

## Chapitre 1

# Une introduction aux problèmes combinatoires inverses

### 1.1. Introduction

Dans le cadre usuel de l'optimisation, une instance d'un problème consiste à déterminer une solution optimale à partir des données définissant l'instance. Dans de nombreux cas pratiques, le problème à résoudre résulte d'un modèle dont les paramètres réels ne sont pas toujours connus avec précision. Bien connaître ces paramètres devient alors un enjeu important pour utiliser le modèle dans des études prédictives. Si le phénomène étudié est sujet aux plans d'expérience et si les variables du problème sont des quantités mesurables, on obtient de manière expérimentale une solution (observée) du problème qui peut nous renseigner sur ses paramètres. L'optimisation inverse consiste justement à *inférer les paramètres à partir d'une solution*. Cette problématique [TARANTOLA 87] est apparue à l'origine pour des problèmes en variables continues, par exemple en géophysique : les modèles de propagation des ondes définissent la forme générale du problème à résoudre. Les paramètres du modèle correspondent à des caractéristiques du sous-sol que l'on ne peut pas (ou difficilement) mesurer directement. A partir d'expériences d'explosions, en interprétant la réaction du sol observée comme le résultat du modèle, l'étude du problème inverse associé permet de remonter au système de paramètres.

Dans le cadre de l'optimisation combinatoire, des problèmes inverses ont été envisagés depuis un peu plus de dix ans (voir par exemple [BURTON 92]) et donnent lieu à de nombreuses études depuis la fin des années 1990. Etant donné un problème

## 2 Optimisation Combinatoire

d'optimisation combinatoire ou linéaire et une solution réalisable, déterminer un système de paramètres (notamment des coefficients de l'objectif) rendant cette solution optimale ne pose en général pas de problème. En particulier, il suffit souvent de rendre l'objectif nul pour garantir l'optimalité de la solution fixée. Aussi, la problématique qui s'impose le plus souvent dans ce cadre est de déterminer, à partir d'une solution fixée et d'un système de paramètres initial, de modifier au minimum les paramètres, au sens d'une norme choisie, afin de rendre la solution fixée optimale. En général, les paramètres à modifier sont les coefficients de l'objectif. Dans ce cas, tout vecteur objectif rendant la solution fixée optimale est solution réalisable du problème inverse ; la déviation (distance) par rapport au vecteur initial mesure alors la valeur de cette solution réalisable.

Du point de vue des applications, cette problématique apparaît, comme dans le cas précédent, pour paramétrer des modèles. Dans certains cas, le vecteur initial peut s'interpréter comme une première estimation imprécise ou incomplète des paramètres à estimer et permet d'éviter des solutions triviales non pertinentes. Dans le cadre géophysique, un problème de cheminement inverse a par exemple été introduit dans le cadre de la prédiction des mouvements des tremblements de terre, voir [MOSER 91]. Les zones géologiques sont quadrillées en un certain nombre de cellules représentées par des sommets. Les relations d'adjacence sont modélisées par des arcs dans le réseau correspondant et les coûts de ces arcs représentent les temps de propagation. Bien que certains temps de transmission soient connus, les valeurs précises sont difficiles à obtenir. En observant un tremblement de terre et le temps d'arrivée des perturbations sismiques à différents capteurs et en supposant que les ondes empruntent les chemins les plus courts, le problème consistant à affiner les estimations des temps de transmission entre les cellules correspond à un problème de plus court chemin inverse.

Mentionnons aussi l'exemple de la localisation d'un centre que nous étudions dans la section 1.4.2. Sous sa forme usuelle, ce problème consiste à trouver une localisation d'un centre opérationnel dans un réseau qui minimise sa plus grande distance aux autres sommets du réseau (donc le temps maximum d'intervention). Par exemple, ce problème se pose typiquement pour placer « idéalement » un hôpital ou une caserne de pompiers. Le problème inverse associé consiste à modifier aussi peu que possible les distances du réseau (voire sa structure) de telle sorte qu'un lieu donné à l'avance devienne centre optimal. Imaginons que l'hôpital d'une ville ne soit plus centre optimal du réseau, à force de modifications successives du tissu urbain ; dans le cadre d'un plan de rénovation de la ville, une alternative à la construction d'un nouvel hôpital est de trouver l'investissement minimal permettant à l'hôpital existant de redevenir centre optimal.

Une autre classe d'applications potentielles concerne des problèmes de tarification ou d'incitation : comment modifier les coûts ou la structure d'un réseau de sorte que les flux de voyageurs suivent une solution idéale (du point de vue de la sécurité, de l'environnement, etc.) ou au moins s'en rapprochent ? Mentionnons enfin la gestion

des transitions : un système régi par un problème d'optimisation exploite depuis longtemps une solution. Si des modifications de cette solution sont commandées par des décisions exogènes au système, il s'agit de déterminer les modifications minimales du système permettant de rendre la nouvelle solution optimale.

Une autre question naturelle consiste à imposer à la nouvelle solution, non pas d'être optimale, mais au moins de ne pas être plus chère dans la nouvelle configuration que la précédente ne l'était dans l'ancienne. On définit alors une variante des problèmes inverses pour laquelle on choisit la valeur optimale et non une solution. Plus généralement, ces différents champs d'applications inspirent de nombreuses généralisations de la problématique inverse. Dans certains cas, les paramètres à modifier portent sur les contraintes et non sur l'objectif ; nous en verrons un exemple à propos d'un problème de flot maximum. Dans de nombreuses situations, il s'avère utile d'imposer des contraintes limitant les possibilités de modification des paramètres ; nous incluons systématiquement cette possibilité dans le formalisme défini à la fin de cette introduction. En particulier, nous envisageons dans ce chapitre des problèmes inverses pour lesquels les paramètres ne peuvent prendre que des valeurs entières, voire la valeur 0 ou 1. A notre connaissance, cette question n'a jamais été envisagée. Or, elle semble intéressante, notamment pour modéliser des situations où il ne s'agit pas de définir de nouveaux paramètres du système mais une nouvelle structure. Enfin, en conclusion, nous évoquons d'autres généralisations ouvrant de larges champs à explorer.

La problématique inverse en optimisation combinatoire est relativement jeune et à été surtout étudiée, jusqu'à présent, pour des problèmes spécifiques. La première question qui se pose, pour l'étude d'un nouveau problème sous ce point de vue, est de caractériser sa difficulté. C'est pourquoi nous avons choisi de structurer ce chapitre en deux parties : dans un premier temps, nous évoquons des problèmes inverses polynomiaux et discutons les méthodes de résolution associées et dans un second temps, nous discutons différents problèmes inverses difficiles. Cette présentation permet aussi d'aborder la question générale de comparer, du point de vue de la complexité algorithmique, le problème initial et ses versions inverses.

