



**HAL**  
open science

## Condition nécessaire pour la contrainte de partitionnement de graphes par des chemins

Nicolas Beldiceanu, Xavier Lorca

► **To cite this version:**

Nicolas Beldiceanu, Xavier Lorca. Condition nécessaire pour la contrainte de partitionnement de graphes par des chemins. Troisièmes Journées Francophones de Programmation par Contraintes (JFPC07), Jun 2007, Rocquencourt / France. inria-00151230

**HAL Id: inria-00151230**

**<https://inria.hal.science/inria-00151230>**

Submitted on 1 Jun 2007

**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.

# Condition nécessaire pour la contrainte de partitionnement de graphes par des chemins

Nicolas Beldiceanu

Xavier Lorca

École des Mines de Nantes, LINA FRE CNRS 2729, FR – 44307 Nantes Cedex 3  
 {Nicolas.Beldiceanu, Xavier.Lorca}@emn.fr

## Abstract

Étant donné un graphe orienté  $\mathcal{G}$ , le problème des  $K$ -chemins disjoints consiste à trouver une partition de  $\mathcal{G}$  en  $K$  chemins noeuds-disjoints, tel que chaque chemin termine sur un noeud appartenant à un sous-ensemble des noeuds de  $\mathcal{G}$ . Cet article fournit une condition nécessaire, pour le problème des  $K$ -chemins disjoints, combinant (1) la structure du graphe réduit associé au graphe  $\mathcal{G}$ , (2) la structure de chaque composante fortement connexe de  $\mathcal{G}$  vis à vis des relations de dominance entre les noeuds de  $\mathcal{G}$ , et (3) la structure des connexions entre les composantes fortement connexes de  $\mathcal{G}$ . Finalement, nous montrons comment exploiter cette condition nécessaire pour en dériver une contrainte de partitionnement de graphes par des chemins.

## 1 Introduction

Les contraintes de partitionnement sont omniprésentes dans de nombreuses applications pratiques comme les problèmes de *tournées de véhicules* [2] ou de *robustesse dans les réseaux* [17]. Les patterns de partitionnement les plus classiques sont les cycles [3, 10], les arbres [4, 12] ou encore les chemins [13, 15]. Dans le contexte des contraintes de partitionnement par des chemins, les conditions nécessaires prévenant la création de circuits et maintenant le fait que chaque noeud du graphe ne possède pas plus d'un prédécesseur ont déjà été introduites dans la littérature [5, 13, 15]. Cependant, aucune de ces conditions ne considère réellement le nombre de chemins à construire. Le but de cet article est de proposer une condition nécessaire relative au nombre de chemins requis pour partitionner un graphe orienté.

Considérons un graphe orienté  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , une partition de  $\mathcal{G}$  en chemins est une collection de chemins noeuds-disjoints  $P_1 = (V_1, A_1), \dots, P_K = (V_K, A_K)$  dans  $\mathcal{G}$ , dont l'union est représentée par  $\mathcal{V}$ , i.e.,  $V_i \cap V_j = \emptyset$ ,

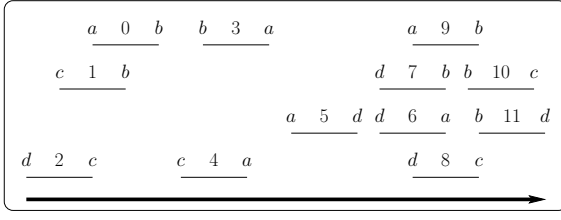
$i \neq j$ , et  $\bigcup_{i=1}^K V_i = \mathcal{V}$ . Le problème des  $K$  chemins noeuds-disjoints [16, 18] (que l'on notera  $K$ -NDP dans la suite) cherche à déterminer une partition de  $\mathcal{G}$  telle que  $K \in [\underline{K}, \bar{K}]$ , où  $\underline{K}$  désigne le nombre minimum de chemins d'une partition de  $\mathcal{G}$  et  $\bar{K}$  est la taille d'un sous-ensemble  $\mathcal{T} \subseteq \mathcal{V}$  de noeuds potentiellement finaux pour chaque chemin.

Trouver une partition en chemins de  $\mathcal{G}$  compatible est un problème généralement NP-complet [9, 16], même lorsque le nombre de chemins est fixé à  $K = 2$ . Cependant, il existe de nombreuses classes de graphes particuliers pour lesquelles ce problème de partitionnement est résoluble en temps polynomial [1, 14, 19]. Dans le contexte d'une contrainte maintenant une partition par des chemins noeuds-disjoints, nous ne pouvons faire aucune hypothèse sur la classe du graphe  $\mathcal{G}$  mais, nous allons proposer une condition nécessaire basée sur la structure de  $\mathcal{G}$  (e.g., le graphe réduit <sup>1</sup> associé à  $\mathcal{G}$ , la relation de dominance [11, 13] entre les noeuds de  $\mathcal{G}$ ).

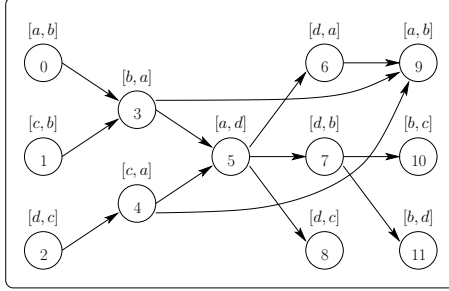
Le problème des  $K$  chemins noeuds-disjoints est utile dans beaucoup d'applications pratiques. L'une d'entre elle consiste à couvrir un ensemble de tâches (e.g., des livraisons, des vols) avec un minimum de ressources (e.g., des camions, des avions). Chaque tâche est représentée comme un intervalle défini par une *date de départ au plus tôt*, une *date de fin au plus tard* et une *durée fixe*. De plus, une tâche possède aussi un *lieu de départ* et une *destination*. Dans une solution valide, une tâche  $t_1$  peut précéder immédiatement une tâche  $t_2$  si (1) la date de fin au plus tôt de  $t_1$  précède la date de fin au plus tard de  $t_2$ , et si (2) la destination de  $t_1$  est identique au lieu de départ de  $t_2$ .

**exemple 1.** Comme exemple de l'application précédente, considérons l'ensemble des tâches fixées représentées par

<sup>1</sup>Le graphe réduit  $G_r$  associé à un graphe orienté  $G$  est dérivé de  $G$  par la transformation suivante : à chaque composante fortement connexe (*scc*) de  $G$  nous associons un noeud de  $G_r$ , à chaque arc de  $G$  connectant deux *scc*'s distinctes correspond un arc de  $G_r$ .



(a)



(b)

FIG. 1: En Partie (a), un ensemble de tâches fixées dans le temps (décrit par l'axe orienté en gras). Chaque tâche  $t$  est décrite par un triplet  $s, i, d$  dont les éléments correspondent respectivement au lieu de départ, l'identifiant de la tâche et le lieu de destination. La Partie (b) représente le graphe correspondant.

la Figure 1a. La Figure 1b fournit le graphe  $\mathcal{G}$  correspondant qui peut être couvert par 6 chemins noeuds-disjoints. Il faut remarquer que le graphe  $\mathcal{G}$  est acyclique puisque toutes les tâches sont complètement fixées dans le temps.

