



Une contrainte de circuit adaptée aux tournées multiples

Nicolas Briot, Philippe Vismara, Christian Bessiere

► To cite this version:

Nicolas Briot, Philippe Vismara, Christian Bessiere. Une contrainte de circuit adaptée aux tournées multiples. 13èmes Journées Francophones de Programmation par Contraintes (JFPC 2017), Jun 2017, Montreuil-sur-Mer, France. pp.135-144. hal-02733550

HAL Id: hal-02733550

<https://hal.inrae.fr/hal-02733550>

Submitted on 2 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Une contrainte de circuit adaptée aux tournées multiples

Nicolas Briot^{1*} Philippe Vismara² Christian Bessiere³

¹ Université Montpellier - LIRMM, Montpellier, France

² Montpellier SupAgro - LIRMM MISTEA, Montpellier, France

³ CNRS - LIRMM, Montpellier, France

{briot,vismara,bessiere}@lirmm.fr

Résumé

Il existe de nombreuses applications réelles contenant un problème de tournées de véhicules. La programmation par contraintes permet d'aborder ces problèmes de façon efficace. Des contraintes de circuits ont été définies pour traiter du problème de voyageur de commerce (TSP) ou de tournées de véhicules (VRP). Ces contraintes sont basées sur la recherche d'un circuit hamiltonien dans un graphe. Dans cet article, nous nous intéressons au problème plus général de tournées multiples dans lequel on cherche à couvrir une partie du graphe par un ensemble de circuits de coût minimal. Nous proposons une nouvelle contrainte globale basée sur la recherche de circuits élémentaires disjoints dans un graphe. Contrairement aux contraintes existantes, on ne cherche pas un circuit hamiltonien mais un ensemble de circuits de Steiner. Après avoir défini cette contrainte, nous montrons que le filtrage au bornes est NP-difficile. Nous proposons donc une méthode de filtrage incomplète basée sur la recherche d'une borne inférieure d'un circuit de Steiner.

De nombreux problèmes réels peuvent être ramenés à des problèmes de tournées. Les problèmes de tournées forment une grande famille dont les deux représentants les plus connus sont le problème du voyageur de commerce (*TSP*) et le problème de tournées de véhicules (*VRP*). Le premier consiste à trouver un circuit hamiltonien de coût minimum dans un graphe et le second est sa généralisation à plusieurs circuits avec un sommet dépôt commun. Ces problèmes se déclinent souvent avec d'autres contraintes comme par exemple, des fenêtres de temps ou des précédences sur les sommets du graphe ou encore des contraintes de taille sur les circuits. Tous ces problèmes sont NP-difficiles.

Les problèmes de base (sans contraintes particu-

lières) ont été étudiés de manière intensive en recherche opérationnelle. Ainsi, les méthodes de résolution exactes les plus efficaces actuellement sont basées soit sur la programmation dynamique soit sur la programmation linéaire en nombres entiers (*ILP*).

L'ajout de contraintes spécifiques au problème peut rendre l'approche *ILP* insuffisante du point de vue de l'expressivité ou de la résolution. La programmation par contraintes (*CP*) offre un cadre général plus à même de capturer et exploiter ces contraintes spécifiques pour résoudre le problème. Pour les problèmes de tournées, des contraintes de recherche de circuits dans un graphe ont été créées, mais elles ne suffisent pas toujours. Comme pour le problème du circuit de Steiner où l'ensemble de sommets à couvrir n'est pas fixe.

Après un rapide survol des principes de la programmation par contraintes, cet article énumère les contraintes de circuits ainsi que les principaux algorithmes de filtrage qui existent à ce jour. Nous étudions leurs limites et proposons une nouvelle contrainte capable d'y répondre. Nous montrons la complexité de la nouvelle contrainte et nous donnons les premiers algorithmes de filtrage.

1 Préliminaires

En *CP*, chaque variable $x \in X$ est associée à un domaine $dom(x)$ qui correspond à l'ensemble des valeurs que peut prendre la variable. Les domaines sont restreints par un ensemble de contraintes C . Une contrainte est une relation qui porte sur un ensemble de variables et qui définit l'ensemble des valeurs que peuvent prendre simultanément les variables. Une so-

*Papier doctorant : Nicolas Briot¹ est auteur principal.

lution au problème est donnée quand tous les domaines sont réduits à un singleton (les variables sont alors instanciées) et quand toutes les contraintes sont satisfaites. Une contrainte est satisfaite si l'ensemble des valeurs prises par les variables satisfait la relation.

En CP, des contraintes dites globales permettent de capturer une partie de la structure combinatoire d'un problème. Chacune de ces contraintes est associée à un algorithme de filtrage qui va détecter des valeurs impossibles dans le domaine des variables. Il est important que le filtrage soit rapide et efficace. Pour cela, plusieurs niveaux de filtrage ont été définis. La cohérence d'arc généralisée (GAC) est un haut niveau de filtrage qui traite les contraintes indépendamment.

Définition. *Étant donné une contrainte $c \in C$ qui porte sur les variables $(x_1, \dots, x_n)$ de domaines respectifs $dom(x_1), \dots, dom(x_n)$, la contrainte c est cohérente d'arc généralisée (GAC) si et seulement si $\forall x_k \ k \in 1..n$ et $\forall v \in dom(x_k)$, il existe un tuple $(v_1, \dots, v, \dots, v_n)$ avec $v_i \in dom(x_i)$ qui satisfait la contrainte.*

Appliquer GAC peut être exponentiel sur certaines contraintes globales. Moins forte, la cohérence aux bornes (BC) peut être une option plus abordable.

