



Conception d'une contrainte globale de chemin

Hadrien Cambazard, Eric Bourreau

► To cite this version:

Hadrien Cambazard, Eric Bourreau. Conception d'une contrainte globale de chemin. JNPC 2004 - 10e Journées nationales sur la résolution pratique de problèmes NP-complets, 2004, Angers, France. pp.107-121. hal-00448531

HAL Id: hal-00448531

<https://hal.science/hal-00448531>

Submitted on 10 Apr 2024

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.

Conception d'une contrainte globale de chemin

Hadrien Cambazard

École des Mines de Nantes
4 rue Alfred Kastler – BP 20722
F-44307 Nantes Cedex 3, France

Éric Bourreau

Bouygues e-lab
1 av. Eugène Freyssinet
F-78061 St Quentin en Yvelines Cedex, France

Résumé

Les contraintes globales sont rapidement devenues des ingrédients clés dans l'efficacité des mécanismes de propagation en programmation par contraintes. Elles présentent les avantages d'offrir à la fois des outils de modélisation et de résolution performants pour des sous-problèmes récurrents. Nous présentons ici une étude théorique d'une contrainte globale de chemin élémentaire. Ce problème n'a reçu que peu d'attention de la part de la communauté contraintes tout en étant au cœur de nombreuses applications. Il constitue un problème très difficile, sans qu'il soit nécessaire de considérer des dimensions supplémentaires comme le coût ou la capacité.

1 Introduction

Les contraintes globales répondent à des problématiques spécifiques en encapsulant des algorithmes de haut niveau dédiés à ces sous-problèmes. Les inférences réalisées par ces algorithmes permettent alors d'élaguer efficacement l'espace de recherche en maintenant un degré de consistance plus fort que l'ensemble des contraintes élémentaires correspondantes [BC94]. Leur conception vise l'efficacité comme la réutilisabilité et la difficulté est de se situer exactement au bon niveau sémantique pour répondre à ces deux objectifs. En effet, ces contraintes offrent à la fois des outils de modélisation de haut niveau pour des problématiques récurrentes (proche des problématiques pratiques, sans qu'il soit nécessaire de recourir à des artifices de modélisation) et des mécanismes de filtrage plus performants (sans descendre trop bas vers des contraintes atomiques, parfaites au sens de la propagation mais ayant perdu le cœur de la problématique globale qu'elles

expriment) [BH03]. Nous nous sommes intéressés à une contrainte appliquée au problème de maintien d'un chemin élémentaire. Notre démarche commence par évaluer différentes sémantiques pour faire de la contrainte un bon outil de modélisation. La contrainte de chemin étant basée sur un graphe support, nous évaluons les intérêts respectifs d'une description sous forme de variables domaines modélisant les nœuds ou alors les arcs du graphe. Une fois cette étape validée, nous nous concentrons sur la pleine compréhension des concepts sous jacents à la définition d'un chemin et à quelques propriétés clefs pour la propagation : la connexité, l'ordre du cheminement ainsi que le degré des nœuds visités sont ainsi mis en exergue. Une fois les leviers de propagation isolés (la mise en évidence d'arcs impossibles et de nœuds obligatoires), les algorithmes de filtrage sont élaborés de manière théorique dans un premier temps (afin de valider la consistance complète) puis plus algorithmique (afin d'identifier la complexité). La suite de cet article commence par une section motivant la conception de pattern de modélisation pour les contraintes globales. La section 3 fournit la sémantique de la contrainte de chemin et la section 4, des définitions utiles pour la construction des techniques de filtrage. La principale section, section 5, concerne le filtrage, restreint dans un premier temps aux graphes acycliques, afin d'être complet dans la propagation, puis étendu au cas général (pouvant être NP-difficile). Enfin nous concluons à la section 6 où des perspectives d'utilisations de la contrainte sont fournies.

2 Motivations

La conception d'un algorithme de filtrage pour une contrainte globale soulève de nombreuses questions. Une fois la problématique sous jacente isolée et la sémantique proprement décrite, il s'agit avant tout de cerner la complexité de l'algorithme de consistance et de savoir où sa complétude bascule. Comment concevoir un algorithme de filtrage pour une contrainte globale ? Nous présenterons ici une démarche s'appuyant sur ces différentes étapes :

- une approche théorique à travers l'identification de certaines propriétés utiles au filtrage;
- la traduction de ces propriétés dans un cadre algorithmique;
- les limites : face à des problèmes trop durs, on s'appuie sur une ou plusieurs relaxations, permettant d'obtenir des règles de propagation incomplètes mais utiles.

Nous nous sommes intéressés à une brique de modélisation sous-estimée dans de nombreuses contraintes : la validation d'un chemin élémentaire dans un graphe. Ces problèmes apparaissent en effet fréquemment comme sous problèmes d'applications réelles dans un grand nombre de domaines : conception de réseau, tournée de véhicules, ... Nous pensons qu'il est intéressant de décrire complètement la structure de filtrage associée et la nature des problèmes auxquels on se heurte. Bien qu'il se pose en pratique de nombreuses dimensions supplémentaires : capacité, fenêtre de temps, distance, etc. les problèmes de cheminement sous jacents à résoudre sont déjà assez difficiles au niveau du chemin simple sans avoir à introduire ces dimensions supplémentaires. Il s'agit donc d'identifier et d'apporter des réponses à des problèmes se trouvant au cœur de différentes contraintes globales plus complexes pour fournir des briques réutilisables.