L'objectif de ce chapitre n'est pas de référer tous les résultats actuellement connus, mais plus de montrer un panel assez large de résultats. Nous avons sélectionné des résultats relativement simples et qui illustrent les différents types de problèmes, de résultats et de techniques connus. Nous insistons aussi sur de nouveaux champs d'études qui nous semblent porteurs.

## 1.2. Définitions et notations

Soit un problème d'optimisation  $\pi_C$  de  $\mathbb{R}^m$  dont chaque instance  $I = (c, \mathcal{D}, f)$  où  $c \in \mathcal{C} \subseteq \mathbb{R}^\ell$ ,  $\mathcal{D} \subseteq \mathbb{R}^m$  et  $f : \mathbb{R}^\ell \times \mathbb{R}^m \rightarrow \mathbb{R}$ , consiste à résoudre  $\min\{f(c, x) \mid x \in \mathcal{D}\}$ .

#### 4 Optimisation Combinatoire

$x \in \mathcal{D}$ . Le vecteur  $x$  représente les *variables* de  $\pi_{\mathcal{C}}$  et l'ensemble  $\mathcal{D}$  décrit la région des *solutions réalisables*. Finalement, la fonction  $f$  associe une valeur à chaque solution et chaque paramètre ; elle est communément appelée *fonction objective*. Sous des hypothèses générales, il existe une solution optimale  $x^*$  de  $\pi_{\mathcal{C}}$  pour chaque instance ; c'est notamment le cas lorsque  $\mathcal{D}$  est compact de  $\mathbb{R}^m$  et que  $f$  est une fonction continue. Souvent, dans le cadre de l'*optimisation combinatoire*, l'ensemble  $\mathcal{D}$  est un ensemble fini pour chaque instance  $I$  et une solution optimale existe pour toute fonction objective.

Le vecteur  $c$  modélise les *paramètres* de l'instance qui seront les *variables* du problème inverse. Souvent,  $\ell = m$  et  $c$  correspond aux coefficients de l'objectif. L'ensemble  $\mathcal{C}$  exprime les *contraintes* sur ces paramètres. Nous désignons par  $\|\cdot\|_p$  la norme  $L_p$  : pour deux vecteurs  $x = (x_1, \dots, x_m)$  et  $y = (y_1, \dots, y_m)$ , leur distance sous cette norme est définie par :  $\|x - y\|_p = (\sum_{i=1}^m |x_i - y_i|^p)^{\frac{1}{p}}$  pour  $p \in \mathbb{N}$ ,  $\|x - y\|_{\infty} = \max_{i=1, \dots, m} |x_i - y_i|$  pour  $p = \infty$ .

Etant donné  $x^* \in \mathcal{D}$ , le cadre inverse consiste à savoir s'il existe un vecteur de paramètres  $c^* \in \mathcal{C}$  tel que  $x^*$  est solution optimale de  $\pi_{\mathcal{C}}$ . Sous des hypothèses générales, nous verrons qu'un tel vecteur  $c^*$  existe pour chaque  $x^* \in \mathcal{D}$ . On s'intéresse alors à trouver les meilleurs vecteurs  $c^*$  possibles, c'est-à-dire ceux qui diffèrent le moins possible de  $c$  ; c'est précisément la problématique des *problèmes d'optimisation combinatoire inverses*. Formellement, le problème inverse associé à  $\pi_{\mathcal{C}}$ , noté  $\text{INV}\pi_{\mathcal{C}}$ , est défini relativement à la norme  $L_p$  par : étant données une instance  $I = (c, \mathcal{D}, f)$  de  $\pi_{\mathcal{C}}$  et une solution  $x^* \in \mathcal{D}$ , déterminer un vecteur de paramètres  $c^* \in \mathcal{C}$  minimisant la quantité  $\|c^* - c\|_p$  et tel que  $x^*$  soit une solution optimale de  $\pi_{\mathcal{C}}$  pour l'instance  $I = (c^*, \mathcal{D}, f)$  (c'est-à-dire,  $f(c^*, x^*) = \min\{f(c^*, x) \mid x \in \mathcal{D}\}$ ).

Le choix de la norme utilisée dépendra souvent du contexte de l'application sous-jacente. Par exemple, pour le problème décrit en introduction, [BURTON 92] modélisent et discutent du problème du plus court chemin inverse sous la norme  $L_2$  qui est utilisée en physique dans des problèmes de dispersion d'énergie. D'autres auteurs s'intéressent pourtant à ce problème sous d'autres normes, citons par exemple [AHUJA 01] pour la norme  $L_1$  ou [ZHANG 99] pour la norme  $L_{\infty}$ . Afin de rester simple et concis, nous présenterons uniquement dans ce document des résultats sous la norme  $L_1$ . Aussi, sans autre précision, il faudra comprendre que l'étude concerne un problème d'optimisation combinatoire inverse sous la norme  $L_1$ .

Selon une terminologie usuelle, le terme *optimisation combinatoire* recouvre ici principalement des problèmes d'optimisation discrets (voire en général discrets finis) ainsi que des problèmes linéaires pour lesquels l'ensemble des solutions de base constitue une structure discrète sous-jacente. La problématique décrite ici pourrait largement dépasser ce cadre ; toutefois, pour rester cohérent avec la terminologie usuelle, le terme *optimisation combinatoire inverse* fait systématiquement référence au cadre que nous venons de décrire (problème  $\text{INV}\pi_{\mathcal{C}}$ ) qui se distingue d'autres problèmes

inverses couramment étudiés en analyse numérique. Dans ce chapitre, pour simplifier le vocabulaire lorsqu'il n'y aura aucune ambiguïté possible, le terme *problème inverse* fera systématiquement référence à l'optimisation combinatoire inverse.