Dans le contexte des contraintes de partitionnement [5, 7, 13], la contribution de cet article est de montrer comment la combinaison d'un raisonnement sur les *flots* et d'un raisonnement sur la relation de *dominance* entre les noeuds du graphe fournit une condition nécessaire pour la contrainte de partitionnement par des chemins. Dans la suite de l'article, nous introduirons tout d'abord dans la Section 2 quelques rappels sur la théorie des flots [6, p.72]. Ensuite, dans la Section 3, une approche basée sur les flots sera étudiée sur le cas des graphes orientés acycliques (DAGs). Puis, la Section 4 généralisera ce modèle au cas non-acyclique, en traitant les "goulots d'étranglement" du graphe réduit : les Sections 4.1 et 4.2 montreront comment déterminer des bornes plus précises sur le nombre de chemins minimum et maximum à partir de (1) la relation de dominance entre les noeuds du graphe  $\mathcal{G}$  et (2) la manière dont les noeuds de deux composants fortement connexes sont reliés entre elles. Finalement, la Section 5 montrera comment exploiter une partie du modèle de flot associé au problème des  $K$  chemins noeuds-disjoints dans le but d'améliorer le filtrage lié à la contrainte de partitionnement par des chemins.

## 2 Préliminaires

Cette section rappelle trois définitions issues de [6, p.72] dans le but de définir les notions classiques de la théorie des flots. De plus, nous rappelons le théorème de Hoffman qui généralise la conservation du flot à tout cocycle d'un réseau. Ces notions seront utilisées tout au long de l'article.

**Définition 1.** Soit  $A$  un ensemble de noeuds d'un graphe  $G = (U, E)$ , soit  $\omega^+(A)$  l'ensemble d'arcs sortant de  $A$ , et soit  $\omega^-(A)$  l'ensemble d'arcs entrant sur  $A$ . Un cocycle est un ensemble d'arcs  $\omega(A) = \omega^+(A) \cup \omega^-(A)$ .

**Définition 2 (Réseau).** Soit  $N = (U \cup \{s, t\}, E)$  un graphe orienté dont chaque arc  $(i, j) \in E$  est défini par une borne inférieure  $\ell_{ij}$  représentée par un entier non-négatif, et une capacité  $c_{ij}$  représentée, elle-aussi, par un entier non-négatif ( $\ell_{ij} \leq c_{ij}$ ). Pour chaque noeud  $i$ , distinct de  $s$  et  $t$ , il existe un chemin dans  $N$  de  $s$  à  $i$  et un chemin de  $i$  à  $t$ . Il y a aussi un arc de  $t$  à  $s$ , appelé arc de retour de  $N$ .

**Définition 3 (Flot).** Un flot dans un réseau  $N = (U \cup \{s, t\}, E)$  est défini par une fonction  $f : E \mapsto \mathbb{N}$  telle que :

1. Contraintes de capacité : quelque soit  $(i, j) \in E$ ,  $\ell_{ij} \leq f(i, j) \leq c_{ij}$ .
2. Conservation du flot :

$$\forall x \in U, \sum_{(i,x) \in \omega^-(\{x\})} f(i, x) = \sum_{(x,j) \in \omega^+(\{x\})} f(x, j) \quad (1)$$

Un flot global  $\mathcal{F}(N)$  d'un réseau  $N$  est fournie par :

$$\mathcal{F}(N) = \sum_{(s,j) \in \omega^+(\{s\})} f(s, j) \quad (2)$$

**Théorème 1 (Hoffman).** Étant donné un réseau  $N = (U, E)$  défini par une fonction  $f : E \mapsto \mathbb{N}$  telle que pour tout  $(i, j) \in E$ ,  $f_{ij} \in [\ell_{ij}, c_{ij}]$ , il existe un flot compatible en  $N$  ssi pour tout cocycle  $\omega(A)$  de  $N$ , on a :

$$\sum_{(i,j) \in \omega^+(A)} c_{ij} - \sum_{(i,j) \in \omega^-(A)} \ell_{ij} \geq 0 \quad (3)$$

## 3 Le problème K-NDP dans le cas de graphes orientés acycliques

Cette section s'intéresse au cas des graphes orientés acycliques pour lesquels le problème K-NDP peut être résolu en temps polynomial [1, 14, 19]. Quand le nombre de chemins  $K$  n'est pas fixé, le point clé dans la résolution d'un tel problème est l'évaluation du nombre de chemins minimum nécessaires pour partitionner le graphe  $\mathcal{G}$ .

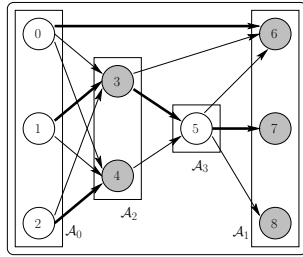


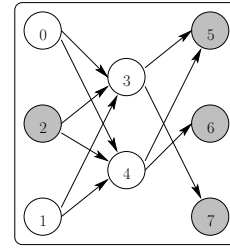
FIG. 2: Un graphe orienté acyclique  $\mathcal{G}$ , où les noeuds potentiellement finaux sont représentés en gris. Les arcs épais représentent une partition en chemins de cardinalité minimale.

Une première piste pour évaluer cette borne inférieure consiste à regarder la taille d'une antichaîne maximale<sup>2</sup> (aussi appelée *largeur*) du graphe  $\mathcal{G}$  (voir [8] pour une caractérisation complète). En effet, une fois que l'antichaîne est traversée, il n'y a aucun moyen de revenir en arrière puisque  $\mathcal{G}$  est acyclique. Cependant, même si la largeur de  $\mathcal{G}$  constitue la borne inférieure réalisable pour le problème de *couverture* par des chemins (voir [14, p.219]), ce n'est pas le cas dans le contexte du problème de partitionnement par des chemins noeuds-disjoints. L'exemple suivant illustre ce point important :

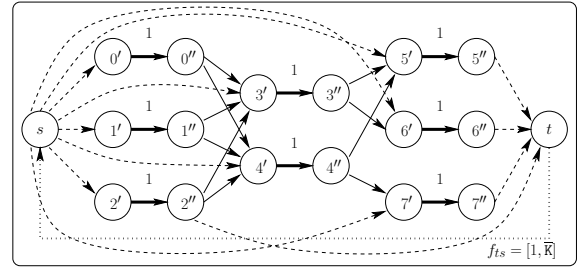
**exemple 2.** Dans la Figure 2, à partir des antichaînes maximales  $\mathcal{A}_0 = \{0, 1, 2\}$  et  $\mathcal{A}_1 = \{6, 7, 8\}$  de  $\mathcal{G}$ , la largeur de  $\mathcal{G}$  est égale à 3 et alors, il existe une couverture de  $\mathcal{G}$  avec 3 chemins (e.g.,  $\langle 0, 6 \rangle$ ,  $\langle 1, 3, 5, 7 \rangle$ ,  $\langle 2, 4, 5, 8 \rangle$ ). Mais, remarquez que dans le contexte du partitionnement, les antichaînes  $\mathcal{A}_2 = \{3, 4\}$  et  $\mathcal{A}_3 = \{5\}$  forment un goulot d'étranglement entre les noeuds de  $\mathcal{A}_0$  et les noeuds de  $\mathcal{A}_1$ . Ainsi, partitionner  $\mathcal{G}$  avec 3 chemins noeuds-disjoints est clairement impossible. En fait, 4 chemins sont nécessaires pour partitionner  $\mathcal{G}$  (e.g.,  $\langle 0, 6 \rangle$ ,  $\langle 1, 3, 5, 7 \rangle$ ,  $\langle 2, 4 \rangle$ , and  $\langle 8 \rangle$ ).