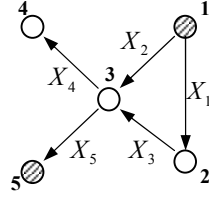


FIG. 1 – Illustration d'un chemin allant de 1 à 5 (nœuds grisés).

3 Sémantique

Définition 1 Soit un graphe $G(V,E)$ avec $n = |V|$ et $m = |E|$. On considère un nœud source p et un nœud puits q appartenant à V ainsi qu'un ensemble X de m variables booléennes associées à chaque arc telle que $X_i = \{0,1\}$ si l'arc est ou non emprunté. La contrainte $Path(X_1, X_2, \dots, X_m, G, p, q)$ assure que :

- l'ensemble $P = \{X_i | X_i = 1\}$ représente un chemin de p à q dans le graphe G .

On peut poser $Path(X_1, \dots, X_5, G, 1, 5)$ sur le graphe illustré figure 1. Après une propagation initiale, la contrainte est réduite à $Path(\{0,1\}, \{0,1\}, \{0,1\}, 0, 1, G, 1, 5)$. Une solution instanciée se traduit par $Path(0, 1, 0, 0, 1, G, 1, 5)$. On peut noter que trois catégories d'arcs sont représentées. Les arcs obligatoires ($X_i = 1$), impossibles ($X_i = 0$) et possibles ($X_i = \{0,1\}$). La consistance de la contrainte consiste donc à identifier les arcs impossibles et obligatoires.

On peut étendre le problème dans un contexte multi-chemins en posant simultanément plusieurs contraintes de chemins. Ainsi, sur la figure 2, le graphe est étendu à 7 nœuds et doit vérifier $Path_a(Y_a, G, 1, 7)$, $Path_b(Y_b, G, 2, 4)$, $Path_c(Y_c, G, 1, 4)$. Les variables X ont été dupliqués en 3 exemplaires : Y_a , Y_b , Y_c de manière à autoriser plusieurs chemins à circuler sur le même graphe. En ajoutant $X \geq Y_a$, $X \geq Y_b$, $X \geq Y_c$, il est possible de retrouver sur le graphe originel l'information de tous les chemins. On a alors $X = \{\{0,1\}, \{0,1\}, 1, 1, \{0,1\}, \{0,1\}, \{0,1\}, 1\}$.

La modélisation *arc* du chemin [Sel02] à l'aide de variables booléennes ne semble pas pertinente dans un contexte contraintes mais permet de résoudre des problèmes de multi-chemins. Une alternative consiste à réaliser un modèle *nœud* [Bou99] utilisant une variable domaine par nœud $S_i = 0 \dots n$ indiquant le successeur du nœud i dans le chemin ou 0 si le nœud n'appartient pas au chemin. Dans une modélisation du chemin par nœuds, on aurait $P_a(S_1, \dots, S_7, G, 1, 7)$ donc des domaines initiaux $S_1 = \{0, 2, 3\}$, $S_2 = \{0, 3\}$, $S_3 = \{0, 4, 5, 6\}$, $S_4 = \{0\}$, $S_5 = \{0, 7\}$, $S_6 = \{0, 5\}$, $S_7 = \{0\}$. Après une réduction initiale, on peut déduire que $S_3 = \{5, 6\}$. L'absence de 0 dans le domaine de S_3 nous indique que le nœud 3 est un nœud obligatoire pour le chemin.

Cette information supplémentaire par rapport à la modélisation *arc* semble intéressante. En revanche, une difficulté se pose pour modéliser un problème multi-chemins. Il est impossible de partager les variables pour plusieurs chemins sur un même graphe (une variable domaine ne pouvant prendre qu'une seule et unique valeur, un nœud ne peut

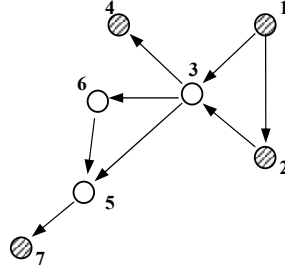


FIG. 2 – *Illustration du problème multi-chemin. On cherche à contraindre l'existence de 3 chemins entre 1 et 7, entre 2 et 4 et entre 1 et 4 partageant les mêmes variables.*

pas se retrouver le croisement de plusieurs chemins).

On retiendra donc une sémantique s'appuyant sur des variables booléennes représentant l'appartenance d'un arc au chemin. L'information, plus riche, liée aux nœuds obligatoires sera exploitée au cœur de la contrainte.

4 Définitions

Nous expliciterons ici les notions d'arc consistance [Mac77] dans le cadre d'une contrainte de chemin ainsi que les propriétés du chemin exploitées au cœur de l'algorithme de filtrage.