Enfin, la dépendance de  $\pi$  par rapport à  $\mathcal{C}$  sera souvent omise ; par contre, pour le problème inverse, nous la mentionnerons systématiquement puisque  $\mathcal{C}$  représente les contraintes des variables du problème inverse. Si  $\mathcal{C}$  n'est pas spécifié, le problème  $\text{INV}\pi$  désigne de manière générique l'ensemble des problèmes inverses associés aux divers ensembles  $\mathcal{C}$  envisageables. Par convention, lorsque  $\mathcal{C} = A_1 \times \dots \times A_n$  et si tous les ensembles  $A_i$  sont équipotents, les contraintes sont dites équilibrées ( $k$ -équilibrées si la cardinalité des  $A_i$  est finie et égale à  $k$ ,  $k$ -valuées si de plus  $A_1 = \dots = A_n$ ). Enfin, quand  $A_1 = \dots = A_n$ , on parle respectivement de problème inverse en variables *continues*, *continues et positives*, *entières*, *bornées*, lorsque  $A_1 = \mathbb{R}$ ,  $A_1 = \mathbb{R}^+$ ,  $A_1 = \mathbb{N}$  et  $A_1 = [a; b]$ ,  $a \leq b$ .

REMARQUE 1.– Cette présentation de la problématique combinatoire inverse suggère que seuls les paramètres de l'objectif sont à modifier. C'est effectivement la situation la plus courante sur laquelle nous avons choisi de mettre l'accent. Toutefois, pour certains problèmes, il est naturel de prendre comme variables du problème inverse des paramètres des contraintes ; nous l'illustrons par le problème de flot maximum inverse. Toutes les définitions peuvent s'adapter immédiatement au cas où les paramètres à modifier portent sur les contraintes ou conjointement sur les contraintes et l'objectif. La principale différence lorsque des variables du problème inverse se rapportent à des contraintes est que la solution fixée  $x_0$  n'est pas réalisable pour certaines valeurs des paramètres et peut notamment ne pas être réalisable pour le problème initial. Dans ce cas, l'existence même de solutions du problème inverse est à étudier au cas par cas.

La littérature propose deux extensions des problèmes d'optimisation inverse : les problèmes d'optimisation *inverse partiel* et *inverse évaluatif*<sup>1</sup>. Dans le premier cas, c'est une solution partielle  $x_j^0$  pour  $j \in J \subseteq \{1, \dots, m\}$  qui est donnée à la place de la solution complète  $x^*$  ; autrement dit, certaines composantes seulement du vecteur sont imposées et non plus le vecteur entier. On cherche alors à modifier aussi peu que possible les paramètres  $c^* \in \mathcal{C}$  de sorte à ce qu'il existe une solution optimale  $x^* = (x_1^*, \dots, x_n^*)$  de l'instance  $I = (c^*, \mathcal{D}, f)$  vérifiant  $x_j^* = x_j^0$  pour  $j \in J$ . Par exemple, pour le problème de l'arbre couvrant de poids minimum inverse partiel, étant données une instance  $I = (G, d)$  et une forêt  $E_0$  où  $G = (V, E)$  est un graphe connexe et  $d$  est une fonction distance de  $E$  dans  $\mathbb{R}^+$ , on cherche à trouver une fonction poids  $d^*$  pour laquelle la quantité  $\sum_{e \in E} |d(e) - d^*(e)|$  est minimum parmi les fonctions de distances  $d^*$  telles que  $E_0$  est comprise dans un arbre couvrant de poids minimum  $T^*$  de  $(G, d^*)$ . Dans [CAI 97], différentes études sur les problèmes inverses partiels ont été proposées. Un problème *inverse évaluatif* n'impose pas à la

---

1. Appelés parfois *reverse problems*.

solution de devenir optimale, mais d'égaliser une valeur donnée. Ainsi, étant données une instance  $I = (c, \mathcal{D}, f)$  de  $\pi_c$  et une valeur  $val^*$ , on cherche à modifier le moins possible les paramètres  $c^*$  de sorte à ce qu'il existe sur l'instance  $I^* = (c^*, \mathcal{D}, f)$  une solution optimale  $x^*$  vérifiant :  $f(c^*, x^*) = val^*$ . Plusieurs résultats concernant cette problématique ont été proposés dans [ZHANG 00].

Les notations et définitions relatives à la théorie des graphes employées dans ce chapitre sont celles de [BERGE 73]. Aussi, nous renvoyons le lecteur à cet ouvrage pour plus d'informations. Nous adoptons également la notation  $\vec{G} = (V, \vec{E})$  pour distinguer le cas d'un graphe orienté. Un arc sera alors désigné par  $\vec{e}$ .

### 1.3. Problèmes inverses polynomiaux et techniques de résolution

Dans cette section, nous proposons des exemples de problèmes inverses ou de classes de problèmes inverses connus pour être polynomiaux. Les méthodes employées jusqu'à présent pour résoudre des problèmes combinatoires inverses peuvent être classées en deux groupes :

1) les méthodes à base de programmation linéaire restent majoritaires, au moins pour le cas des normes  $L_1$  et  $L_\infty$ . Après établissement d'une formulation linéaire du problème inverse, les solutions proposées dans la littérature exploitent la méthode du simplexe révisée [AHUJA 01, ZHANG 96a, ZHANG 99]), la méthode des ellipsoïdes [LIU 03, YANG 99] ou encore des méthodes de génération de colonnes [ZHANG 96b, ZHANG 97, HU 98]. Même si, l'utilisation faite de ces méthodes pour différents problèmes inverses ne conduit pas toujours à un algorithme polynomial, nous les mentionnons ici car elles sont à la base de la plupart des preuves de cas polynomiaux actuellement connus.

2) pour certains problèmes, des algorithmes combinatoires ont été mis au point. Il s'agit parfois d'algorithmes résultant d'une solution explicite par programmation linéaire [AHUJA 01, HU 98, ZHANG 96b, ZHANG 97]. Nous proposons des solutions de ce type dans le paragraphe 1.3.1 pour un problème de cheminement et un problème de coupe. Dans d'autres cas, des méthodes combinatoires directes sont mises au point [ALFANDARI 04, YANG 97]; nous en évoquons dans les paragraphes 1.3.2 et 1.3.4.

Pour une cartographie complète des méthodes employées pour chaque problème, nous renvoyons le lecteur à [HEUBERGER 04].

Les paragraphes 1.3.1 et 1.3.3 fournissent deux situations canoniques de problèmes inverses résolus par programmation linéaire. Au paragraphe 1.3.1, nous expliquons pourquoi les versions inverses de programmes linéaires sont des programmes linéaires, en décrivons une solution et l'appliquons à un problème de cheminement ainsi qu'à un problème de coupe. Le paragraphe 1.3.3 décrit une classe relativement

large de problèmes combinatoires (ne se formulant pas directement par un programme linéaire), pour lesquels une version inverse pour la norme  $L_1$  ou  $L_\infty$  peut se formuler comme un programme linéaire. Pour cette classe, le problème inverse est polynomial dès que le problème initial l'est.