Dans le cadre de graphes orientés acycliques, cette section introduit un modèle de flot classique (Exemple 3) fournissant une condition nécessaire et suffisante pour le problème K-NDP, ainsi qu'une borne inférieure réalisable sur le nombre minimum de chemins noeuds-disjoints nécessaires pour partitionner  $\mathcal{G}$ . Dans cette approche, on doit construire pour chaque K partition en chemins du graphe, un K flot dans un graphe dérivé. Une idée simple est d'ajouter pour chaque noeud  $i$  deux arcs  $(s, i)$  et  $(i, t)$  (où  $s$  et  $t$  sont deux noeuds supplémentaires). Alors, pour chaque chemin  $\langle i_1, \dots, i_n \rangle$  de la K partition, on peut ajouter deux arcs de flot  $(s, i_1)$  et  $(i_n, t)$ , qui permettent de construire un K flot. Le biais de cette construction est que le graphe résultant admet toujours un flot nul, ce qui conduit à une borne

<sup>2</sup>Une *antichaîne* dans un ensemble partiellement ordonné  $P$  est un sous-ensemble  $A$  de  $P$  tel que chaque paire d'éléments de  $A$  est incomparable, i.e., pour tout  $x, y$  dans  $A$ , on ne peut avoir ni  $x \leq y$ , ni  $y \leq x$  (dans notre propos, il n'y a ni de chemin de  $x$  à  $y$ , ni de chemin de  $y$  à  $x$ ).



(a)  $\mathcal{G}$



(b)  $\mathcal{N}$

FIG. 3: Pour le graphe orienté  $\mathcal{G}$  représenté en Partie (a), la Partie (b) fournit le réseau correspondant  $\mathcal{N}$ . Les arcs représentés par des tirets sont des arcs *extrémités de chemins*, les arcs en gras représentent les arcs associés aux *noeuds dédoublés*, l'arc en pointillé représente l'*arc de retour*, et les arcs plein sont les arcs associés au *dag*.

triviale inutile. Dans le but d'imposer un flot non-nul au travers de chaque noeud du graphe, nous coupons chaque noeud  $i$  en deux noeuds  $i'$  et  $i''$ , et nous ajoutons l'arc  $(i', i'')$ . Ainsi, la conservation du flot nous assure l'existence d'un flot non-nul de  $i'$  à  $i''$ .

**exemple 3.** La Figure 3 illustre les Définitions 4 et 5. Chaque noeud  $i$  de la Figure 3a est dédoublé en deux noeuds distincts  $i'$  et  $i''$  dans la Figure 3b. L'arc en pointillé de la Figure 3b coorespond à l'arc de retour de la Définition 5, avec un flot  $[1, \bar{K}]$ , les arcs en gras correspondent aux arcs associés aux noeuds dédoublés, avec un flot unitaire, les arcs représentés par des tirets correspondent aux extrémités de chemins, avec un flot  $[0, \bar{K}]$ , et les arcs plein correspondent quand à eux aux arcs associés au dag, avec un flot  $[0, \bar{K}]$ .

**Définition 4.** Etant donné un graphe orienté  $\mathcal{G}$ , le réseau  $\mathcal{N} = (\mathcal{X}, \mathcal{E}, c)$  est défini de la manière suivante :

- À chaque noeud  $i$  de  $\mathcal{G}$  correspond deux noeuds  $i'$  (appelé noeud entrée) et  $i''$  (appelé noeud sortie), ainsi qu'un arc  $(i', i'')$  associé aux noeuds dédoublés de  $\mathcal{E}$ . Soient  $\mathcal{I}'$  et  $\mathcal{I}''$  les deux ensembles de noeuds correspondant. Finalement, deux noeuds supplémentaires  $s$  (source) et  $t$  (puits) sont ajoutés dans  $\mathcal{X}$ .
- À chaque arc  $(i, j)$  de  $\mathcal{G}$  correspond un arc  $(i'', j')$  de  $\mathcal{E}$ , où  $i''$  est le noeud de sortie de  $i$  et  $j'$  le noeud d'entrée de  $j$ .

- Il existe un arc  $(s, i')$  dans  $\mathcal{E}$ , pour tout noeud  $i' \in \mathcal{T}'$ , ainsi qu'un arc  $(i'', t)$  pour tout noeud de sortie  $i'' \in \mathcal{T}''$  d'un noeud  $i$  de  $\mathcal{T}$ .
- Un arc de retour de  $t$  à  $s$  est ajouté.

Maintenant, nous sommes en mesure de définir la fonction  $f$  associée au réseau  $\mathcal{N}$ . Dans ce but, nous distinguons différentes classes d'arcs correspondant respectivement aux arcs de retour, aux arcs associés aux noeuds dédoublés, aux arcs du dag, et aux arcs extrémités de chemins.

**Définition 5.** *Étant donné un graphe acyclique  $\mathcal{G}$ , le flot porté par chaque arc du réseau  $\mathcal{N}$ , associé à  $\mathcal{G}$ , est défini de la manière suivante :*

1. L'arc de retour :  $f(t, s) = f_{ts} \in [1, \bar{K}]$ , où  $\bar{K} = |\mathcal{T}|$ .
2. Les arcs associés aux noeuds dédoublés :  $f(i', i'') = f_i = 1$ .
3. Les arcs du dag :  $f(i'', j') = f_{ij} \in [\ell_{ij}, c_{ij}]$  où  $\ell_{ij} = 0$  et  $c_{ij} = \bar{K}$ .
4. Les arcs extrémités de chemins :  $f(s, i) = f(i, t) = f_{si} = f_{it} \in [0, \bar{K}]$ , pour tout  $(s, i)$  et  $(i, t)$  de  $\mathcal{E}$ .

On peut ensuite observer que chaque cycle dans  $\mathcal{N}$  contient l'arc  $(t, s)$  car le graphe  $\mathcal{G}$  originellement acyclique. Alors, la conservation du flot assure que le flot global de  $\mathcal{N}$  est fourni par :

$$\mathcal{F}(\mathcal{N}) = \sum_{(s,j) \in \omega^+(\{s\})} f_{sj} = f_{ts} \quad (4)$$

Dans la suite, le flot global de  $\mathcal{N}$  sera assimilé au flot  $f_{ts}$  traversant l'arc de retour  $(t, s)$ .

**Théorème 2.** *Un graphe orienté acyclique  $\mathcal{G}$  peut être partitionné en  $K$  chemins noeuds-disjoints ssi il existe un flot compatible  $f_{ts} = K$  dans le réseau  $\mathcal{N}$  associé à  $\mathcal{G}$ .*

*Démonstration.* On prouve d'abord que la condition est nécessaire. Supposons que (a) il existe une partition en chemins de  $\mathcal{G}$  de taille  $K \in [1, \bar{K}]$ , et (b) qu'il n'existe pas un flot compatible dans  $\mathcal{N} = (\mathcal{X}, \mathcal{E})$  avec  $\mathcal{F}(\mathcal{N}) \in [1, \bar{K}]$ . Montrons qu'il existe une contradiction. D'après (b), il n'existe pas de flot compatible dans  $\mathcal{N}$  avec  $\mathcal{F}(\mathcal{N}) = K$  alors, il existe un ensemble de noeuds  $A \subseteq \mathcal{X}$  tel que  $s \in A$ , pour lequel le théorème de Hoffman assure :

$$U_{\omega^+(A)} = \sum_{(i,j) \in \omega^+(A)} c_{ij} < \sum_{(i,j) \in \omega^-(A)} \ell_{ij} = U_{\omega^-(A)} \quad (5)$$

Pour le cocycle  $\omega(A)$ ,  $(t, s)$  est l'unique arc atteignant  $A$  (car  $s \in A$  et  $\mathcal{N} \setminus \{(t, s)\}$  est acyclique). De plus, par définition de  $\mathcal{N}$ ,  $f_{ts} \in [1, \bar{K}]$ , donc :

$$1 \leq U_{\omega^+(A)} < U_{\omega^-(A)} \leq \bar{K} \quad (6)$$

Ceci signifie que le flot minimum atteignant  $\omega(A)$  dépasse la capacité maximum pouvant sortir de  $\omega(A)$ . La contradiction est alors évidente en se rappelant que  $\omega(A)$  est sans cycles.