4.1 Arc consistance

On raisonnera par la suite sur les nœuds obligatoires et les arcs impossibles, c'est l'information la plus fine qui permet de retrouver l'information des arcs obligatoires et des nœuds impossibles. En effet, la détection d'un arc obligatoire est la conséquence de la détection de deux nœuds obligatoires reliés par un arc unique. De la même manière, un nœud impossible correspond à un ensemble d'arc impossibles (les arcs entrants et sortants du nœud en question). **Les éléments obligatoires d'un chemin sont donc couverts à travers les nœuds (*ker* dans la définition 2), les éléments impossibles sont couverts à travers les arcs (*imp* dans la définition 2).** Les propriétés introduites par la suite s'appuieront donc sur ces éléments uniquement car ils suffisent pour atteindre l'arc consistance.

Définition 2 Soit un graphe $G(V,E)$ avec $n = |V|$, $m = |E|$ et X , un ensemble de m variables booléennes. On définit sur X :

- $\text{ker}(X)$: l'ensemble des nœuds de passages obligatoires tel que :

$$\forall i \in \text{ker}(X), \exists e = (i,j) \in E, X_e \neq 0$$

$$\text{et } \exists e' = (j',i) \in E, X_{e'} \neq 0$$
- $\text{imp}(X)$: l'ensemble des arcs de passages impossibles tel que :

$$\forall e \in \text{imp}(X), X_e = 0$$

Au point de choix courant, le maintien de la consistance se borne à ajouter des éléments à $\ker(X)$ et à $\text{imp}(X)$. Lorsque deux nœuds obligatoires ne sont plus reliés que par un arc unique i , on a bien sûr $X_i = 1$. La recherche de solutions satisfaisant la contrainte consiste à réduire l'ensemble des éléments possibles et augmenter celui des éléments obligatoires jusqu'à obtention d'un chemin valide.

Définition 3 *Une consistance complète assure que :*

- Pour tout arc e de $E \setminus \text{imp}(X)$, il existe un chemin P passant par $\ker(X)$ et e ;
- Pour tout nœud $i \in V \setminus \ker(X)$, il existe un chemin P passant par $\ker(X)$ sans passer par i .

4.2 Propriétés du chemin

Un chemin élémentaire [GM95] dans un graphe est défini de la façon suivante :

Définition 1 *Un chemin de longueur k est une séquence de k arcs : $P = \{u_1, u_2, \dots, u_k\}$ avec :*

$$\begin{aligned} u_1 &= (i_0, i_1) \\ u_2 &= (i_1, i_2) \\ u_k &= (i_{k-1}, i_k) \end{aligned}$$

Un chemin est donc une séquence d'arcs telle que :

- Chaque arc u_r ait une extrémité commune avec l'arc u_{r-1} et l'autre extrémité avec l'arc u_{r+1} .
- Tous les arcs soient orientés dans le même sens.

Un chemin élémentaire est alors un chemin tel qu'en le parcourant, on ne rencontre pas deux fois le même sommet. Dans un chemin élémentaire, tous les sommets sont donc de degré 2 au plus.

Cette définition met en évidence trois éléments qui vont se trouver au cœur de notre démarche pour concevoir un algorithme de filtrage : un chemin est un ensemble **connexe** et **ordonné** d'éléments, au plus de **degré 2** quand il s'agit d'un chemin élémentaire. On introduira par la suite un filtrage fondé sur ces trois propriétés du chemin.

5 Filtrage

Le filtrage sera abordé dans le cadre de graphes orientés. Deux problématiques seront distinguées : les graphes acycliques et les graphes quelconques. Nous essaierons de mettre en évidence l'origine de la complexité dans le second cas et de fournir une relaxation s'appuyant sur les résultats donnés dans le cas acyclique.

5.1 Graphes orientés acycliques

On note $<$ la relation binaire définie en extension par le graphe acyclique G . On a $i < j$ pour un arc (i,j) . La transitivité de $<$ est notée $\prec : \exists x, \exists y, x < y$, et $\exists z, y < z \Leftrightarrow x \prec z$. Ce qui signifie que x est avant z dans l'ordre topologique des nœuds de G . Nous désignerons par p et q , l'origine et la destination du chemin. Sur les figures suivantes, pour illustrer nos propriétés, les arcs impossibles seront indiqués en pointillés et les obligatoires en hachés.

5.1.1 Approche théorique

Élimination des arcs impossibles

Il n'existe que deux raisons pour caractériser des arc impossibles : la connexité (propriété 1) et le rang par rapport aux nœuds obligatoires (propriété 2).

Propriété 1 *Un arc (x,y) de E tel que $\neg(p \prec x)$ ou $\neg(y \prec q)$ est impossible.*

Il s'agit de la définition de la connexité.

Propriété 2 *Un arc (x,y) de E tel que : $\exists O \in \ker(X)$, avec $\neg(y \prec O)$ et $\neg(O \prec x)$ est impossible.*

Preuve : Soit un chemin C de p à q passant par O . Deux positions sont possibles par rapport à O pour tout arc (x,y) de C :

- $x < y \prec O$ ce qui contredit $\neg(y \prec O)$ ou
- $O \prec x < y$ ce qui contredit $\neg(O \prec x)$

Un arc vérifiant la propriété 2 ne peut donc pas appartenir à C . C étant quelconque, un arc vérifiant la propriété 2 ne peut appartenir à aucun chemin passant par O .