Définition. *Étant donné une contrainte $c \in C$ qui porte sur les variables $(x_1, \dots, x_n)$, la contrainte c est cohérente aux bornes (BC) si et seulement si $\forall x_i \in c$: les tuples $(v'_1, \dots, LB(v_i), \dots, v'_n)$ et $(v_1, \dots, UB(v_i), \dots, v_n)$ satisfont la contrainte avec $LB(x_k)$ (resp. $UB(x_k)$) la valeur minimale (maximale) de $dom(x_k)$ et v_j, v'_j ($j \neq i$) des nombres dans l'intervalle $[LB(v_j), \dots, UB(v_j)]$.*

2 Les contraintes de circuit

Considérons un graphe orienté $G = (V, A, c)$, avec $V = \{1, \dots, n\}$ l'ensemble des sommets et A l'ensemble des arcs valués par une fonction de coût c .

En CP, la recherche de circuits hamiltoniens dans G peut être modélisée en posant des contraintes globales sur un ensemble de variables décrivant le successeur de chaque sommet.

Pour tout sommet $i \in V$, notons $Next_i$, la variable qui représente le sommet successeur du sommet i . Le domaine $dom(Next_i)$ correspond à l'ensemble des sommets adjacents à i dans le graphe, c'est-à-dire $j \in dom(Next_i)$ si et seulement si $(i, j) \in A$.

Par définition, les variables successeurs $Next_i$ décrivent un sous-graphe orienté. Pour les graphes non-orientés, une arête peut être représentée par deux arcs entre les deux sommets. Dans ce cas, la recherche d'un cycle dans un graphe non-orienté

correspond à la recherche d'un circuit dans le graphe orienté équivalent. Dans la suite, nous désignerons indifféremment par TSP le problème du voyageur de commerce orienté (symétrique) ou non-orienté (asymétrique).

La contrainte

$$\text{CIRCUIT}(Next_1, \dots, Next_n) [1]$$

est satisfaite si et seulement si l'ensemble des arcs définis par $\{(i, Next_i) \in A\}$ forme un circuit hamiltonien dans G .

Établir la GAC pour cette contrainte est NP-difficile puisque cela revient à chercher des circuits hamiltoniens dans un graphe.

Des approches incomplètes ont donc été proposées pour le filtrage de la contrainte CIRCUIT. Premièrement, la contrainte requiert que deux sommets n'aient pas le même successeur. Cette propriété correspond à la contrainte ALLDIFFERENT($Next_1, \dots, Next_n$) qui possède un filtrage GAC efficace. Deuxièmement, l'élimination des sous-circuits a été proposée dans Pesant *et al.* [12] et Caseau *et al.* [3]. L'algorithme interdit les affectations du type $Next_i = j$ si l'arc (i, j) provoque un sous-circuit dans le sous-graphe induit par les arcs $\{(i, Next_i) \mid Next_i \text{ est instanciée}\}$. Il existe aussi un algorithme de filtrage moins utilisé basé sur la recherche de séparateur de graphes (voir [11]).

Une version de la contrainte prenant en compte le coût du circuit a été proposée en premier par Focacci *et al.* [7, 5, 6] puis par Benchimol *et al.* [2]. Cette contrainte a été développée pour répondre précisément au problème du TSP.

Soit Z une variable entière qui représente le coût du circuit hamiltonien. Cette fois, la contrainte

$$\text{WEIGHTEDCIRCUIT}[G](Next_1, \dots, Next_n, Z)$$

est satisfaite si et seulement si l'ensemble des arcs définis par $\{(i, Next_i) \in A\}$ forme un circuit hamiltonien dans G de coût $\leq Z$ où le coût du circuit est la somme des poids des arcs du circuit.

La première partie du filtrage de la contrainte WEIGHTEDCIRCUIT est équivalente au filtrage de la contrainte CIRCUIT.

Durant la recherche, des valeurs dans le domaine des variables successeurs seront retirées. Ces valeurs correspondent à des arcs du graphe de départ et, au fur et à mesure, le graphe potentiel se vide jusqu'à obtenir un circuit. Dès qu'un arc est retiré du graphe, la borne inférieure du coût du circuit $LB(Z)$ peut être mise à jour. Une partie du travail de l'algorithme de filtrage

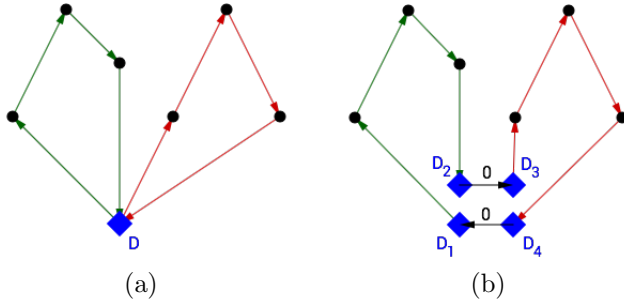


FIGURE 1 – Transformation d’un problème de VRP à deux tournées en un problème de TSP. Le nœud dépôt est dupliqué en autant de paires de sommets (départ/arrivée) que de tournées, afin de se ramener à un circuit hamiltonien.

est donc de trouver la meilleure borne possible pour $LB(Z)$. Cette borne peut être calculée grâce à deux relaxations du TSP : le problème de l’arborescence (ou arbre en version non-orienté) couvrante de coût minimum (MSA) et le problème d’affectation (AP). AP consiste à relâcher la contrainte d’élimination des sous-tours dans le TSP et revient donc à chercher un ensemble de circuits élémentaires couvrant le graphe et de coût minimum.