Nous prouvons maintenant que la condition est suffisante. Nous montrons qu'à partir de n'importe quel flot compatible de  $\mathcal{N}$ ,  $\mathcal{F}(\mathcal{N}) = f_{ts} = K$ , on peut construire un partition de  $\mathcal{G}$  en  $K$  chemins noeuds-disjoints. Dans ce but, considérons un flot compatible  $f_{ts}$  de  $\mathcal{N}$  et un DAG  $\mathcal{N}' = (\mathcal{X}', \mathcal{E}')$  défini à partir du flot  $f_{ts}$  de la manière suivante :

- $\mathcal{X}' = \mathcal{X} \setminus \{s, t\}$ .
- $\mathcal{E}' = \{(i, j) \in \mathcal{E} \setminus (t, s) \mid f_{ij} = 1\}$ .

Le flot  $f_{ts}$  assure que  $\mathcal{N}'$  est un DAG composé de  $K$  composantes connexes telles que chacune corresponde à un chemin élémentaire. Cette propriété est directement dérivée du fait que pour chaque arc du dag  $(i, j)$  de  $\mathcal{N}$ , soit  $f_{ij} = 1$  ou  $f_{ij} = 0$  (car dans le cas des DAGs chaque arc associé aux noeuds dédoublés  $(i', i'')$  a un flot  $f_i = 1$ ). Alors, en contractant chaque arc de  $\mathcal{N}'$  associé aux noeuds dédoublés en un unique noeud,  $\mathcal{N}'$  devient un graphe partiel de  $\mathcal{G}$  induit par  $\mathcal{E}' \subseteq \mathcal{A}$ , et composé de  $K$  chemins noeuds-disjoints.  $\square$

**Corollaire 1.** *Étant donné un graphe orienté acyclique  $\mathcal{G}$  et le réseau  $\mathcal{N}$  associé. La borne inférieure du nombre de chemins noeuds-disjoints partitionnant  $\mathcal{G}$  est donnée par le flot minimum  $f_{ts}^*$  de  $\mathcal{N}$  :  $f_{ts}^* = \underline{K}$ .*

*Démonstration.* Directement dérivée du Theorem 2.  $\square$

## 4 Le problème $K$ -NDP dans le cas de graphes quelconques

Comme nous l'avons vu dans la section précédente, dans le cas où  $K$  n'est pas fixé, le point clé d'une approche résolvant le problème  $K$ -NDP repose sur l'évaluation d'une borne inférieure du nombre minimum de chemins partitionnant le graphe  $\mathcal{G}$ . En effet, considérons un graphe orienté  $\mathcal{G}$  et un ensemble  $\mathcal{T}$  de noeuds potentiellement finaux. À partir d'une partition  $\mathcal{P}$  quelconque de taille  $p \in [1, |\mathcal{T}|]$ , on peut construire un partition  $\mathcal{P}'$  de taille  $p' \in [p+1, |\mathcal{T}|]$  en décomposant, éventuellement, un de ces chemins par un noeud potentiellement final. Cependant, déterminer à partir de  $\mathcal{P}$ , une partition  $\mathcal{P}''$  de taille  $p'' \in [1, p-1]$  est un problème NP-complet (supposons que  $p = 2$  alors,  $p'' = 1$  et par conséquent, nous devons construire un chemin Hamiltonien).

Dans la Section 3, un modèle de flot a été proposé dans le cadre de graphes acycliques. Cette section montre comment étendre aux graphes quelconques ce modèle, en considérant le graphe réduit  $\mathcal{G}_r$  associé à  $\mathcal{G}$ . Un noeud du graphe réduit  $\mathcal{G}_r$  correspond à une composante fortement connexe du graphe initial  $\mathcal{G}$ . Par conséquent nous devons

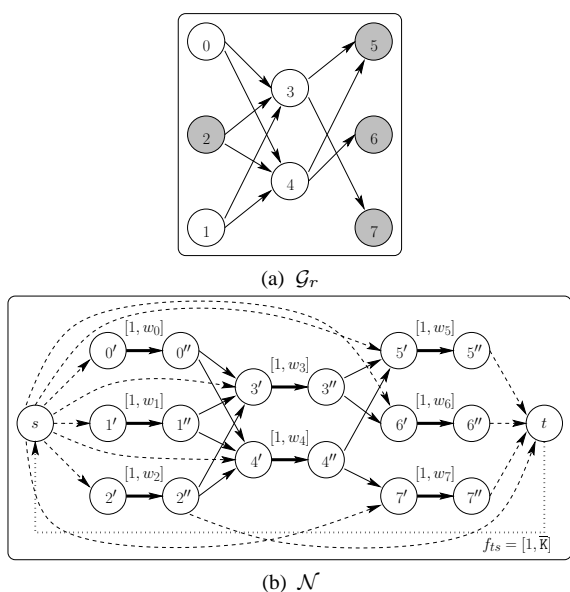


FIG. 4: Le graphe orienté  $\mathcal{G}_r$ , représenté par les parties (a) et (b) fournit le réseau  $\mathcal{N}$  correspondant. Les arcs avec des tirets représentent des arcs extrémités de chemins, les arcs en gras représentent les arcs associés aux noeuds dédoublés, l'arc en pointillé  $(t, s)$  représente l'arc de retour du réseau.

distinguer les arcs du *dag* (i.e., les arcs de  $\mathcal{G}_r$ ) et les arcs associés aux *noeuds dédoublés* (i.e., les arcs représentant les *scc*'s de  $\mathcal{G}$ ), et déterminer la borne inférieure en tenant compte des deux informations. Ainsi, les Définitions 4 et 5 sont étendues au cas des graphes non acyclique de la manière suivante (Figure 4) :

- Le réseau  $\mathcal{N} = (\mathcal{X}, \mathcal{E}, c)$  est dérivé à partir du graphe réduit  $\mathcal{G}_r$  associé à  $\mathcal{G}$ .
- Le flot lié à chaque arc associé aux noeuds dédoublés  $(i', i'')$  de  $\mathcal{N}$  est maintenant évalué de la manière suivante :  $f(i', i'') = f_i \in [\ell_i, c_i]$ , où  $\ell_i$  (resp.  $c_i$ ) correspond à une borne inférieure (resp. supérieure) sur le nombre minimum (resp. maximum) de chemins noeuds-disjoints qui partitionnent  $C_i$ . Nous avons  $f_i \in [1, w_i]$  où  $w_i$  représente le nombre de noeuds dans  $C_i$  qui sont soit des noeuds potentiellement finaux dans  $\mathcal{G}$ , ou des noeuds directement connectés, par un unique arc de  $\mathcal{G}$ , à une *scc*  $C_j$  de  $\mathcal{G}$  ( $j \neq i$ ).

Alors, la condition nécessaire et suffisante introduite par le Théorème 2 peut être directement généralisée à une condition nécessaire dans le cas de graphes quelconques :

**Théorème 3.** *S'il existe une partition en chemins de  $\mathcal{G}$  de taille  $K \in [\underline{K}, \overline{K}]$  alors, il existe un flot compatible  $f_{ts} = K$  dans le réseau  $\mathcal{N}$  correspondant au graphe  $\mathcal{G}$ .*

De la même manière, le nombre minimum de chemins introduit par le Lemme 1 est généralisé à une borne infé-

rieure sur le nombre de chemins noeuds-disjoints partitionnant le graphe  $\mathcal{G}$ .