Tout arc qui permet de passer *par dessus* O et dont l'une des extrémités au moins ne serait pas comparable à O (au sens de $\prec$) est donc un arc impossible.

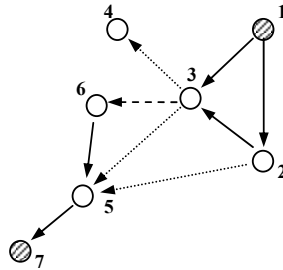


FIG. 3 – Illustration des propriétés. L'arc $(3,4)$ est impossible selon la propriété 1. En supposant l'arc $(3,6)$ obligatoire, on en déduit par la propriété 2 que les arcs $(3,5)$ et $(2,5)$ sont impossibles. Le nœud 5 est obligatoire par la propriété 3, donc l'arc $(5,7)$ également.

Théorème 1 *Soit les ensembles d'arcs impossibles $E1$ et $E2$:*

- $E1 = \{e \in E \mid e \text{ vérifie 1}\}$

$$- E2 = \{e \in E \mid e \text{ vérifie } 2\}$$

$$imp(X) = E1 \cup E2$$

Preuve : Il s'agit de montrer qu'un arc est impossible si et seulement si il vérifie les propriétés 1 ou 2.

- $\Leftarrow$: Déjà démontrée par les preuves précédentes. On sait qu'un arc vérifiant 1 ou 2 est impossible.
- $\Rightarrow$: On procédera par l'absurde en supposant sur un graphe G l'existence d'un arc impossible (x,y) pour un chemin allant de p à q passant par les éléments obligatoires et ne vérifiant ni 1 ni 2.
 (x,y) ne vérifiant pas 1, on a donc $(p \prec x)$ et $(y \prec q)$. Il existe donc une séquence d'arcs $\{u_p, \dots, u_q\}$ avec un arc $u_i = (x,y)$ (cf définition du chemin au 4.2). Par hypothèse, l'existence de nœuds obligatoires indique qu'il existe un arc $u_j = (i_{j-1}, O)$ (et également un arc $u_{j+1} = (O, i_{j+1})$) pour tout nœud O . u_i possède donc 2 positions possibles dans la séquence par rapport à tous les u_j . Soit u_i est avant u_j et dans ce cas, on a $(y \prec O)$, soit il est après et on a $(O \prec x)$. Ceci étant valable pour tout O , on obtient une contradiction avec la propriété 2.

Les arcs impossibles le sont donc uniquement pour des raisons de connexité et d'ordre par rapport aux éléments obligatoires du chemin. Ces deux propriétés servent à la conception d'un algorithme de filtrage et garantissent sa complétude concernant la détermination des arcs impossibles.

Ajout de nœuds obligatoires

Un nœud obligatoire est un nœud par lequel passent tous les chemins de p à q . Il s'agit d'un point d'articulation du graphe. Son retrait de G déconnecte p et q [Ber70].

Propriété 3 : O est obligatoire si et seulement si : $\forall (x,y) \in E, y \prec O$ ou $O \prec x$

Par définition, si O est obligatoire, tous les chemins passent par O donc tout arc (x,y) est situé avant ou après O . On obtient donc

$$ker(X) = \{n \in N \mid n \text{ vérifie la propriété } 3\}$$

5.1.2 Approche algorithmique

La détection des arcs impossibles se fait à travers le respect des deux propriétés 1 et 2. La connexité doit être vérifiée de p à tout nœud et de tout nœud à q . Une manière compacte d'appréhender ce problème consiste à ajouter un arc virtuel (q,p) et à procéder à une décomposition en composantes fortement connexes. Immédiatement, tous les arcs non inclus dans la composante fortement connexe de p et q sont impossibles. La vérification de la propriété 1 s'effectue en $O(M)$ ce qui correspond au coût de la décomposition en composantes fortement connexes [GM95].

Pour identifier les arcs répondant à la propriété 2, il faut procéder à un tri topologique du graphe en $O(M + N)$ [CLRS01]. L'analyse des rangs des extrémités d'un arc par rapport à celui du dernier point de passage obligatoire traité permet d'éliminer les arcs impossibles qui *sautent* les points de passages obligatoires.

Enfin la propriété 3 se traduit par la détection de points d'articulation dans un graphe acyclique, ce qui est possible en $O(M)$ [GM95].

Théorème 2 *L'arc-consistance pour une contrainte de chemin dans le cas acyclique peut être assurée en $O(M + N)$.*

5.2 Graphes orientés

Trouver un chemin passant par des points de passages obligatoires est un problème NP-Complet [FHW80]. Assurer l'arc-consistance de la contrainte de chemin est par conséquent NP-Complet. Nous chercherons des relaxations intéressantes du problème pour la détermination des éléments impossibles et obligatoires. Caractériser les arcs impossibles nous permet notamment de comprendre l'origine de la difficulté et de proposer une relaxation pertinente.