Benchimol *et al.* [2] ont proposé d’utiliser la relaxation 1-tree définie par Held et Karp [8, 9] qui donne une relaxation strictement meilleure que l’arbre couvrant de coût minimum. Ces deux relaxations (Held & Karp et AP) se calculent en temps polynomial. Ducommun *et al.* [4] ont montré que les bornes obtenues par ces filtrages étaient incomparables et que les filtrages pouvaient être utilisés indépendamment pour filtrer $LB(Z)$.

Disposer d’une bonne estimation de $LB(Z)$ permet également de filtrer les variables $Next_i$ en éliminant certains arcs incompatibles avec la valeur de $UB(Z)$ (voir Benchimol *et al.* [2] et Focacci *et al.* [6]).

Les contraintes de circuit présentées ici sont limitées à la recherche d’un circuit hamiltonien. Elles s’appliquent donc assez bien à des problèmes comme le TSP, le TSP avec fenêtres de temps, etc. Modéliser un problème plus général comme le VRP avec une contrainte de circuit nécessite une modélisation plus complexe. La figure 1 illustre la transformation nécessaire pour passer d’un VRP à un TSP et pouvoir ainsi utiliser une contrainte de circuit. Cette modélisation impose d’avoir au moins un sommet fixé dans chaque circuit (ici, le sommet dépôt), ce qui n’est pas forcément le cas pour certains problèmes de couverture de sommets.

Les contraintes de circuit hamiltonien ne peuvent

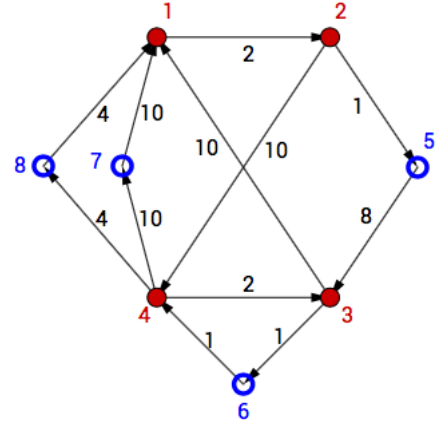


FIGURE 2 – Exemple du problème du circuit de Steiner. En rouge (plein) les sommets terminaux et en bleu (cercle) les sommets potentiels

pas non plus être utilisées lorsque l’ensemble des sommets intervenant dans le circuit n’est pas totalement connu.

Considérons un graphe $G = (V, A, c)$ et soit $V^* \subseteq V$ un sous-ensemble de sommets dit obligatoires (ou terminaux). Le *problème du circuit de Steiner* dans G pour V^* consiste à trouver un circuit élémentaire dans G de coût minimum passant par tous les sommets de V^* . Le circuit de coût minimum peut aussi passer par des sommets extérieurs à V^* , ce sont les nœuds de Steiner. Le TSP est un cas particulier où $V^* = V$. Ce problème est fortement lié au problème plus connu de l’arbre de Steiner [10].

Exemple 1. Soit $G = (V, A, c)$ le graphe de la figure 2 et $V^* = \{1, 2, 3, 4\}$ les sommets terminaux (obligatoires). La solution au circuit de Steiner est la séquence de sommets $(1, 2, 5, 3, 6, 4, 8, 1)$ de coût 21. Dans cette solution, le sommet 7 n’intervient pas dans le circuit optimal.

3 La contrainte WeightedSubCircuits

Dans cette section, nous proposons une nouvelle contrainte qui répond aux limites des contraintes de circuit présentées précédemment. La nouvelle contrainte cherche au plus K circuits élémentaires disjoints de coût minimum couvrant une partie du graphe. Un circuit élémentaire est un circuit qui passe une et une seule fois sur chaque sommet du circuit. Dans la suite, nous admettons qu’une boucle sur un sommet est un circuit élémentaire (composé d’un arc).

Soit $G = (V, A, c)$ un graphe orienté avec $V = \{1, \dots, n\}$ l’ensemble des sommets, $A = \{(i, j) \mid i, j \in$

V l'ensemble des arcs et la fonction de coût $c : A \rightarrow \mathbb{N}$ associée à chaque arc de G .

Définition. Étant données les variables :

- $Set_k \forall k \in \{1, \dots, K\}$, une variable ensembliste représentant les sommets de G qui sont dans le $k^{\text{ème}}$ circuit. On a $\text{dom}(Set_k) = 2^V$ (l'ensemble des parties de V).
- Set_{dummy} , la variable ensembliste représentant l'ensemble des sommets exclus de tous les circuits. $\text{dom}(Set_{dummy}) = 2^V$
- $Next_i \forall i \in V$, représente le sommet successeur du sommet i . $\text{dom}(Next_i) \subseteq V$.
- Z et $Cost_k \forall k \in \{1, \dots, K\}$ des variables entières qui représentent respectivement le coût total des circuits et le coût du circuit k .