A côté de ces résultats à base de programmation linéaire, nous proposons au paragraphe 1.3.2, une démarche combinatoire pour un problème de flot maximum inverse de nature particulière puisqu'il s'agit de modifier les contraintes et non l'objectif. Pour différents problèmes, nous envisageons des versions inverses en variables entières qui ouvrent de nouveaux champs d'investigation. Nous proposons notamment en conclusion de cette section (paragraphe 1.3.4), un cas polynomial de problème de couplage inverse en variables 0,1 obtenu par un algorithme de flot. Nous montrons dans la section 1.4 que dans le cas général, ce dernier problème est **NP**-difficile.

### 1.3.1. Cas de la programmation linéaire

Le cas de la programmation linéaire, le vecteur  $c \in \mathbb{R}^m$  et  $f$  est bilinéaire :  $f(c, x) = \langle c, x \rangle$  où  $\langle x, y \rangle = \sum_{i=1}^m x_i y_i$  désigne le produit scalaire de  $x$  et  $y$ , et  $\mathcal{D}$  est défini par  $n$  contraintes linéaires. L'ensemble  $\mathcal{C}$ , quant à lui, peut être tout polyèdre de  $\mathbb{R}^m$  mais dans la plupart des cas naturels, il s'agit de  $\mathbb{R}^{+m}$  (poids positifs), de  $[a, b]^m$ , voire de  $\mathbb{R}^m$  lorsqu'aucune restriction n'est imposée.

Notre objectif étant de présenter les idées maîtresses à partir d'exemples simples et significatifs, nous nous limiterons à  $\mathcal{C} = \mathbb{R}^m$ . Considérons un problème linéaire  $PL$  dont chaque instance se formule sous forme standard de la manière suivante :

$$PL(c) \begin{cases} \min & \langle c, x \rangle \\ & Ax \geq b \\ & x \geq 0 \end{cases} \quad [1.1.a] \quad [1.1]$$

où  $c \in \mathbb{R}^m$ ,  $b \in \mathbb{R}^n$  et  $A$  est une matrice à  $n$  lignes et  $m$  colonnes et à coefficients réels. Bien entendu, l'instance est déterminée par  $c$ ,  $b$  et  $A$ , toutefois, nous ne mettons en évidence que la dépendance en  $c$  afin de pouvoir faire référence à un programme  $PL(c')$  obtenu à partir de  $PL(c)$  en remplaçant  $c$  par  $c'$ .

$PL$  admet  $x$  comme variable ; pour envisager sa version inverse, nous fixons  $c$  comme vecteur de paramètres initial et fixons  $x^*$  comme solution cible. Il s'agit de déterminer un vecteur objectif  $c^*$  minimisant  $\|c^* - c\|_1$  tel que  $x^*$  est solution optimale de  $PL(c^*)$ . Pour écrire plus formellement le problème inverse  $\text{INVPL}_{\mathbb{R}^m}$ , nous

utiliserons  $\tilde{c}$  pour exprimer la variable de ce problème inverse,  $c^*$  en étant alors une solution optimale. Chaque instance de la version inverse de  $PL$  s'écrit :

$$\text{INVPL}_{\mathbb{R}^m}(c, x^*) \left\{ \begin{array}{l} \min \quad \|\tilde{c} - c\|_1 \\ x^* \text{ solution de } PL(\tilde{c}) \end{array} \right. \quad [1.2]$$

REMARQUE 2.– Avant d'aller plus loin dans l'analyse de ce problème, notons que, comme pour tout problème inverse, une hypothèse implicite est que pour tout  $\tilde{c}$ ,  $PL(\tilde{c})$  admet au moins une solution réalisable  $x^*$ .

Notons même que 0 est évidemment une solution réalisable du problème inverse puisque  $x^*$  est une solution optimale du problème :

$$PL(0) \left\{ \begin{array}{l} \min \quad 0 \\ Ax \geq b \\ x \geq 0 \end{array} \right.$$

Nous verrons dans le théorème 1.1 qu'il existe toujours une solution au problème inverse.

Remarquons que  $PL(\tilde{c})$  admet une solution optimale si et seulement si son dual  $DL(\tilde{c})$  admet des solutions réalisables.  $DL(\tilde{c})$  se formule par :

$$DL(\tilde{c}) \left\{ \begin{array}{l} \max \quad \langle b, y \rangle \\ {}^tAy \leq \tilde{c} \\ y \geq 0 \end{array} \right. \quad [1.3.a] \quad [1.3]$$

Il suffit donc de limiter la recherche de  $\tilde{c}$  à l'ensemble  $S_{PL} = \{\tilde{c}, \exists y, {}^tAy \leq \tilde{c}, y \geq 0\} = \mathbb{R}^m \setminus \{x, Ax \geq 0, x \geq 0, x \neq 0\}$ . Bien entendu  $0 \in S_{PL}$ .

Le problème  $\text{INVPL}_{\mathbb{R}^m}(c, x^*)$  peut se formuler comme un programme linéaire. Pour cela, une possibilité est d'exprimer l'optimalité de  $x^*$  pour le problème  $PL(\tilde{c})$  par des contraintes primales-duales. Pour écrire les contraintes de complémentarité de manière explicite, on définit les notations suivantes :

$$\begin{aligned} I &= \{i \in \{1, \dots, n\}, \sum_{j=1}^m a_{i,j}x_j^* = b_i\} \\ J &= \{j \in \{1, \dots, m\}, x_j^* > 0\} \\ K &= \{j \in \{1, \dots, m\}, x_j^* = 0\} \end{aligned}$$

Nous désignerons de même par  $A_I$  la matrice extraite de  $A$  en sélectionnant les lignes dans  $I$ . On note  $\pi$  le vecteur des variables duales associées à la contrainte [1.3.a] ;

étant données les contraintes de complémentarité, les composantes non nulles de  $\pi$  sont d'indice dans  $I$ , donc seul le vecteur  $\pi_I$  est à déterminer.

$x^*$  est alors solution optimale de  $PL(\tilde{c})$  si et seulement si il existe  $(\pi_I, \lambda_K) \in \mathbb{R}^{|I|} \times \mathbb{R}^{|K|}$  tels que :

$$\begin{aligned} \sum_{i \in I} a_{i,j} \pi_i &= \tilde{c}_j & \forall j \in J & \quad [1.4.a] \\ \sum_{i \in I} a_{i,j} \pi_i + \lambda_j &= \tilde{c}_j & \forall j \in K & \quad [1.4.b] \\ \pi_I, \lambda_K &\geq 0 & & \quad [1.4.c] \end{aligned} \quad [1.4]$$

où  $\lambda_K$  permet d'écrire la condition [1.4.b] avec une égalité.