5.2.1 Approche théorique

La difficulté tient dans la combinatoire induite par l'ordre des éléments de ker . Il faudrait être en mesure de déterminer les arcs impossibles et nœuds obligatoires pour toutes les permutations des éléments obligatoires (tous les ordres possibles). L'introduction des cycles remet en question la complétude des propriétés précédentes car il n'y a plus d'ordre implicite (fourni par le tri topologique dans le cas acyclique) entre les éléments obligatoires.

Par ailleurs, un deuxième problème se pose, celui du caractère élémentaire du chemin. L'absence de cycle garantissait implicitement son respect, comment le vérifier à présent ?

5.2.2 Elimination des arcs impossibles

L'idée de retrouver un ordre entre les éléments obligatoires nous amène à une première relaxation du problème consistant à travailler sur le graphe acyclique des composantes fortement connexes (CFC).

Soit $G'(V', E')$ le graphe des composantes fortement connexes de G , p' et q' les composantes incluant p et q . Une composante $x' \in V'$ est dite obligatoire si elle contient un élément obligatoire, on peut ainsi définir le nouveau noyau : $ker'(X)$ qui contient les CFC obligatoires. De la même manière $imp'(X)$ désigne les arcs impossibles appartenant à E' .

Les arcs impossibles obtenus pour des raisons de connexité dans le cas acyclique sont toujours identifiables à travers notre relaxation :

Propriété 1 bis *Un arc (x, y) incident ou appartenant à une composante $c_i \in V'$ tel que $\neg(p' \prec c_i)$ ou $\neg(c_i \prec q')$ est impossible.*

Preuve : Le graphe des CFC étant acyclique, on se ramène à la propriété 1. Deux cas se présentent pour un arc (x, y) :

- Soit $(x, y) \in G'$ est incident à une CFC, (*i.e.* dont l'une des extrémités seulement appartient à une CFC) et c'est l'application directe de la propriété 1.
- Soit (x, y) est inclus dans c_i . La composante c_i étant éliminée par suppression de ces arcs incidents (cas précédent), les arcs qui la composent sont impossibles.

Les arcs impossibles en raison de leur ordre par rapport aux éléments obligatoires ne se retrouvent qu'en partie dans la relaxation :

Propriété 2 bis *Un arc (x',y') de E' tel que : $\exists O' \in \ker'(X)$, avec $\neg(y' \prec O')$ et $\neg(O' \prec x')$ est impossible.*

Preuve : L'unique différence avec la propriété 2 précédente réside dans le fait que plusieurs arcs peuvent relier deux CFC. Si ce lien est impossible, ils le sont tous. En revanche, la réciproque n'est pas vraie (voir figure 4) : bien que l'arc (c_1, c_2) soit possible car $(1,3)$ est possible, l'arc $(1,6)$ est impossible si les nœuds 3 et 6 sont obligatoires.

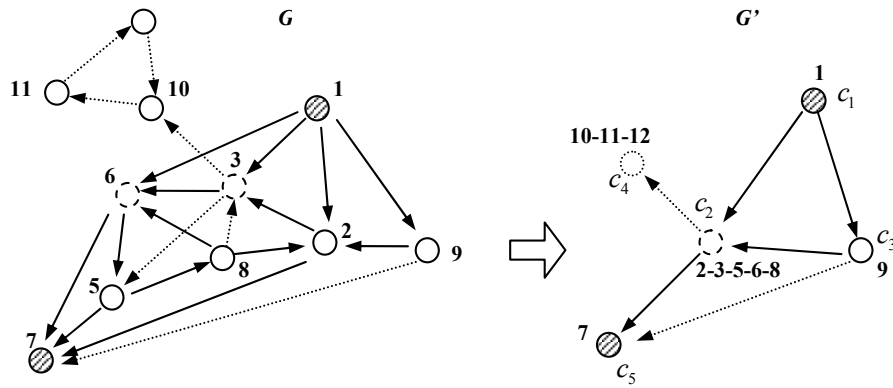


FIG. 4 – *Illustration. G' est le graphe des CFC de G . Les arcs $(11,12)$, $(12,10)$, $(10,11)$ et $(3,10)$ sont impossibles pour des raisons de connexité. L'identification de la CFC c_4 sur G' par la propriété 1bis permet de déduire cette information. En supposant le nœud 6 obligatoire, on peut en déduire par la propriété 2bis que $(9,7)$ est impossible pour des raisons d'ordre avec les éléments obligatoires (ici c_2). Si on suppose à présent que les nœuds 3 et 6 sont obligatoires, l'arc $(1,6)$ devient impossible mais échappe au filtrage sur G' . Enfin, l'arc $(8,3)$ est également impossible pour des raisons de degré et s'inscrit dans le cadre de la propriété 4.*

Abordons à présent le nouveau problème du degré. Certains arcs se révèlent impossibles car ils entraînent la violation du caractère élémentaire du chemin sur un ensemble de nœuds. On note par $deg(x)$ le degré entrant du nœud x associé à un chemin P passant par x . Il s'agit du nombre d'arcs de P ayant pour extrémité x . L'arc $(8,3)$ de la figure 4 en est un bon exemple. Emprunter l'arc $(8,3)$ entraîne $deg(6) = 2$ ou $deg(5) = 2$. Autrement dit, les chemins de p à 3 et de 8 à q s'intersectent tous en 6 ou en 5, ce qui rend cet arc $(8,3)$ impossible. La propriété 4 exprime cette idée.