**Corollaire 2.** *Étant donné un graphe orienté  $\mathcal{G}$  et son réseau correspondant  $\mathcal{N}$ , une borne inférieure sur le nombre de chemins noeuds-disjoints partitionnant  $\mathcal{G}$  est donnée par le plus petit flot compatible  $f_{ts}^*$  de  $\mathcal{N}$  :  $f_{ts}^* \leq \underline{K}$ .*

Dans la suite, nous proposons deux manières d'améliorer l'évaluation du plus petit flot compatible dans le réseau  $\mathcal{N}$  associé au graphe orienté  $\mathcal{G}$  : Premièrement, dans la Section 4.1, nous montrons comment estimer le nombre de chemins partitionnant chaque *scc* de  $\mathcal{G}$  dans le but d'améliorer l'évaluation du flot traversant chaque arc de  $\mathcal{N}$  associé aux noeuds dédoublés. Deuxièmement, dans la Section 4.2, nous montrons comment estimer le nombre de chemins entre deux *scc*'s dans le but de préciser l'évaluation du flot traversant les arcs du *dag*.

#### 4.1 Estimer le nombre de chemins partitionnant une *scc*

Une première voie pour améliorer la précision de la relaxation du problème K-NDP, lorsque des *scc*'s de  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$  contiennent plus d'un noeud (i.e.,  $\mathcal{G}$  n'est pas un DAG), est de raffiner les bornes sur le flot  $f_i \in [\ell_i, c_i]$  pour chaque arc associé aux noeuds dédoublés représentant la *scc*  $C_i$  de  $\mathcal{G}$ . Cette section montre comment améliorer la précision de  $\ell_i$  (i.e., du nombre minimum de chemins partitionnant  $C_i$ ) qui était à l'origine initialisée à la valeur 1 dans la Définition 5. L'idée est d'identifier un noeud  $d$  de  $C_i$ , dont le retrait augmente le nombre de *scc*'s de  $C_i \setminus \{d\}$  (i.e., le graphe correspondant aux *scc*'s de  $C_i$  auquel le noeud  $d$  a été supprimé), dans le but de ré-appliquer le Lemme 2 sur le nouveau DAG construit à partir du graphe réduit associé à  $C_i \setminus \{d\}$ . Ceci est réalisé en utilisant la relation de dominance entre les noeuds de  $C_i$ , définie de la manière suivante :

**Définition 6** ([11]). *Étant donné une graphe orienté  $G$  et deux noeuds  $i, j$  de  $G$  tel qu'il existe au moins un chemin de  $i$  à  $j$ , un noeud  $d$  est un dominant du noeud  $j$  par rapport à un noeud  $i$  ssi il n'existe aucun chemin de  $i$  à  $j$  dans  $G \setminus \{d\}$ . L'ensemble des dominants de  $j$  par rapport à  $i$  est noté par  $DOM_{(\mathcal{G}, i)}(j)$ .*

**Notations 1.** *Étant donné un graphe orienté  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , soit  $C_i$  une *scc* de  $\mathcal{G}$  :*

- $C_i = (C_i, \mathcal{A}_i)$  représente le sous-graphe de  $\mathcal{G}$  correspondant à  $C_i$ .
- $\mathcal{T}_i$  représente le sous-ensemble des noeuds de  $\mathcal{V}_i$  qui sont soit des noeuds potentiellement finaux de  $\mathcal{G}$ , soit des noeuds atteignant, en un seul arc, une autre *scc*  $C_j$  de  $\mathcal{G}$ .

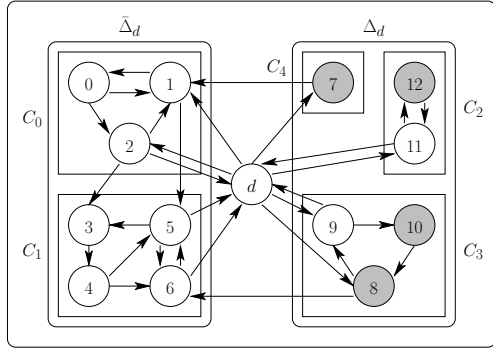


FIG. 5: Cette Figure représente une  $scc$   $C_i$  de  $\mathcal{G}$ , un noeud dominant  $d$  de  $D_i = \{1, 4, 8, 9, 11, d\}$ . Les noeuds en gris sont représentés par l'ensemble  $\mathcal{T}_i$ . Les rectangles contenus dans  $\Delta_d$  et  $\bar{\Delta}_d$  représentent les  $scc$ 's créées par le retrait de  $d$  dans  $C_i$ .

- Soit  $D_i$  l'ensemble des noeuds dominants de  $C_i$  définis grâce à l'ensemble  $\mathcal{T}_i$  :

$$D_i = \{d \mid \exists v \in \mathcal{V}_i, \forall w \in \mathcal{T}_i, d \in \text{DOM}_{(C_i, v)}(w)\} \quad (7)$$

Ainsi, le retrait du noeud dominant  $d \in D_i$  créé deux sortes de  $scc$ 's (la Figure 5 illustre ce point) :

- $\Delta_d$  est l'ensemble, possiblement vide, des nouvelles  $scc$ 's à partir desquelles un noeud de  $\mathcal{T}_i$  peut atteindre par au moins un chemin qui ne contient pas  $d$ . Soit  $\Delta_d^*$  l'ensemble des noeuds contenus dans les  $scc$ 's de  $\Delta_d$ .
- $\bar{\Delta}_d$  est l'ensemble, possiblement vide, des nouvelles  $scc$ 's à partir desquelles aucun noeud de  $\mathcal{T}_i$  ne peut être atteint par un chemin ne contenant pas  $d$ .

**exemple 4.** Dans la  $scc$   $C_i$  de  $\mathcal{G}$  représentée par la Figure 5, nous avons  $\mathcal{T}_i = \{7, 8, 10, 12\}$ . L'ensemble des noeuds dominants de  $C_i$  par rapport à  $\mathcal{T}_i$  est  $D_i = \{1, 4, 8, 9, 11, d\}$ . Par exemple, tout chemin d'un noeud 3 à un noeud de  $\mathcal{T}_i$  rencontre les noeuds 4 et  $d$ . Ainsi, les noeuds 4 et  $d$  sont des dominants du noeud 3 par rapport aux noeuds de  $\mathcal{T}_i$ . La Figure 5 illustre aussi le partitionnement de la  $scc$   $C_i$  dans les composantes fortement connexes, produites par le retrait du noeud  $d$  :

- $\Delta_d = \{C_2^d, C_3^d, C_4^d\}$  and  $\Delta_d^* = \{7, 8, 9, 10, 11, 12\}$ ,
- $\bar{\Delta}_d = \{C_0^d, C_1^d\}$  and  $\bar{\Delta}_d^* = \{0, 1, 2, 3, 4, 5, 6\}$ .

Nous introduisons maintenant une proposition qui va nous permettre de réduire le problème d'estimation du nombre minimum de chemins partitionnant  $C_i$  au problème de trouver le nombre minimum de chemins partitionnant  $\Delta_d$ .