La linéarisation de l'objectif du problème  $\text{INVPL}_{\mathbb{R}^m}$  est classique : on introduit  $2m$  variables  $\alpha_i, \beta_i, i = 1, \dots, m$ , toutes positives, de sorte que  $\tilde{c} = c + \alpha - \beta$ .  $\text{INVPL}_{\mathbb{R}^m}$  s'écrit alors :

$$\text{INVPL}'_{\mathbb{R}^m}(c, x^*) \left\{ \begin{array}{l} \min \quad \sum_{j=1}^m (\alpha_j + \beta_j) \\ \sum_{i \in I} a_{i,j} \pi_i - \alpha_j + \beta_j = c_j, \quad \forall j \in J \quad [1.4.a'] \\ \sum_{i \in I} a_{i,j} \pi_i + \lambda_j - \alpha_j + \beta_j = c_j, \quad \forall j \in K \quad [1.4.b'] \\ \pi_I, \lambda_K, \alpha, \beta \geq 0 \end{array} \right.$$

Dans l'absolu, cette formulation suffit pour résoudre le problème inverse  $\text{INVPL}_{\mathbb{R}^m}$ , au moins dans le cas de coefficients rationnels. Toutefois, l'étude de ce programme permet d'obtenir une solution analytique et de déduire des solutions explicites pour divers exemples de problèmes combinatoires pouvant se formuler comme des programmes linéaires. Nous en verrons deux exemples pour un problème de cheminement et un problème de coupe de capacité minimum.

Comme seules les variables  $\alpha$  et  $\beta$  interviennent dans l'objectif et comme chaque indice  $j$  n'apparaît que dans l'une des contraintes [1.4.a', 1.4.b'], on a nécessairement à l'optimum  $\beta_K = 0$ , ce qui permet d'écrire le problème équivalent

$$\text{INVPL}''_{\mathbb{R}^m}(c, x^*) \left\{ \begin{array}{l} \min \quad \sum_{j=1}^m (\alpha_j + \beta_j) \\ \sum_{i \in I} a_{i,j} \pi_i - \alpha_j + \beta_j = c_j, \quad \forall j \in J \quad [1.4.a''] \\ \sum_{i \in I} a_{i,j} \pi_i + \lambda_j - \alpha_j = c_j, \quad \forall j \in K \quad [1.4.b''] \\ \pi_I, \lambda_K, \alpha_{K \cup J}, \beta_J \geq 0 \end{array} \right.$$

Dans ce problème, les différentes variables ne jouent pas le même rôle. Pour proposer une solution analytique, l'idée est de considérer le dual  $D''$  de  $\text{INVPL}''_{\mathbb{R}^m}$ , puis de

déterminer chaque variable de  $\text{INVPL}_{\mathbb{R}^m}$  comme une variable duale d'une contrainte de  $D''$ .

En notant  $\tilde{y}_j, j = 1, \dots, m$  les variables de  $D''$ , on obtient (pour chaque contrainte, on indique entre parenthèses la variable de  $\text{INVPL}_{\mathbb{R}^m}$  associée) :

$$D'' \left\{ \begin{array}{ll} \max & \langle c, \tilde{y} \rangle \\ & \sum_{j=1}^n a_{i,j} \tilde{y}_j \leq 0 \quad i \in I \quad (\pi_I) \\ & \tilde{y}_j \leq 0 \quad j \in K \quad (\lambda_K) \\ & -\tilde{y}_j \leq 1 \quad j \in \{1, \dots, m\} \quad (\alpha) \\ & \tilde{y}_j \leq 1 \quad j \in J \quad (\beta_J) \end{array} \right.$$

Cette formulation rappelle le programme initial. Pour s'en rapprocher encore, on fait le changement de variable  $y = (x^* - \tilde{y})$ . Notons qu'un tel changement de variable n'affecte pas le dual. On obtient alors, en utilisant  $A_I x^* = b_I$  et les définitions des ensembles  $J, K$  :

$$D\text{INVPL} \left\{ \begin{array}{ll} \min & \langle c, y \rangle \\ & A_I y \geq b_I \quad (\pi_I) \\ & y_j \geq 0 \quad j \in K \quad (\lambda_K) \\ & y_j \leq 1 \quad j \in K \quad (\alpha_K) \\ & y_j \leq 1 + x_j^* \quad j \in J \quad (\alpha_J) \\ & y_j \geq x_j^* - 1 \quad j \in J \quad (\beta_J) \end{array} \right.$$

Notons que  $D\text{INVPL}$  est proche du problème initial ; la différence est que seules les contraintes de  $PL$  saturées par  $x^*$  apparaissent dans  $D\text{INVPL}$  et que, par contre, ce dernier est un problème en variables bornées (contraintes associées à  $\alpha_J$  et  $\beta_J$ ). Or, il admet  $x^*$  comme solution réalisable et admet donc systématiquement une solution optimale ; il en est donc de même pour son dual.

On est maintenant à même de proposer une solution au problème inverse  $\text{INVPL}_{\mathbb{R}^m}$ . Soit  $y^*$  une solution de  $D\text{INVPL}$  et  $\pi_I^*$  la valeur de la variable duale associée à la contrainte  $A_I y \geq b_I$  dans une solution optimale de  $\text{INVPL}_{\mathbb{R}^m}$ . On a :

$$\begin{aligned} -\alpha_j + \beta_j &= c_j - \sum_{i \in I} a_{i,j} \pi_i^*, & \forall j \in J & \quad [1.5.a] \\ -\alpha_j &= c_j - \sum_{i \in I} a_{i,j} \pi_i^* - \lambda_j, & \forall j \in K & \quad [1.5.b] \end{aligned} \quad [1.5]$$

En posant  $c_j^\pi = c_j - \sum_{i \in I} a_{i,j} \pi_i^*$  pour tout  $j$ , la simple lecture des contraintes [1.5.a, 1.5.b] et de l'objectif de  $\text{INVPL}_{\mathbb{R}^m}$  permet de déterminer les valeurs optimales de  $\alpha_j$  et  $\beta_j$  en fonction du signe de  $c_j^\pi$ .