Propriété 4 *Un arc (x,y) est impossible si :*

$$\forall C_1 \in Ch_{p \rightsquigarrow x}, C_2 \in Ch_{y \rightsquigarrow q} \text{ on a : } C_1 \wedge C_2 \neq \emptyset$$

La question est de savoir si on peut identifier un tel arc (x,y) dans un temps polynomial. On se rend compte aisément que le problème consiste à identifier ces ensembles d'articulations (*cutset*) où le degré de certains nœuds sont violés. Pour comprendre le problème nous introduisons ici la notion de *s-t cutset* [AMO93]. Il s'agit d'un ensemble

de nœuds dont le retrait du graphe déconnecte la paire (s, t) de sorte qu'aucun chemin ne permette d'atteindre t depuis s . Si on note $Art_{(s \rightsquigarrow t) \setminus z}$ l'ensemble des s - t cutsets minimaux dans le sous-graphe $G_{V \setminus z, E}$, cela revient dans notre exemple à examiner les ensembles d'articulations minimaux entre p et 8 ainsi que 3 et q . On a $Art_{p \rightsquigarrow 8 \setminus 3} = \{\{6\}, \{5\}\}$ et $Art_{3 \rightsquigarrow q \setminus 8} = \{\{6, 5\}\}$. On remarque que l'intersection de toute paire d'éléments de $Art_{p \rightsquigarrow 8 \setminus 3}$ et $Art_{3 \rightsquigarrow q \setminus 8}$ est non vide. Tous les chemins se croisent sur ces ensembles d'articulations ce qui explique le viol du degré.

Le problème d'identifier tous les s - t cutset minimaux d'un graphe cyclique est très difficile et bien connu dans le monde des télécommunications pour être un problème NP-Difficile (théorème de Menger de 1927). Ce type d'arc n'est pas pris en compte (même partiellement) par la relaxation et nous l'envisagerons à travers des règles élémentaires de filtrage.

5.2.3 Ajout de nœuds obligatoires

Une fois l'ensemble des arcs impossibles éliminés complètement, on pourrait faire l'hypothèse que tout arc appartient à au moins un chemin de p à q passant par les éléments obligatoires (ker). Sous cette hypothèse, les nœuds obligatoires restants sont les points d'articulation entre p et q . Malheureusement, on connaît la difficulté à éliminer tous les arcs impossibles. Or, en leur présence, si les points d'articulation entre p et q restent des nœuds obligatoires, ce ne sont plus les seuls. La relaxation s'appuyant sur le graphe des CFC semble ici moins intéressante. En effet, si un arc impossible sur G' l'est également sur G , une CFC obligatoire ne nous indique pas le nœud obligatoire même si c'est déjà une information supplémentaire. Il semble plus intéressant ici de s'appuyer sur une autre relaxation évitant de se confronter au problème de l'ordre des éléments de ker .

Propriété 3 bis : Si $\exists O \in ker(X)$, pour lequel x est un point d'articulation entre p et O ou O et q alors x est obligatoire¹.

Preuve : On note Ch l'ensemble des chemins de p à q passant par ker , c'est-à-dire par tous les éléments obligatoires. Ch_O désigne l'ensemble des chemins passant par un seul élément obligatoire O . On peut ainsi écrire que $Ch = \bigcap_{O \in ker} Ch_O$ et donc $\forall O \in ker, Ch \subseteq Ch_O$. Considérons un élément $O \in ker$ telle que x soit un point d'articulation entre p et O (Le cas O et q est identique). x étant un point d'articulation entre p et O , $\forall P \in Ch_O, x \in P$ donc $\forall P \in Ch, x \in P$ puisque $Ch \subseteq Ch_O$. Ainsi tous les chemins de p à q passant par ker , passent par x , donc x est obligatoire.

On a alors $\{x \in N \mid x \text{ vérifie 3bis}\} \subseteq ker(X)$ ce qui correspond bien à une relaxation du problème.

5.2.4 Approche algorithmique

Devant la NP-difficulté du problème nous proposons deux types de relaxation de l'arc-consistance dédiées d'une part, aux arcs impossibles et d'autre part aux nœuds obligatoires. La première est fondée sur le traitement du graphe des composantes fortement connexes. La deuxième sur l'analyse des éléments de ker indépendamment les uns des autres.

Définition 4 La consistance relachée dite R -consistance s'écrit :

- Pour tout arc e de $E' \setminus imp'(X)$, il existe un chemin P passant par e .

1. x est ici un p - O ou O - q cutset réduit à un élément

- Pour tout nœud $i \in V' \setminus \ker'(X)$, il existe un chemin P passant par $\ker'(X)$ sans passer par i .
- Pour tout nœud $i \in V \setminus \ker(X)$, il existe un chemin P passant par chaque élément de $\ker$ sans passer par i .