**Proposition 1.** *S'il existe une partition en chemins de  $\mathcal{G}$  alors, pour chaque noeud dominant  $d \in D_i$  d'une  $scc$   $C_i$ , il existe un chemin Hamiltonien, terminant sur un prédecesseur de  $d$ , dans le sous-graphe de  $\mathcal{G}$  induit par l'ensemble des noeuds de  $\bar{\Delta}_d^*$ .*

*Démonstration.* Rappelons-nous que, par construction de  $\Delta_d^*$ , un chemin ne peut pas finir sur un noeud de  $\bar{\Delta}_d^*$ . Puisque  $d$  domine tout noeud de  $\mathcal{T}_i \subset \Delta_d^*$  par rapport à tout noeud de  $\bar{\Delta}_d^*$ ,  $d$  est la seule sortie possible pour les noeuds de  $\bar{\Delta}_d^*$ . Par conséquent, s'il n'existe pas un chemin Hamiltonien dans  $\bar{\Delta}_d^*$  alors, le noeud  $d$  est atteint par au moins deux chemins noeuds-disjoints couvrant  $\bar{\Delta}_d^*$  : c'est une contradiction.  $\square$

**Proposition 2.** *Considérons le graphe réduit  $\mathcal{G}_{\Delta_d}$  (resp.  $\mathcal{G}_{\bar{\Delta}_d}$ ) associé au sous-graphe d'une  $scc$   $C_i$  de  $\mathcal{G}$  induit par les noeuds de  $\Delta_d^*$  (resp.  $\bar{\Delta}_d^*$ ). Soit  $x_{\Delta_d}$  une borne inférieure du nombre minimum de chemins noeuds-disjoints partitionnant  $\mathcal{G}_{\Delta_d}$ . Une borne inférieure du nombre minimum de chemins noeuds-disjoints partitionnant  $C_i$  (noté  $l_i^d$ ), par rapport à un noeud dominant  $d$  de  $C_i$ , est définie par :*

- $x_{\Delta_d} - 1 \leq l_i^d$ , s'il existe un arc  $(u, v)$  dans  $C_i$  tel que  $u \in \Delta_d^*$ ,  $v \in \bar{\Delta}_d^*$ , et la  $scc$  contenant le noeud  $v$  est un noeud source dans  $\mathcal{G}_{\bar{\Delta}_d}$ .
- $x_{\Delta_d} \leq l_i^d$ , sinon.

La borne inférieure  $x_{\Delta_d}$  est définie par le Lemme 2 basé sur la relaxation du flot traversant le réseau associé au graphe  $\mathcal{G}_{\Delta_d}$  par rapport à l'ensemble des noeuds potentiellement finaux  $\mathcal{T}_i$ . Finalement, la valeur maximale dans l'ensemble des  $l_i^d$ 's ( $d \in D_i$ ) fournit une borne inférieure du nombre minimum de chemins dans la partition couvrant  $C_i$ .

**Proposition 3.** *Soit  $l_i$  le nombre minimum de chemins noeuds-disjoints partitionnant la  $scc$   $C_i$  de  $\mathcal{G}$  et soit  $D_i$  l'ensemble des noeuds dominants de  $C_i$ , une borne inférieure de  $l_i$  est donnée par :*

$$1 \leq \max(\{l_i^d \mid d \in D_i\}) \leq l_i \quad (8)$$

## 4.2 Estimer le nombre de chemins entre deux $scc$ 's

Une seconde manière de préciser la relaxation du problème K-NDP est d'ajuster les bornes sur le flot  $f_{ij} \in [l_{ij}, c_{ij}]$ , pour chaque arc du dag. Cette section montre comment améliorer la précision de la borne supérieure  $c_{ij}$ , qui était originellement fixée à  $\bar{\kappa} = |\mathcal{T}|$  dans la Définition 5 (i.e., le nombre de noeuds potentiellement finaux dans  $\mathcal{G}$ ). Ceci est réalisé en calculant la taille d'un couplage de cardinalité maximum entre les noeuds  $i$  incidents à un arc sortant d'une  $scc$   $C_i$ , et entrant sur un noeud  $j$  d'une  $scc$   $C_j$ .

**Notations 2.** *Étant donné deux  $scc$ 's distinctes  $C_i$  et  $C_j$  de  $\mathcal{G}$  et un noeud  $i$  de  $C_i$ ,*

- $\varphi_{ij}^+$  représente le nombre d'arcs sortant de  $i$  et entrant sur un noeud  $j$  de  $C_j$ .
- $\varphi_{ij}^-$  représente le nombre d'arcs entrant sur un noeud  $j$  de  $C_j$  et sortant du noeud  $i$ .

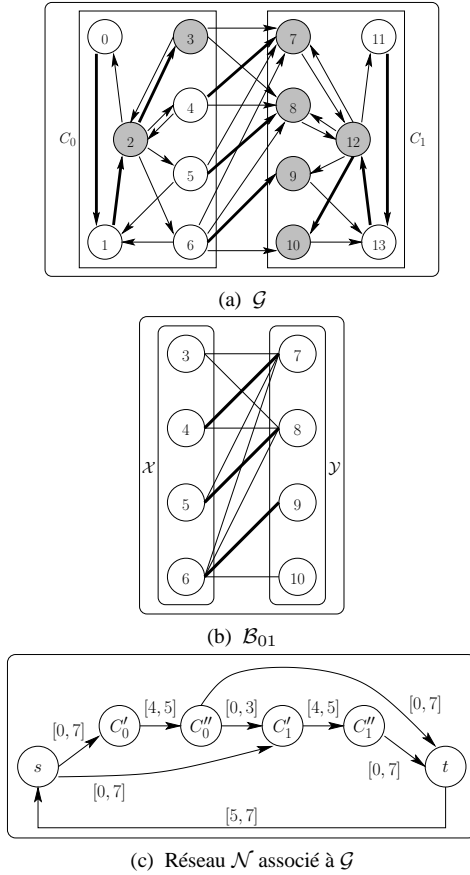


FIG. 6: La Partie (a) représente le graphe orienté  $\mathcal{G}$  composé par deux *scc*'s  $C_0$  et  $C_1$ . Les arcs en gras représentent une solution compatible maximisant le nombre de chemins noeuds-disjoints partitionnant  $\mathcal{G}$  (5 chemins). Les noeuds en gris représentent les noeuds potentiellement finaux de  $\mathcal{G}$ . La Partie (b) illustre le graphe bipartite  $\mathcal{B}_{01}$  extrait des *scc*'s  $C_0$  et  $C_1$ . La Partie (c) représente le réseau  $\mathcal{N}$  associé à  $\mathcal{G}$ .

Étant donné deux *scc*'s distinctes  $C_i$  et  $C_j$  de  $\mathcal{G}$  telles qu'il existe au moins un arc d'un noeud de  $C_i$  à un noeud de  $C_j$ , le nombre maximum de chemins sortant de  $C_i$  et entrant directement sur  $C_j$  peut être calculé à partir d'un graphe bipartite, extrait depuis  $C_i$  et  $C_j$ , défini de la manière suivante :

**Définition 7.** Le graphe bipartite  $\mathcal{B}_{ij} = (\mathcal{X}, \mathcal{Y}, \mathcal{E})$  associé avec une paire de *scc*'s distinctes  $C_i$  et  $C_j$  de  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , tel qu'il existe un arc depuis un noeud de  $C_i$  à un noeud de  $C_j$ , est définie par :

- $\mathcal{X} = \{i \in C_i \mid \varphi_{ij}^+ > 0\}$ ,
- $\mathcal{Y} = \{j \in C_j \mid \varphi_{ij}^- > 0\}$ ,
- $\mathcal{E} = \{(i, j) \in \mathcal{A} \mid i \in C_i, j \in C_j\}$ .

**Proposition 4.** Soit  $\mathcal{M}_{ij}$  la taille d'un couplage de cardinalité maximum dans le graphe bipartite  $\mathcal{B}_{ij}$  associé avec une paire de *scc*'s distinctes,  $C_i$  et  $C_j$ , de  $\mathcal{G}$ . La capacité

$c_{ij}$ , traversant un arc du dag  $(i'', j')$  du réseau  $\mathcal{N}$  associé à  $\mathcal{G}$ , est bornée par  $\mathcal{M}_{ij} : c_{ij} \leq \mathcal{M}_{ij}$ .