La contrainte WEIGHTEDSUBCIRCUITS, notée

$$\text{WSC}[G](\{Set_k\}_{k=1..K}, Set_{dummy}, \{Next_i\}_{i=1..n}, \{Cost_k\}_{k=1..K}, Z) \quad (1)$$

est satisfaite si et seulement si les conditions suivantes sont respectées :

1. $\forall k, k' \in 1..K \cup dummy, (k \neq k' \Rightarrow Set_k \cap Set_{k'} = \emptyset)$;
2. $Set_1 \cup \dots \cup Set_K \cup Set_{dummy} = V$;
3. $\forall k \in 1..K$:
 - si $Set_k = \emptyset$ alors $Cost_k = 0$.
 - sinon, $G[Set_k]$ le sous-graphe de G induit par l'ensemble de sommets Set_k et par l'ensemble d'arcs $\{(i, Next_i) \mid i \in Set_k\}$ est un circuit élémentaire de G de coût $\leq Cost_k$.
4. $\forall i \in V, i \in Set_{dummy} \Leftrightarrow Next_i = i$;
5. $\sum_{k=1}^K Cost_k \leq Z$;

La contrainte WSC impose un ensemble d'au plus K circuits élémentaires disjoints dans G de coût total $\leq Z$. Les conditions 1 et 2 correspondent à une partition des sommets dont certaines parties peuvent être vides. La condition 3 autorise des circuits vides dont le coût associé vaut 0. Si l'ensemble possède au moins un sommet, alors le sous-graphe induit par les sommets de Set_k et les arcs $\{(i, Next_i) \mid i \in Set_k\}$ est un circuit élémentaire de G dont le coût ne dépasse pas $Cost_k$. S'il n'y a qu'un seul sommet i , le circuit est réduit à une boucle de coût $c(i, i)$.

Il faut noter qu'un sommet i est exclu des K circuits si et seulement si la valeur i est dans l'ensemble Set_{dummy} et la condition 4 associe la valeur i à $Next_i$. À l'inverse, si $Set_{dummy} = \emptyset$ alors tous les sommets sont impliqués dans les circuits. De même, si $i \notin D(Next_i)$ alors i doit être dans un circuit. Enfin,

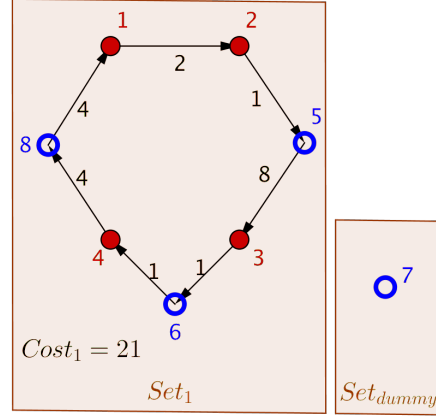


FIGURE 3 – La solution optimale au problème du circuit de Steiner sur le graphe de la figure 2. Les variables $Next_i$ sont représentées par les flèches et les ensembles par les rectangles.

le coût total des circuits est borné par Z dans la condition 5.

Dans cet article, nous proposons une définition de la contrainte utilisant des variables ensemblistes pour décrire l'ensemble des sommets associés à chaque circuit. Une perspective intéressante serait d'étudier d'autres représentations de ces ensembles, par exemple par des variables entières.

Exemple 2. Reprenons le graphe de la figure 2. Pour $K = 1$ on modélise le problème du circuit de Steiner par le réseau de contraintes suivant :

- $LB(Set_1) = \{1, 2, 3, 4\}, UB(Set_1) = V$;
- $LB(Set_{dummy}) = \emptyset, UB(Set_{dummy}) = V$;
- $Next_1 = 2, Next_2 \in \{4, 5\}, Next_3 \in \{1, 6\}, Next_4 \in \{7, 8\}, Next_5 \in \{3, 5\}, Next_6 \in \{4, 6\}, Next_7 \in \{1, 7\}$ et $Next_8 \in \{1, 8\}$;
- $Cost_1, Z \in \mathbb{N}$;
- $\text{WSC}[G](Set_1, Set_{dummy}, Next_1, \dots, Next_8, Cost_1, Z)$

sur lequel on cherche à minimiser Z .

Dans cet exemple, un seul circuit est recherché et $Cost_1$ peut être confondu avec Z . La solution optimale est représentée par la figure 3. Le circuit a un coût de 21. Le sommet 7 est exclu du circuit car il fait partie de l'ensemble Set_{dummy} et sa variable successeur $Next_7$ est instanciée à l'unique valeur 7 (non représentée sur la figure).

Théorème 1. Établir BC et établir GAC pour la contrainte WSC est NP-difficile.

Démonstration. Décider si un graphe $G = (V, A, c)$ admet un circuit hamiltonien de coût inférieur ou égal à une valeur p est NP-complet. Considérons la contrainte

$WSC[G](\{Set_1, Set_2\}, \{Next_i\}_{i=1..n}, \{Cost_1\}, Z)$, où $LB(Set_1) = V$, où chaque $Next_i$ a pour domaine l'ensemble des successeurs du sommet i dans A et où $Cost_1 \in \{0..p\}$, $Z \in \{0..p\}$. BC vide le domaine de Z si et seulement si le graphe G admet un circuit hamiltonien de coût inférieur ou égal à p . $\square$

3.1 Décomposition

Pour implémenter le filtrage de la contrainte WEIGHTEDSUBCIRCUITS, une première approche consiste à remplacer la contrainte par un ensemble de contraintes équivalentes en termes d'expressivité. Cela permet de bénéficier directement des filtres associés à ces contraintes. Nous verrons par la suite que cet ensemble de contraintes n'est pas équivalent à la contrainte WEIGHTEDSUBCIRCUITS au niveau du filtrage.