Une décomposition en composantes fortement connexes s'effectue en $O(M)$ [GM95], l'arc consistance sur le graphe acyclique des CFC se fait en $O(M + N)$ (Le nombre de CFC est au pire le nombre de noeuds). La détection des nœuds vérifiant 3bis impose pour chaque nœud du graphe d'examiner sa connexité avec tous les éléments de $\ker$. Pour un nœud donné, il s'agit d'examiner la connexité de p à tous les éléments de $\ker$ et de tous les éléments de $\ker$ à q dans le sous-graphe $G'' = G_{V \setminus x, E}$. En procédant à une exploration dans le graphe des successeurs et des prédécesseurs de G'' , on obtient cette information en $O(M)$, ce qui nous permet d'affirmer :

Théorème 3 *La R-consistance pour une contrainte de chemin peut être assurée en $O(MN)$.*

Extensions. Au sein de chaque composante fortement connexe, des règles de propagation simples peuvent être mise en oeuvre pour fournir des réponses locales mais efficaces. Ainsi, à l'image des lois de Kirchoff, un arc devient obligatoire lorsque l'une de ses extrémités est obligatoire et qu'il est le seul arc sortant ou entrant de l'extrémité en question. L'élimination des cycles sur les *bouts* de chemins instanciés est prise en compte par l'élimination d'arcs retour sur les éléments contigus du chemin. L'application de ces deux règles garantit le caractère élémentaire du chemin mais n'offre pas de prise en compte globale des arcs impossibles pour des raisons de degré.

6 Bilan

Une synthèse de l'ensemble des résultats est donnée au tableau 1. Les arcs impossibles et noeuds obligatoires peuvent être identifiés de façon complète dans un contexte acyclique. Deux types d'arcs impossibles pour des raisons d'ordre de cheminement et de connexité ont été mis en évidence. Dans le cas cyclique, si les arcs impossibles pour des raisons de connexité sont identifiables de manière polynomiale, l'ordre de cheminement est plus difficile à établir. Pour cette raison, deux types de relaxations ont été proposées, la première s'appuie sur la décomposition du graphe en composantes fortement connexes et permet de retrouver une partie des arcs impossibles pour des raisons d'ordre alors que la deuxième est destinée à l'identification des noeuds obligatoires. Enfin, les arcs qui entraînent la violation du degré du chemin sont pris en compte à travers l'utilisation de règles de filtrage.

7 Conclusion, Perspectives

Nous avons présenté une contrainte globale de chemin élémentaire. La progression, dans la construction de l'algorithme de filtrage pour réaliser la réduction des domaines des variables associées, a suivi un canevas précis :

- une description de la sémantique claire et justifiée pour la contrainte : contrainte de chemin élémentaire sur un graphe représenté par une liste de variables booléennes

	Arcs impossibles		Noeuds obligatoires	AC	RC
Graphe acyclique	Connexité	$O(M)$	$O(M)$	$O(M + N)$	-
	Ordre	$O(M + N)$			
Graphe cyclique	Connexité	$O(M)$	AC: NP-Complet RC: $O(MN)$	NP-Complet	$O(MN)$
	Ordre	AC			
		RC			
	Degré	NP-Complet			

TAB. 1 – *Synthèse des résultats*

placées sur les arcs. Les leviers utiles à la propagation sont les arcs impossibles et les nœuds obligatoires.

- une approche théorique afin d’identifier des propriétés clefs : un chemin peut se décrire par une séquence d’arcs. Des propriétés, liées à la connexité, à l’ordre de visite ainsi qu’au degré des nœuds empruntés ont été abordées. Le graphe support a d’abord été considéré acyclique avant d’être abordé dans un contexte plus général. Notons également que l’identification de ces propriétés à l’origine du filtrage est un bon point de départ à l’introduction d’explications [Jus03] car elles mettent en évidence les raisons théoriques qui justifient l’ajout de noeuds obligatoires et le retrait d’arcs impossibles.
- une traduction de ces propriétés dans un cadre algorithmique : un tri topologique et la recherche de composantes fortement connexes permettent de prendre en compte les différentes facettes inhérentes à la problématique du chemin. Dans le cas cyclique, la contraction des composantes fortement connexes du graphe permet de se ramener au cas acyclique.
- enfin, la consistance complète étant inaccessible en temps polynomial dans le cas général, des cas polynomiaux ont été mis en évidence. Le cas acyclique est cependant résolu complètement; le cas cyclique possède des lacunes dans la propagation sur certains types d’arcs mais des relaxations pertinentes ont été identifiées.

L’implémentation intégrant un souci d’incrémentalité est l’étape suivante. On trouve déjà dans [AHR⁺90] une version incrémentale en insertion et destruction d’arcs du tri topologique. En revanche, la décomposition en composantes fortement connexes est plus délicate dans sa version incrémentale. Nous nous sommes ici focalisés sur la démarche théorique et algorithmique préliminaire. C’est à notre sens une composante critique de l’élaboration d’une contrainte globale qui se heurte à des problèmes NP-Complets.