**exemple 5.** À partir du graphe  $\mathcal{G}$  représenté par la Figure 6a, nous construisons le graphe bipartite  $\mathcal{B}_{01}$  représenté par la Figure 6b. Un couplage de cardinalité maximum de  $\mathcal{B}_{01}$ , de taille  $\mathcal{M}_{01} = 3$ , est dépeint par les arcs en gras de la Figure 6b. Le nombre minimum de chemins noeuds-disjoints partitionnant  $\mathcal{G}$  est égal à 5 (les arcs en gras de la Figure 6a dépeignent les chemins suivants :  $\{0, 1, 2, 3\}$ ,  $\{4, 7\}$ ,  $\{5, 8\}$ ,  $\{6, 9\}$  et  $\{11, 13, 12, 10\}$ ). La Figure 6c montre le réseau résultant  $\mathcal{N}$  associé à  $\mathcal{G}$ . Notons que pour les arcs associés aux noeuds dédoublés ( $C'_0, C''_0$ ), la borne inférieure du nombre minimum de chemins couvrant la *scc*  $C_0$  est raffinée à la valeur 4 grâce à la détection du noeud dominant 2 (voir Section 4.1). De manière similaire, la borne inférieure sur le nombre de chemins couvrant la *scc*  $C_1$  est raffinée à la valeur 4 grâce au noeud dominant 12. Dans le même ordre d'idée, la borne supérieure sur le nombre maximum de chemins disjoints pouvant sortir de la *scc*  $C_0$ , et atteignant la *scc*  $C_1$  (représenté par la capacité de l'arc associés aux noeuds dédoublés ( $C'_0, C'_1$ )), est raffinée à la valeur  $\mathcal{M}_{01} = 3$ .

Ainsi, la valeur d'un flot minimum de  $\mathcal{N}$  est égale à 5. En effet, supposons que la valeur soit 4 (i.e.,  $f_{ts} = 4$ ) alors, 4 unités de flot peuvent être injectées sur l'arc  $(s, C'_0)$  car on sait que  $\ell_0 = 4$ , et que au plus 3 unités doivent atteindre  $C'_1$  : il y a donc une contradiction avec  $\ell_1 = 4$ . Ainsi, il n'existe pas de flot compatible avec  $f_{ts} = 4$ .

## 5 Une contrainte de partitionnement par des chemins

Une contrainte de partitionnement par des chemins peut être définie par un graphe orienté  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ , un nombre de chemins variable NPATH, et un ensemble  $\mathcal{T} \subseteq \mathcal{V}$  de noeuds potentiellement finaux de  $\mathcal{G}$ . Sans perte de généralité, la description de la contrainte de chemin proposée suit la description faite de la contrainte d'arbre "tree" introduite dans [4]. Ceci nous permet d'envisager d'utiliser tout le filtrage développé pour tree afin de prévenir la création de cycles, etc... En plus, il est alors possible d'utiliser la restriction sur le degré de chaque noeud du graphe proposée dans [5] pour forcer que chacun possède exactement un prédécesseur.

Ainsi, la contrainte de chemin peut être formalisée sous la forme suivante path(NPATH, NODES), où NPATH est une variable domaine<sup>3</sup> spécifiant le nombre de chemins autorisés dans la partition, et NODES est la collection de  $n$  noeuds NODES[1], ..., NODES[n] du graphe  $\mathcal{G}$ . Chaque noeud  $v_i =$

<sup>3</sup>Une variable domaine  $V$  est une variable définie sur un ensemble d'entiers noté  $dom(V)$ .  $min(V)$  et  $max(V)$  représentent respectivement la valeur minimum et maximum de  $V$ .



NODES[ $i$ ] possède les attributs suivants, complétant la description de  $\mathcal{G}$  :

- $I$  est un entier défini sur  $[1, n]$ . Il peut être interprété comme l’*identifiant* du noeud  $v_i$ .
- $S$  est une variable domaine (ou variable *successeur*) dont le domaine est constitué des éléments (l’identifiant des noeuds du graphe) de  $[1, n]$ . Il peut être interprété comme l’unique successeur du noeud  $v_i$ . De plus, si  $i \in \text{dom}(\text{NODES}[i].S)$  alors, nous dirons que  $v_i$  est un noeud *potentiellement final* pour un chemin de la partition.

Pour chaque  $i \in [1, n]$ , les termes NODES[ $i$ ]. $I$  et NODES[ $i$ ]. $S$  représentent respectivement les attributs  $I$  et  $S$  de NODES[ $i$ ]. De plus, le graphe  $\mathcal{G}$  associé à la contrainte de chemin peut être formellement défini de la manière suivante :

**Définition 8.** *Le graphe orienté  $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ ,  $|\mathcal{V}| = n$  et  $|\mathcal{A}| = m$ , ainsi que l’ensemble  $\mathcal{T}$  associés à une contrainte path(NPATH, NODES) sont définis comme suit :*

- $\mathcal{V} = \{i \mid i \in [1, n]\}$ ,
- $\mathcal{A} = \{(i, j) \mid j \in \text{dom}(\text{NODES}[i].S), i \neq j\}$ ,
- $\mathcal{T} = \{i \mid i \in \text{dom}(\text{NODES}[i].S)\}$ .

Finalement, pour compléter la description de la contrainte, nous introduisons la description formelle d’une instantiation complète et valide des variables. Pour des raisons de lisibilité, nous allons directement raisonner sur le graphe  $\mathcal{G}$  qui modélise la contrainte de chemin.

**Définition 9.** *Une instantiation complète des variables associées à une contrainte path(NPATH, NODES) est valide ssi :*

- $\forall i \in [1, n] : \text{NODES}[i].I = i$ .
- $\mathcal{G}$  est composé de NPATH composantes connexes telles que chacune soit un chemin élémentaire terminant sur un noeud potentiellement final de  $\mathcal{T}$ .

Le reste de la section est organisée de la manière suivante : la sous-section 5.1 fournit deux conditions nécessaires au partitionnement du graphe  $\mathcal{G}$  associé à la contrainte de chemin, dérivées de la condition nécessaire introduite par le Théorème 3. La sous-section 5.2 montre comment exploiter cette condition nécessaire dans le but de filtrer NPATH ainsi que les variables successeurs modélisant le graphe  $\mathcal{G}$ .

## 5.1 Faisabilité

Basée sur le Théorème 3 de la Section 4, nous introduisons un algorithme vérifiant la faisabilité de la contrainte de chemin vis-à-vis d’un nombre maximum de chemins autorisés  $\max(\text{NPATH})$  :

- Vérifier l’existence d’un flot compatible dans le réseau  $\mathcal{N}$  associé à  $\mathcal{G}$ .

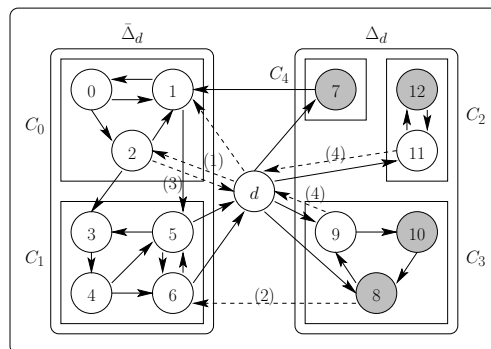


FIG. 7: À partir de la Figure 5 de la Section 4.1, les arcs en pointillés représentent les arcs impossibles de  $\mathcal{G}$  détectés par l’algorithme introduit en Section 5.2.