Proposition 1.

$$\begin{aligned}
& WSC[G](\{Set_k\}_{k=1..K}, Set_{dummy}, \{Next_i\}_{i=1..n}, \\
& \quad \{Cost_k\}_{k=1..K}, Z) \equiv \\
& \quad ALLDIFFERENT(Next_1, \dots, Next_n) \quad (2) \\
\wedge & \quad PARTITION(Set_1, \dots, Set_K, Set_{dummy}) \quad (3) \\
\wedge & \quad \forall i \in V, i \in Set_{dummy} \Leftrightarrow Next_i = i \quad (4) \\
\wedge & \quad \forall i \in V, \forall k = 1..K, dummy, i \in Set_k \Leftrightarrow Next_i \in Set_k \quad (5) \\
\wedge & \quad \forall k = 1..K, NOSUBTOURS(Set_k, Next_1, \dots, Next_n) \quad (6) \\
\wedge & \quad \forall k = 1..K, \sum_{i \in Set_k} c(i, Next_i) = Cost_k \quad (7) \\
\wedge & \quad \sum_{k=1}^K Cost_k \leq Z \quad (8)
\end{aligned}$$

La première contrainte impose que deux variables successeurs ne peuvent pas avoir la même valeur.

La deuxième contrainte concerne uniquement les variables ensemblistes. Rappelons que chaque variable $Set_k \forall k \in 1..K$ représente l'ensemble des sommets de G présents dans le $k^{\text{ème}}$ circuit élémentaire (ou boucle) et la variable Set_{dummy} est l'ensemble des sommets exclus des circuits. Ainsi, chaque sommet est associé à un unique ensemble. La contrainte PARTITION utilisée ici n'est pas une partition au sens mathématique car elle autorise des ensembles vides.

Les contraintes de *channelling* (4) et (5) assurent la cohérence entre les variables ensemblistes Set_k et les variables de successeurs $Next_i$.

La contrainte (6) garantit que les sommets de chaque ensemble Set_k font partie d'un circuit élémentaire décrit par les variables de successeurs $Next_i$. Nous utilisons ici une version adaptée de la contrainte de Pesant *et al.* [12] vue plus haut. Notre contrainte

interdit les affectations $Next_i = j$ si l'arc (i, j) provoque un sous-circuit de taille strictement inférieure à $|Set_k|$ dans le sous-graphe induit $G[Set_k]$.

La contrainte (7) garantit que le coût du circuit associé à chaque Set_k est majoré par la variable $Cost_k$. On peut définir une contrainte globale dont la portée sera $(Set_k, \{Next_i\}, Cost_k)$ et dont le filtrage correspond à une somme partielle. La contrainte (8) garantit que la somme des coûts des circuits ne dépasse pas Z .

La proposition 1 permet de propager la contrainte WEIGHTEDSUBCIRCUITS en se basant sur des contraintes existantes ou des variantes relativement proches. Néanmoins, cette approche ne permettra pas d'atteindre le niveau de filtrage équivalent à celui de la contrainte WEIGHTEDCIRCUIT. Dans la prochaine section nous montrerons comment améliorer ce filtrage en tenant compte des coûts des circuits.

4 Filtrage basé sur les coûts

Caseau et Laburthe [3] ont été les premiers à proposer un algorithme de filtrage basé sur le coût du circuit. Depuis, plusieurs travaux [6, 2] ont montré l'intérêt d'avoir ce filtrage complémentaire. À partir d'une instanciation partielle des variables, le but de cet algorithme est d'avoir une meilleure estimation de $LB(Cost_k)$, de $LB(Z)$ et de détecter certains arcs obligatoires ou interdits dans les circuits.

Les algorithmes de filtrage présentés dans cette section sont appliqués lorsque le graphe $G = (V, A, c)$ respecte la propriété d'*inégalité triangulaire faible* :

Propriété 1. Nous dirons qu'un graphe $G(V, A, c)$ respecte l'*inégalité triangulaire faible* ssi $\forall (i, j) \in A$, $c(i, j) \leq \text{coût de tout chemin entre } i \text{ et } j \text{ dans } G$.

À la manière de la contrainte WEIGHTEDCIRCUIT, nous cherchons une borne inférieure à $Cost_k$ pour tout $k = 1..K$. $Cost_k$ est la variable qui représente le coût du circuit hamiltonien dans $G[Set_k]$, le sous-graphe induit de G par les sommets présents dans Set_k . Par la suite, nous considérons seulement les ensembles tels que $|LB(Set_k)| > 1$, $\forall k = 1..K$. Suivant l'état de la variable ensembliste, on peut distinguer deux cas.

Premièrement, si Set_k est instanciée (c'est-à-dire $LB(Set_k) = UB(Set_k)$) alors tous les sommets de l'ensemble et uniquement eux participent au $k^{\text{ème}}$ circuit élémentaire de G . Le circuit est donc un circuit hamiltonien de $G[Set_k]$. Dans ce cas, on peut utiliser les mêmes relaxations que pour WEIGHTEDCIRCUIT sur $G[Set_k]$ afin de déterminer une borne inférieure du coût du circuit.

Deuxièmement, si Set_k n'est pas instanciée alors les sommets de $UB(Set_k)$ sont partagés en deux catégories. Il y a les sommets de $LB(Set_k)$ forcément impli-

qués dans le circuit. Ces sommets sont dit *terminaux*. Les autres sont les sommets dit *potentiels* et appartiennent à $UB(Set_k) \setminus LB(Set_k)$.