Les applications de cette contrainte de chemin sont très variées. Citons à titre d’exemple les problématiques de Recherche Opérationnelle ayant trait au problème du voyageur de commerce (Traveling Salesman Problem). Le problème pur du voyageur de commerce ne s’y prête pas particulièrement car les modélisations actuelles s’appuient sur des permutations (*alldifferent* + *nocycle*) [CL97] car le chemin est hamiltonien. Par ailleurs il s’agit d’un pur problème d’optimisation et la contrainte pourrait se montrer plus efficace sur des problèmes plus contraints du point de vue du chemin. Par exemple, l’ajout de fenêtres de temps permettrait peut-être de raisonner sur les chemins possibles rapidement et de réaliser des inférences utiles. De même, les problèmes de tournées *VRP*

(Vehicule Routing Problem) et leur multiple déclinaisons entrent dans cette catégorie. Ainsi une autre branche d'extension de la contrainte consisterait à prendre en compte le coût sur les arcs telle que la contrainte de plus court chemin introduite par Sellmann [Sel03]. D'une manière générale, les problèmes de chemins se posent avec des dimensions (ou ressources) supplémentaires comme la capacité dans les problèmes de *CVRP* (Capacited Vehicule Routing Problem) ou de conception de réseau ([BCP⁺02]); On peut citer également les fenêtres de temps ou toute autre dimension permettant de caractériser le chemin (par exemple la sécurité pour un problème de dimensionnement de réseau). Les problèmes de multifiots monoroutés mettent d'ailleurs en jeu de nombreux problèmes de chemins élémentaires; Il existe aussi des modélisations faisant intervenir des problèmes de chemins sous forme de contraintes symboliques dans un contexte de programmation linéaire en nombres entiers. On trouvera dans [ABE⁺03] les détails concernant l'apport de telles contraintes et la mention d'une contrainte de chemin. L'intérêt pratique de la contrainte pourrait se révéler sur des problèmes de cheminement contraints et d'une taille raisonnable. En effet, les problèmes cités précédemment peuvent s'avérer très difficiles sur de toutes petites instances sans que la difficulté soit inhérente au chemin lui-même mais à toutes ses dimensions supplémentaires.

Ainsi, si les applications sont nombreuses, le problème a étonnamment reçu peu d'attention sous la forme la plus simple dans la communauté contraintes. Une implémentation de la contrainte permettra de remédier à ce manque.

Références

- [ABE⁺03] E. Althaus, A. Bockmayr, M. Elf, T. Kasper, M. Jünger, and K. Mehlhorn. Scil-symbolic constraints in integer linear programming. In E.K. Burke and P. De Causmaecker, editors, *10th European Symposium on Algorithms (ESA'02)*, number 2461 in Lecture Notes in Computer Science, pages 75–87. Springer-Verlag, 2003.
- [AHR⁺90] Bowen Alpern, Roger Hoover, Barry K. Rosen, Peter F. Sweeney, and F. Kenneth Zadeck. Incremental evaluation of computational circuits. In *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pages 32–42. Society for Industrial and Applied Mathematics, 1990.
- [AMO93] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Networks flows*. Prentice Hall, 1993.
- [BC94] N. Beldiceanu and E. Contejean. Introducing global constraints in CHIP. *Mathematical and Computer Modelling*, 12:97–123, 1994.
- [BCP⁺02] R. Bernard, J. Chambon, C. Le Pape, L. Perron, and J.C. Régim. Résolution d'un problème de conception de réseau avec parallel solver. In *JFPLC'02*, 2002.
- [Ber70] C. Berge. *Graphes et hypergraphes*. Dunod, 1970.
- [BH03] Christian Bessière and Pascal Van Hentenryck. To be or not to be ... a global constraint. In *Proceedings of the 9th intern. Conference on the Principles and Practice of Constraint Programming (CP)*, number 2833 in Lecture Notes in Computer Science, pages 789–794. Springer-Verlag, 2003.

- [Bou99] E. Bourreau. *Traitement de contraintes sur les graphes en programmation par contraintes*. PhD thesis, Université Paris 13, March 1999.
- [CL97] Y. Caseau and F. Laburthe. Solving small TSPs with constraints. *International Conference on Logic Programming*, pages 316–330, 1997.
- [CLRS01] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2001.
- [FHW80] S. Fortune, J. Hopcroft, and J. Wyllie. The directed subgraph homeomorphism problem. *Theoretical Computer Science*, 10(2):111–121, 1980.
- [GM95] M. Gondran and M. Minoux. *Graphes et algorithmes*. Eyrolles, 1995. Troisième édition revue et augmentée.
- [Jus03] Narendra Jussien. *The versatility of using explanations within constraint programming*. PhD thesis, Université de Nantes, Nantes, France, 18 September 2003. also available as RR-03-04 research report at École des Mines de Nantes.
- [Mac77] K. Macworth. Consistency in networks of relations. *Artificial Intelligence*, 8(1):99–118, 1977.
- [Sel02] Meinolf Sellmann. *Reduction Techniques in Constraint Programming and Combinatorial Optimization*. PhD thesis, University of Paderborn, 2002.
- [Sel03] Meinolf Sellmann. Cost-based filtering for shorter path constraints. In *Proceedings of the 9th intern. Conference on the Principles and Practice of Constraint Programming (CP)*, number 2833 in Lecture Notes in Computer Science, pages 694–708. Springer-Verlag, 2003.