- Pour chaque scc  $C_i$  de  $\mathcal{G}$ , pour chaque noeud dominant  $d$  de  $C_i$ , vérifie qu’il existe au plus un chemin couvrant le graphe  $\mathcal{G}_{\Delta_d}$  (dérivé de la Proposition 1).<sup>4</sup>

## 5.2 Filtrage

Cette section montre comment filtrer les domaines des variables successeurs NODES[1]. $S, \dots, \text{NODES}[n].S$ , ainsi que celui de la variable NPATH à partir du graphe  $\mathcal{G}$  associé à la contrainte de chemin :

- Ajuster le minimum de NPATH au flot minimum compatible dans le réseau  $\mathcal{N}$  associé à  $\mathcal{G}$ .
- Pour chaque scc  $C_i$  de  $\mathcal{G}$ , pour chaque noeud dominant  $d \in C_i$ , supprimer chaque arc  $(i, j)$  tel que :
  - Cas (1) :  $i = d$  et  $j \in \Delta_d$ .
  - Cas (2) :  $i \in \Delta_d$  et  $j \in \Delta_d$  tels que la scc, créée par le retrait de  $d$  et contenant  $j$ , n’est pas un noeud source dans le graphe  $\mathcal{G}_{\Delta_d}$ .
  - Cas (3) :  $j = d$  et  $i \in \Delta_d$  tels que la scc, créée par le retrait de  $d$  et contenant  $i$ , n’est pas un noeud puit dans le graphe  $\mathcal{G}_{\Delta_d}$ .
  - Cas (4) :  $i \in \Delta_d, j = d$ ,  $\Delta_d \neq \emptyset$ .

**exemple 6.** *La Figure 7 illustre les arcs impossibles détectés par l’algorithme précédent. Les arcs  $(d, 1)$  et  $(d, 2)$  sont détectés par le Cas (1), l’arc  $(8, 6)$  l’est par le Cas (2), l’arc  $(2, d)$  par le Cas (3) et, les arcs  $(9, d)$  et  $(11, d)$  sont finalement détectés par le Cas (4).*

Il faut observer que l’on aurait aussi pu supprimer tout arc  $(i, j)$  de  $\mathcal{G}$ , tel que  $i$  et  $j$  appartiennent à des scc’s distinctes  $C_i$  et  $C_j$ , si l’arc de  $\mathcal{N}$  associé aux noeuds dédoublés, correspondant à l’arc  $(i, j)$ , ne peut porter aucun flot (dans tout flot compatible de  $\mathcal{N}$ ). Ce filtrage n’a pas été inclus dans l’algorithme précédent puisqu’à l’heure actuelle, nous ne connaissons pas de méthode “efficace” captant le

<sup>4</sup>Rappelons-nous que  $\mathcal{G}_{\Delta_d}$  est le graphe réduit associé au sous-graphe de  $\mathcal{G}$  induit par les noeuds de  $\Delta_d$ .

fait qu'un arc d'un réseau ne peut porter aucun flot dans tout flot compatible de ce réseau.

## 6 Conclusion

Les algorithmes de filtrage proposés dans les contraintes de chemins précédentes étaient uniquement basés sur la prévention de circuits, ainsi que sur des contraintes maintenant le degré de chaque noeud (chaque noeud ayant au plus un prédécesseur/successeur). Cet article considère le problème de partitionnement par des chemins pour lequel le nombre de chemins n'est pas fixé mais limité à une valeur donnée. Dans ce contexte, nous proposons un modèle de flot combinant la structure du graphe réduit associé à  $\mathcal{G}$ , la structure de chaque composante fortement connexe de  $\mathcal{G}$  vis-à-vis de la relation de dominance entre les noeuds, et la manière dont les noeuds entre paires de composantes fortement connexes sont connectés.

Cependant, plusieurs questions restent ouvertes. Premièrement, dans notre approche, chaque noeud dominant est considéré de manière indépendante des autres, et on peut supposer qu'il existe une interaction entre les noeuds dominants, de laquelle il serait peut être possible de déduire une amélioration des bornes sur chaque arc associé aux noeuds dédoublés du réseau, ainsi qu'un filtrage plus précis. Deuxièmement, dans la Section 5.2, nous avons suggéré une filtrage lié à la détection des arcs qui n'appartiennent à aucun flot compatible du réseau  $\mathcal{N}$ . À notre connaissance, ce problème, bien que polynomial, ne connaît pas de réponse efficace (autre qu'un algorithme basé sur un test pour chaque arc du réseau).

## Références

- [1] S. Rao Arikati and C. Pandu Rangan. Linear algorithm for optimal path cover problem on interval graphs. *Inf. Process. Lett.*, 35(3) :149–153, 1990.
- [2] M.L. Balinski and R.E. Quandt. On an Integer Program for Delivery Problem. *Operations Research*, 12(2) :300–304, 1964.
- [3] N. Beldiceanu and E. Contejean. Introducing global constraint in CHIP. *Mathl. Comput. Modelling*, 20(12) :97–123, 1994.
- [4] N. Beldiceanu, P. Flener, and X. Lorca. The tree constraint. In *CP-AI-OR'05*, volume 3524 of *LNCS*, pages 64–78. Springer-Verlag, 2005.
- [5] Nicolas Beldiceanu, Pierre Flener, and Xavier Lorca. Combining tree partitioning, precedence, incomparability, and degree constraints, with an application to phylogenetic and ordered-path problems. Technical Report 2006-020, Department of Information Technology, Uppsala University, Sweden, April 2006.
- [6] C. Berge. *Graphes et Hypergraphes*. Dunod, Paris, 1970. In French.
- [7] H. Cambazard and E. Bourreau. Conception d'une contrainte globale de chemin. In *JNPC'04*, pages 107–120, 2004. In French.
- [8] R. P. Dilworth. A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51 :161–166, 1950.
- [9] Steven Fortune, John E. Hopcroft, and James Wylie. The directed subgraph homeomorphism problem. *Theor. Comput. Sci.*, 10 :111–121, 1980.
- [10] Latife Genç Kaya and John N. Hooker. A filter for the circuit constraint. In *CP'06*, volume 4204, pages 706–710. Springer-Verlag, 2006.
- [11] T. Lengauer and R.E. Tarjan. A fast algorithm for finding dominators in a flowgraph. *ACM Trans Program. Lang. Syst.*, 1(1) :121–141, 1979.
- [12] P. Prosser and C. Unsworth. Rooted tree and spanning tree constraints. Technical Report cppo-13-2006, CP Pod research group, May 2006.
- [13] Luis Quesada, Peter Van Roy, Yves Deville, and Raphaël Collet. Using dominators for solving constrained path problems. In *PADL'06*, volume 3819 of *LNCS*, pages 73–87, 2006.
- [14] A. Schrijver. *Combinatorial Optimization*. Springer, Berlin, 2003.
- [15] M. Sellmann. Cost-based filtering for shortest path constraints. In *CP'03*, volume 2833 of *LNCS*, pages 694–708. Springer-Verlag, 2003.
- [16] George Steiner. On the k-path partition of graphs. *Theor. Comput. Sci.*, 290(3) :2147–2155, 2003.
- [17] J.W. Suurballe. Disjoint Paths in a Network. *Networks*, 4 :125–145, 1974.
- [18] Jens Vygen. NP-completeness of some edge-disjoint paths problems. *Discrete Appl. Math.*, 61(1) :83–90, 1995.
- [19] Jing-Ho Yan and Gerard J. Chang. The path-partition problem in block graphs. *Inf. Process. Lett.*, 52(6) :317–322, 1994.