On cherche ici une borne inférieure au coût du $k^{\text{ème}}$ circuit élémentaire en sachant que tous les sommets terminaux en font obligatoirement partie. Chercher ce circuit correspond au problème du circuit de Steiner présenté plus haut. La suite est consacrée à l'établissement d'une borne inférieure au coût du circuit de Steiner.

4.1 Calcul d'une borne inférieure au coût d'un circuit de Steiner

Comme le MSA pour le problème du voyageur de commerce, le problème de l'arborescence de Steiner (Directed Steiner Tree) est une relaxation du circuit de Steiner. Ce problème correspond à la recherche d'une arborescence de poids min dans un graphe enraciné telle que l'arborescence contient au moins tous les sommets terminaux. C'est une généralisation à la fois du MSA et du problème plus connu de l'arbre de Steiner (Steiner Tree Problem) dans les graphes non-orientés.

Malheureusement, les problèmes de l'arborescence et de l'arbre de Steiner sont NP-difficiles [13, 10]. Il n'est donc pas envisageable d'utiliser ces relaxations pour trouver une borne inférieure au circuit de Steiner dans notre algorithme de filtrage. On peut faire la même remarque pour le problème d'assignement.

Nous allons également montrer que raisonner uniquement sur le sous-graphe induit par les sommets terminaux est incorrect.

Proposition 2. Soit $G = (V, A, c)$ un graphe respectant l'inégalité triangulaire faible et $V^* \subseteq V$. Le coût d'un MSA (ou AP) sur le sous-graphe $G[V^*]$ peut être supérieur au coût du circuit de Steiner couvrant dans G les sommets terminaux de V^* .

Démonstration. Prenons par exemple le graphe de la figure 4 et $V^* = \{1, 2, 3, 4\}$. Le coût du MSA sur $G[V^*]$ vaut 22 alors que le coût du circuit de Steiner sur (G, V^*) vaut 21. $\square$

Afin de pouvoir établir une borne inférieure du coût d'un circuit hamiltonien sur chaque Set_k , nous proposons de calculer un MSA (ou AP) dans un nouveau graphe orienté $G^* = (V^*, A^*, c^*)$.

Définition 1. Soit $G = (V, A, c)$ un graphe orienté respectant l'inégalité triangulaire faible et $V^* \subseteq V$ un ensemble de sommets terminaux. Le graphe extrapolé $G^* = (V^*, A^*, c^*)$ est défini ainsi :

- si $(i, j) \in A$ alors $(i, j) \in A^*$ et $c^*(i, j) = c(i, j)$.
- sinon, s'il existe au moins un chemin entre i et j dans G composé exclusivement de sommets $\notin V^*$

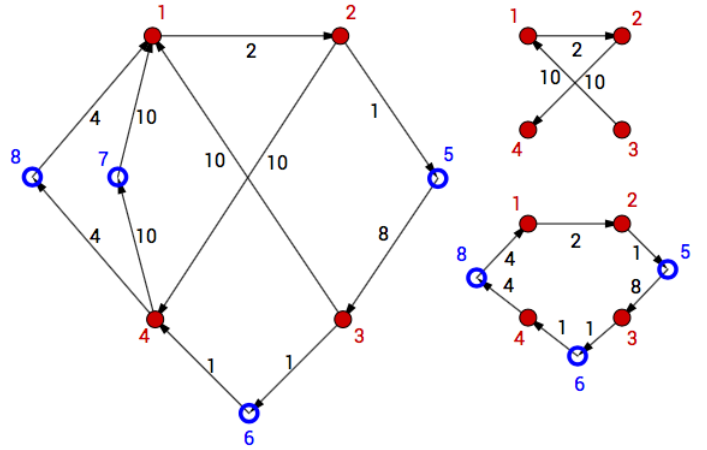


FIGURE 4 – Graphe avec des sommets terminaux en rouge (plein) et potentiels en bleu (cercle). En haut : la solution du MSA sur les sommets terminaux. En bas : la solution au circuit de Steiner sur le graphe entier.

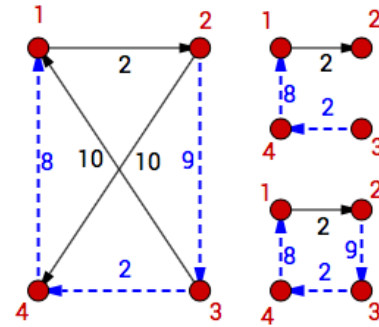


FIGURE 5 – Graphe G^* à partir du graphe de la figure 4. En haut, la solution au MSA et en bas la solution au AP sur G^* .

(sauf i et j) alors $(i, j) \in A^*$ et $c^*(i, j)$ est égal au coût du plus court de ces chemins.

Exemple 3. La figure 5 présente le graphe extrapolé correspondant au graphe de la figure 4. La transformation en G^* ajoute un arc $(4, 1)$ car il existe un chemin entre les sommets 4 et 1 ($4 - 8 - 1$) de poids 8 (plus court chemin). Même chose pour $(3, 4)$ et $(2, 3)$ de poids respectifs 2 et 9.

Nous allons montrer que le coût d'une solution au AP (ou MSA) dans le graphe G^* est une borne inférieure au coût du circuit de Steiner dans G pour V^* .

Lemme 1. Étant donné $G = (V, A, c)$ respectant l'inégalité triangulaire faible et $G^* = (V^*, A^*, c^*)$ le graphe extrapolé défini sur $V^* \subseteq V$. Le coût d'un AP (resp. MSA) sur G^* est une borne inférieure au coût du circuit de Steiner dans G pour V^* .