	$c_j^\pi > 0$	$c_j^\pi < 0$	$c_j^\pi = 0$
$j \in J$	$\left. \begin{array}{l} \beta_j = c_j^\pi \\ \alpha_j = 0 \end{array} \right\} \Rightarrow c_j^* = c_j - c_j^\pi$	$\left. \begin{array}{l} \alpha_j = -c_j^\pi \\ \beta_j = 0 \end{array} \right\} \Rightarrow c_j^* = c_j - c_j^\pi$	$\left. \begin{array}{l} \alpha_j = 0 \\ \beta_j = 0 \end{array} \right\} \Rightarrow c_j^* = c_j$
$j \in K$	$\left. \begin{array}{l} \alpha_j = 0 \\ \lambda_j = c_j^\pi \end{array} \right\} \Rightarrow c_j^* = c_j$	$\left. \begin{array}{l} \alpha_j = -c_j^\pi \\ \lambda_j = 0 \end{array} \right\} \Rightarrow c_j^* = c_j - c_j^\pi$	$\left. \begin{array}{l} \alpha_j = 0 \\ \lambda_j = 0 \end{array} \right\} \Rightarrow c_j^* = c_j$

**Tableau 1.1.** Discussion de l'équation [1.5] en fonction du signe de  $c_j^\pi$

On déduit donc les valeurs suivantes pour la solution optimale  $c_j^*$  de  $\text{INVPL}_{\mathbb{R}^m}$  sous la norme  $L_1$  :

$$c_j^* = \begin{cases} c_j - |c_j^\pi| & \text{si } c_j^\pi > 0 \text{ et } x_j^* > 0 \\ c_j + |c_j^\pi| & \text{si } c_j^\pi < 0 \\ c_j & \text{dans les autres cas} \end{cases} \quad [1.6]$$

On résume cette discussion par le théorème 1.1.

**THÉORÈME 1.1.**– [AHUJA 01] Soit  $PL$  un problème dont chaque instance s'écrit

$$PL(c) \begin{cases} \min & \langle c, x \rangle \\ & Ax \geq b \\ & x \geq 0 \end{cases}$$

Soit  $x^*$  une solution réalisable, on note  $I = \{i \in \{1, \dots, n\}, \sum_{j=1}^m a_{i,j} x_j^* = b_i\}$ ,  $J = \{j \in \{1, \dots, m\}, x_j^* > 0\}$  et  $K = \{j \in \{1, \dots, m\}, x_j^* = 0\}$ . Alors le problème  $\text{INVPL}_{\mathbb{R}^m}(c, x^*)$  admet une solution unique.

Pour la déterminer, on considère le problème :

$$D\text{INVPL} \begin{cases} \min & \langle c, y \rangle \\ & A_I y \geq b_I, \\ & y_j \geq 0 & j \in K \\ & y_j \leq 1 & j \in K \\ & y_j \leq 1 + x_j^* & j \in J \\ & y_j \geq x_j^* - 1 & j \in J \end{cases}$$

Soit alors  $y^*$  une solution optimale de  $D\text{INVPL}$  et  $\pi_1^*$  la valeur de la variable duale associée à la contrainte  $A_I y \geq b_I$  dans le dual de  $D\text{INVPL}$ , et soit  $c^\pi = c - {}^t A_I \pi^*$ .

L'unique solution de  $\text{INVPL}_{\mathbb{R}^m}(c, x^*)$  est donnée par :

$$c_j^* = \begin{cases} c_j - |c_j^\pi| & \text{si } c_j^\pi > 0 \text{ et } x_j^* > 0 \\ c_j + |c_j^\pi| & \text{si } c_j^\pi < 0 \\ c_j & \text{dans les autres cas} \end{cases}$$

Faisons quelques remarques avant d'exploiter ce résultat pour différents problèmes inverses.

REMARQUE 3.– Si l'on s'intéresse à des versions contraintes  $\text{INVPL}_{\mathcal{C}}$  où  $\mathcal{C}$  exprime des contraintes d'intervalle sur les coefficients de  $\tilde{c}$ , le problème inverse a également une formulation linéaire. Les contraintes supplémentaires pour  $\tilde{c}$  se traduisent en contraintes sur  $\alpha$  et  $\beta$  dans la formulation de  $\text{INVPL}'$ .

Si le problème linéaire dont on étudie une version inverse n'est pas sous la forme standard [1.1], on obtient aussi une formulation linéaire du problème inverse. Notons toutefois que tant que la version équivalente sous forme standard a les mêmes variables et le même objectif, le résultat ci-dessus peut directement être appliqué à cette forme standard. C'est en particulier le cas pour un problème sous forme canonique. Par contre, si le problème initial n'a pas de contrainte de signe, la version équivalente sous forme standard a plus de variables (chaque variable de signe quelconque est décomposée en deux variables positives), donc a un vecteur objectif de plus grande dimension. La version inverse de cette forme standard n'est pas équivalente au problème inverse initial puisque les coefficients dans l'objectif des variables issues d'une même variable de signe quelconque peuvent varier indépendamment.

Bien entendu, si les coefficients de  $\tilde{c}$  doivent être entiers ou de type 0,1, le problème inverse devient un programme linéaire en nombres entiers. Nous verrons que dans certains cas, de telles contraintes peuvent être prises en compte en temps polynomial. C'est notamment le cas pour un problème de cheminement et un problème de coupe de capacité minimum. Par contre, pour d'autres problèmes, cette restriction devient difficile.

REMARQUE 4.– Supposons que  $PL$  soit le relaxé d'un problème  $PLNE$  en variables bivalentes et que  $x^*$  est une solution de  $PLNE$ . Dans ce cas,  $PL$  est de la forme :

$$PL \left\{ \begin{array}{l} \min \quad \langle c, x \rangle \\ Ax \geq b \\ x \leq \mathbf{1} \\ x \geq 0 \end{array} \right.$$

En posant  $\tilde{A} = \begin{pmatrix} A \\ -Id \end{pmatrix}$ , on applique ce qui précède.  $J$  correspond aux indices pour lesquels  $x_j^* = 1$ , c'est-à-dire pour lesquels la contrainte  $x \leq \mathbf{1}$  est saturée. Le problème  $D$  est donc :

$$D \left\{ \begin{array}{l} \min \langle c, x \rangle \\ A_I y \geq b \\ y_J \leq \mathbf{1}_J \\ y_J \leq \mathbf{2}_J \\ y \geq 0 \end{array} \right.$$

où  $\mathbf{1}_J$  et  $\mathbf{2}_J$  correspondent aux vecteurs de dimension  $|J|$  dont les coordonnées valent respectivement 1 et 2. La contrainte  $y_J \leq \mathbf{2}_J$  est redondante et peut donc être supprimée. Si toutes les contraintes  $Ax \geq b$  sont saturées par  $x^*$ , ce problème correspond exactement à  $PL$ .

