 2003*, 2833 :694–708, 2003.

[13] Meinolf Sellmann. Theoretical Foundations of CP-Based Lagrangian Relaxation. *Principles and Practice of Constraint Programming CP 2004*, 3258 :634–647, 2004.

[14] Robert Endre Tarjan. Applications of Path Compression on Balanced Trees. *J. ACM*, 26 :690–715, 1979.

[15] Renato Werneck. *Design and Analysis of Data Structures for Dynamic Trees*. PhD thesis, Princeton University, 2006.