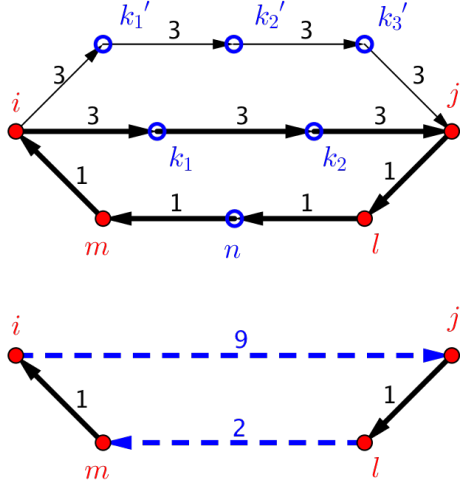


FIGURE 6 – En haut le graphe avec les sommets terminaux en rouge (plein) et potentiels en bleu (cercle). $\mathcal{C}$ est représenté par les arcs en gras. En bas, $\mathcal{C}^*$ du graphe extrapolé G^* sur les sommets terminaux. Le sous-chemin (i, k_1, k_2, j) de $\mathcal{C}$ est remplacé par l'arc (i, j) dans $\mathcal{C}^*$.

Démonstration. Soit $\mathcal{C}$, une solution optimale au circuit de Steiner dans G pour V^* . Comme le montre la figure 6, le circuit $\mathcal{C}$ comporte des sous-chemins de la forme $(i, k_1, \dots, k_p, j)$ où $i, j \in V^*$ et $k_1, \dots, k_p \in V \setminus V^*$. Par construction de G^* , il existe nécessairement un arc (i, j) dans A^* tel que $c^*(i, j) \leq (c(i, k_1) + \dots + c(k_p, j))$. En remplaçant dans $\mathcal{C}$ tous les sous-chemins $(i, k_1, \dots, k_p, j)$ par l'arc (i, j) correspondant dans A^* , on obtient un circuit hamiltonien de G^* . Notons $\mathcal{C}^*$ ce circuit.

En supprimant un arc de $\mathcal{C}^*$, on obtient une arborescence recouvrante de G^* de coût inférieur au coût de $\mathcal{C}^*$ (car tous les poids des arcs sont positifs). Le coût d'un MSA de G^* est donc inférieur au coût du circuit de Steiner dans G pour V^* .

De manière analogue, on peut remarquer que le cycle $\mathcal{C}^*$ est une solution au problème d'affectation (AP) sur le graphe G^* . $\square$

Exemple 4. Reprenons l'exemple 3. Dans G^* , le coût du MSA est de 11 et le coût d'un AP est de 21. Ces deux valeurs sont bien des bornes inférieures au cycle de Steiner dans le graphe G sur V^* .

4.2 Application au filtrage des coûts des circuits

Pour filtrer la borne inférieure de $Cost_k$, nous pouvons calculer sur le graphe $G^* = (LB(Set_k), A^*, c^*)$ une des deux relaxations (MSA ou AP) pour avoir une minoration du coût du circuit. Comme le calcul d'un

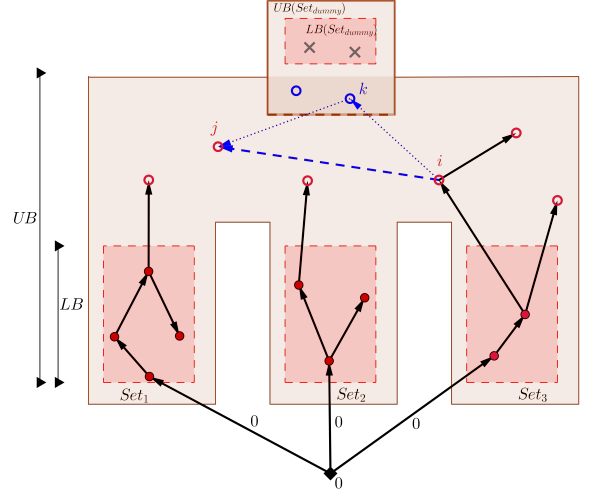


FIGURE 7 – Illustration du MSA sur G^* . En rouge (plein et cercle) sont les sommets obligatoires. Les cercles bleus représentent les sommets potentiels et les cruciformes sont ignorés.

MSA ou d'un AP est polynomial tout comme la création de G^* , l'algorithme de filtrage proposé ici reste polynomial.

Comme pour la contrainte WEIGHTEDCIRCUIT [2, 6], disposer d'une bonne estimation du coût minimal de chaque circuit de Set_k permet aussi d'éliminer certains arcs incompatibles avec la borne supérieure de $Cost_k$. On peut ainsi filtrer des valeurs dans les domaines de certains $Next_i$ (pour $i \in LB(Set_k)$).

Intéressons nous maintenant à la variable Z qui correspond au coût global des circuits dans les Set_k .

La contrainte globale WEIGHTEDSUBCIRCUITS peut permettre un filtrage plus fin que celui réalisé uniquement au niveau du circuit de chaque Set_k . L'ensemble Set_{dummy} correspond aux sommets qui ne feront partie d'aucun circuit formé par un ensemble Set_k pour $k \leq K$. Parmi les sommets qui ne sont pas dans $UB(Set_{dummy})$, certains sont déjà affectés à un circuit et font partie d'un ensemble $LB(Set_k)$ ($k \leq K$). Les autres sommets feront nécessairement partie d'un circuit mais ils ne sont pas pris en compte par les filtrages réalisés pour chaque circuit. Il est donc possible de déterminer une borne inférieure au coût total des futurs circuits en considérant l'ensemble des sommets qui ne seront pas dans Set_{dummy} . Notons $\mathcal{O}$ l'ensemble de ces sommets obligatoires. On a $\mathcal{O} = V \setminus UB(Set_{dummy})$.