### 1.3.1.1. Problème de cheminement optimal inverse

Soit  $\vec{G} = (V, \vec{E})$  un graphe orienté arcs-valué; on note  $c(\vec{e})$ ,  $\vec{e} \in \vec{E}$  la valeur de l'arc  $\vec{e}$ ,  $n = |V|$  et  $m = |\vec{E}|$ . Soit  $s$  et  $t$  deux points spécifiés de  $V$ . On s'intéresse dans ce paragraphe au problème inverse de chemin de  $s$  à  $t$  de valeur minimum. On considère donc un chemin de  $s$  à  $t$ ,  $P_{s,t} \subset \vec{E}$ , vu comme un ensemble d'arcs. On désigne par  $x_{s,t}^*$  le vecteur caractéristique associé à ce chemin ( $x_{s,t}^* \in \mathbb{R}^m$ ). Le problème inverse associé à ce chemin consiste donc à modifier le vecteur des valuations pour rendre  $P_{s,t}$  optimal. Plus formellement, dans le cas de la norme  $L_1$ , le problème s'énonce :

$$\left\{ \begin{array}{l} \min \|\tilde{c} - c\|_1 \\ P_{s,t} \text{ chemin de } s \text{ à } t \text{ de valeur minimum pour } \tilde{c} \end{array} \right.$$

Remarquons que le cadre inverse impose *a priori* l'existence d'un chemin de  $s$  à  $t$ . Nous supposons en outre qu'aucun chemin de  $s$  à  $t$  rencontre un circuit absorbant. Dans ces conditions, le problème de cheminement de coût minimum de  $s$  à  $t$  admet des solutions et peut alors se formuler par le programme linéaire suivant :

$$\left\{ \begin{array}{l} \min \langle c, x \rangle \\ \sum_{j, (j,i) \in \vec{E}} x_{ji} - \sum_{j, (i,j) \in \vec{E}} x_{ij} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{sinon} \end{cases} \\ x \geq 0 \end{array} \right. \quad [1.7]$$

Un résultat classique est qu'en l'absence de circuit absorbant, ce problème admet des solutions optimales à composantes 0, 1 correspondant à un chemin optimal. Pour s'en rendre compte, il suffit de considérer le problème dual qui consiste à déterminer le potentiel  $\pi_s^*$  des chemins optimaux à partir de  $s$  vérifiant les conditions usuelles

d'optimalité des problèmes de cheminement. En attribuant aux arcs d'un chemin optimal (qui existe sous les hypothèses que nous avons faites) la valeur 1 et aux autres arcs la valeur 0, les arcs de ce chemin saturent les contraintes duales : la différence de potentiel vaut exactement le coût de l'arc. Toutes les contraintes du primal [1.7] étant saturées, les contraintes de complémentarité sont satisfaites, ce qui prouve l'optimalité de la solution proposée.

Un corollaire immédiat est qu'on peut ajouter des contraintes de bornes sur les variables sans changer la valeur du problème :

REMARQUE 5.– Le problème en variables bornées obtenu en ajoutant des contraintes  $x_{ij} \leq d_{ij}$  avec  $d_{ij} \geq 1$  dans le programme [1.7] admet aussi un chemin optimal comme solution.

Pour appliquer le théorème 1.1, la contrainte d'égalité est considérée comme la conjonction de deux inégalités. Notons que toute solution réalisable sature toutes les contraintes, donc  $I = \{1, \dots, n\}$ . Par ailleurs, l'ensemble  $J$  correspond aux arcs du chemin  $P_{s,t}$ . Le problème  $D$  correspond donc au problème [1.7] avec des contraintes de borne sur les variables :  $x_{ij} \leq 1$  ou  $x_{ij} \leq 2$ . D'après la remarque 5, il s'agit exactement du problème de cheminement de  $s$  à  $t$ . On note  $\pi^+$  et  $\pi^-$  les variables duales associées respectivement aux contraintes  $Ax \geq 1, -1, 0$  et  $-Ax \geq -1, 1, 0$ . Une solution optimale du dual est telle que  $\pi^* = (\pi^+)^* - (\pi^-)^*$  correspond au potentiel des chemins optimaux à partir de  $s$ . Pour un arc  $(i, j)$ , on a  $c_{ij}^\pi = c_{ij} - \pi^*(j) + \pi^*(i)$ . Pour tout arc, cette valeur est positive ou nulle, donc d'après le théorème 1.1, on en déduit le théorème 1.2 :

THÉORÈME 1.2.– [AHUJA 01] Soit  $\vec{G} = (V, \vec{E}, c)$  un graphe orienté arcs-pondéré,  $s$  et  $t$  deux sommets et  $P_{s,t} \subseteq \vec{E}$  un chemin de  $s$  à  $t$ . On note  $|V| = n$  et  $|\vec{E}| = m$ . Alors une solution du problème inverse  $\text{INVCHÉMINMIN}_{\mathbb{R}^m}$  associé à  $c$  et  $P_{s,t}$  pour la norme  $L_1$  peut être déterminée de la manière suivante : on détermine  $\pi^*$  le potentiel optimal des chemins à partir de  $s$  :

$$c_{ij}^* = \begin{cases} \pi^*(j) - \pi^*(i), & (i, j) \in P_{s,t} \\ c_{ij} & \text{sinon} \end{cases}$$

Pour justifier pleinement ce résultat, il faut remarquer que la transformation des coûts ne crée pas de circuit absorbant. En effet, le potentiel  $\pi^*$  vérifie, pour tout arc  $(i, j)$ ,  $\pi^*(j) - \pi^*(i) \leq c_{ij}^*$ . En sommant ces inégalités sur un circuit, on montre que le coût, après transformation, d'un circuit est positif.

Notons enfin que cette solution du problème inverse ne modifie pas la valeur des potentiels optimaux à partir de  $s$ .

Remarquons par ailleurs que si tous les coûts des arcs sont entiers, alors il en est de même du potentiel des chemins optimaux et de la solution déterminée par le théorème 1.2, d'où le corollaire 1.1 :

**COROLLAIRE 1.1.**— Si les coûts des arcs sont des entiers relatifs, alors le problème  $\text{INVCHÉMINMIN}_{\mathbb{R}^m}$  admet une solution optimale à valeurs entières et donc les problèmes  $\text{INVCHÉMINMIN}_{\mathbb{R}^m}$  et  $\text{INVCHÉMINMIN}_{\mathbb{Z}^m}$  sont équivalents.