Considérons le graphe $G' = (V \cup \{0\}, A', c')$ tel que $A \subset A'$ et A' contient un arc reliant le nouveau sommet 0 à un seul sommet de chaque ensemble $LB(Set_k)$, pour $k \leq K$. On se place ici dans le cas où $|LB(Set_k)| \geq 1, \forall k \leq K$. La fonction de coût c' est identique à c avec un coût nul pour tout arc issu de 0.

On construit le graphe extrapolé $G^{*'} = (\mathcal{O} \cup \{0\}, A^{*'}, c^{*'})$ sur G' . Comme précédemment, on peut montrer qu'un MSA de $G^{*'}$ est une borne inférieure à la somme des coûts des circuits élémentaires sur les Set_k .

Par exemple, la figure 7 illustre le MSA sur le graphe $G^{*'}$. Le graphe $G^{*'}$ est construit avec les sommets obligatoires déjà associés à un circuit (en rouge plein) et les sommets qui feront partie d'un circuit (en cercle rouge). Les sommets dans $UB(Set_{dummy}) \setminus LB(Set_{dummy})$ (en cercle bleu) sont aussi pris en compte dans la construction : l'arc (i, j) est créé car le chemin (i, k, j) existe. Le sommet 0 et les arcs incidents de poids nul sont ajoutés.

5 Conclusion

Dans cet article, nous nous sommes intéressés aux contraintes de circuits qui sont notamment utilisées pour traiter des problèmes de voyageur de commerce (TSP) ou de tournées (VRP). Nous avons montré que les contraintes CIRCUI et WEIGHTEDCIRCUIT proposées dans la littérature ne permettent pas de modéliser tous les types de problèmes. Nous avons proposé une nouvelle contrainte nommée WEIGHTEDSUBCIRCUITS qui permet de chercher un ensemble de circuits élémentaires disjoints sur un graphe valué en excluant certains sommets. Nous avons montré que même la cohérence aux bornes (BC) n'est pas atteignable en temps polynomial. Nous avons également montré que les méthodes efficaces proposées pour filtrer la contrainte WEIGHTEDCIRCUIT ne sont pas directement applicables à la nouvelle contrainte. Nous avons proposé une approche originale pour déterminer une borne inférieure aux coûts des circuits ainsi qu'au coût total de ces circuits. Notre approche repose sur une méthode originale pour déterminer une borne inférieure du coût d'un circuit de Steiner. Ces résultats permettent d'envisager plusieurs possibilités de filtrage pour la contrainte WEIGHTEDSUBCIRCUITS.

Références

- [1] Nicolas Beldiceanu and Evelyne Contejean. Introducing global constraints in CHIP. *math comp.*, 20(12) :97–123, 1994.
- [2] Pascal Benchimol, Willem Jan van Hoeve, Jean-Charles Régin, Louis-Martin Rousseau, and Michel Rueher. Improved filtering for weighted circuit constraints. *Constraints*, 17(3) :205–233, 2012.
- [3] Yves Caseau and François Laburthe. Solving small tsp with constraints. In Lee Naish, editor, *Logic Programming, Proceedings of the Fourteenth International Conference on Logic Programming, Leuven, Belgium, July 8-11, 1997*, pages 316–330. MIT Press, 1997.
- [4] Sylvain Ducomman, Hadrien Cambazard, and Bernard Penz. Alternative filtering for the weighted circuit constraint : Comparing lower bounds for the TSP and solving TSPTW. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3390–3396, 2016.
- [5] Filippo Focacci, Andrea Lodi, and Michela Milano. Cost-based domain filtering. In Joxan Jaffar, editor, *Principles and Practice of Constraint Programming - CP'99, 5th International Conference, Alexandria, Virginia, USA, October 11-14, 1999, Proceedings*, volume 1713 of *Lecture Notes in Computer Science*, pages 189–203. Springer, 1999.
- [6] Filippo Focacci, Andrea Lodi, and Michela Milano. Embedding relaxations in global constraints for solving TSP and TSPTW. *Ann. Math. Artif. Intell.*, 34(4) :291–311, 2002.
- [7] Filippo Focacci, Andrea Lodi, Michela Milano, and Daniele Vigo. Solving TSP through the integration of OR and CP techniques. *Electronic Notes in Discrete Mathematics*, 1 :13–25, 1999.
- [8] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18(6) :1138–1162, 1970.
- [9] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees : Part II. *Math. Program.*, 1(1) :6–25, 1971.
- [10] Frank K Hwang, Dana S Richards, and Pawel Winter. *The Steiner tree problem*, volume 53. Elsevier, 1992.
- [11] Latife Genç Kaya and John N Hooker. A filter for the circuit constraint. In *International Conference on Principles and Practice of Constraint Programming*, pages 706–710. Springer, 2006.
- [12] Gilles Pesant, Michel Gendreau, Jean-Yves Potvin, and Jean-Marc Rousseau. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, 32(1) :12–29, 1998.
- [13] Dimitri Watel. *Approximation de l'arborescence de Steiner. (Approximation of the Directed Steiner Tree Problem)*. PhD thesis, Versailles Saint-Quentin-en-Yvelines University, France, 2014.