### 1.3.1.2. Problème de coupe de capacité minimum inverse

Un second exemple classique d'utilisation du théorème 1.1 est la résolution d'un problème de coupe de capacité minimum inverse. Pour cet exemple, nous choisissons une formulation non polynomiale du problème de coupe dans un réseau de transport. Étant donné un graphe orienté  $\vec{G} = (V, \vec{E})$  et deux sommets  $s$  et  $t$ , on suppose que chaque arc  $(i, j)$  est muni d'une capacité  $c_{ij} > 0$ . Une *coupe séparant  $s$  et  $t$*  est un ensemble de sommets  $K \subset V$  contenant  $s$  et pas  $t$ . On associe à  $K$  le *cocycle sortant* (voir [GONDRAN 98]),  $\Omega^+(K) = \{(i, j) \in \vec{E}, i \in K \text{ et } j \notin K\}$ . La capacité de  $K$  est la somme des capacités des arcs du cocycle associé (les capacités inférieures sont nulles dans cet exemple). Le problème de coupe (séparant  $s$  et  $t$ ) de capacité minimum, noté  $\text{COUPEMIN}$ , consiste à déterminer une coupe séparant  $s$  et  $t$  de capacité minimum. Pour la version inverse, on se donne une coupe  $K^*$  séparant  $s$  et  $t$  et il s'agit de modifier les capacités afin de rendre cette coupe de capacité minimum. Formellement, le problème *de la coupe de capacité minimum inverse*, noté  $\text{INVCOUPEMIN}$ , consiste à trouver une fonction de capacité  $c^*$  pour laquelle :

(i) le sous-ensemble de sommets  $K^*$  donné à l'avance avec  $s \in K^*$  et  $t \notin K^*$  est une coupe de capacité minimum du réseau  $(\vec{G}, c^*)$  ;

(ii) que la quantité  $\sum_{(i,j) \in \vec{E}} |c_{ij} - c_{ij}^*|$  est minimum.

Pour formuler  $\text{INVCOUPEMIN}_{\mathbb{R}^m}$  par la programmation linéaire, on considère  $C(\vec{G}, s, t)$  l'ensemble des chemins de  $s$  à  $t$  dans  $\vec{G}$ . On a alors la formulation suivante :

$$\begin{cases} \min & \langle c, x \rangle \\ & \sum_{(i,j) \in P} x_{ji} \geq 1, \text{ pour tout } P \in C(\vec{G}, s, t) \\ & x \geq 0 \end{cases} \quad [1.8]$$

Pour toute solution réalisable  $x$  à composantes 0,1, on note  $\vec{E}(x) = \{(i, j) \in \vec{E} : x_{ij} = 1\}$ . Si  $x$  est minimale (c'est-à-dire  $\forall y \leq x, y \neq x, y$  n'est pas réalisable), il existe une coupe  $K$  séparant  $s$  et  $t$  telle que  $\vec{E}(x) = \Omega^+(K)$ . Il suffit en effet de prendre pour  $K$  l'ensemble des sommets accessibles à partir de  $s$  par un chemin ne comprenant aucun arc de  $\vec{E}(x)$ . Dans ce cas, la capacité de  $K$  vérifie  $c(K) = \langle c, x \rangle$ .

Réciproquement, toute coupe séparant  $s$  et  $t$  correspond à une solution réalisable minimale à composantes 0,1 du programme [1.8].

Comme dans le cas du problème de cheminement, le programme linéaire [1.8] admet une solution optimale à composantes 0,1 qui correspond à une coupe de capacité minimum.

Pour s'en rendre compte, il suffit de considérer le dual du problème [1.8] qui s'écrit :

$$\left\{ \begin{array}{l} \max \sum_{P \in \mathcal{C}(\vec{G}, s, t)} y_P \\ \sum_{P \in \mathcal{C}(\vec{G}, s, t), (i, j) \in P} y_P \leq c_{ij}, \text{ pour tout } (i, j) \in \vec{E} \\ y \geq 0 \end{array} \right. \quad [1.9]$$

A partir d'un flot compatible (positif et satisfaisant les capacités sur chaque arc) maximum  $\phi^*$  de  $s$  à  $t$ , il est possible de décomposer  $\phi^*$  comme la somme de flots positifs  $\phi^* = \phi_1 + \dots + \phi_k$  où chaque  $\phi_i$  est nul sauf sur un chemin  $P_i$  de  $s$  à  $t$ . Pour construire une telle décomposition, on peut par exemple remarquer que si  $\phi^*$  est non nul, il existe nécessairement un chemin  $P_1$  de  $s$  à  $t$  portant, sur chacun de ses arcs, un flux strictement positif.  $\phi_1$  a pour valeur  $v(P_1)$  le plus petit flux des arcs de  $P_1$ , chaque arc de  $P_1$  portant alors  $v(P_1)$ . Il suffit alors de réitérer ce processus sur  $\phi^* - \phi_1$ . Une autre manière de procéder est de reprendre l'algorithme de Ford et Fulkerson en remarquant que pour chaque augmentation le long d'une chaîne, si le flot courant se décompose, en  $\phi_1 + \dots + \phi_l$ , le nouveau flot se décompose en  $\phi'_1 + \dots + \phi'_l + \phi'_{l+1}$  (les arcs inverses de la chaîne impliquent une redistribution des flots). On définit alors la solution duale  $y_{P_i} = v(\phi_i)$ ,  $i = 1, \dots, k$  et  $y_P = 0$  pour tout autre chemin. Comme  $\phi^*$  est compatible, cette solution est réalisable pour le programme linéaire [1.9] et de valeur  $v(\phi^*)$ . Par conséquent, la valeur du programme [1.8] vaut  $v(\phi^*)$ , c'est-à-dire la capacité minimum d'une coupe. Ceci prouve qu'une coupe de capacité minimum correspond à une solution optimale du programme linéaire [1.8] qui est à composantes 0,1. On peut donc ajouter au problème [1.8] des contraintes du type  $x_{ij} \leq l_{ij}$ , avec  $l_{ij} \geq 1$  sans changer la valeur optimale ; le problème admet toujours une solution optimale à composantes 0,1 associée à une coupe de capacité minimum.

Considérons maintenant le problème inverse  $\text{INVCOUPEMIN}_{\mathbb{R}^+}^{m}$  associé à une coupe  $K^*$  séparant  $s$  et  $t$ . Soit  $x^*$  le vecteur caractéristique de  $\Omega^+(K^*)$ . Tout chemin de  $s$  à  $t$  rencontre nécessairement  $\Omega^+(K^*)$  au moins une fois. Les chemins de  $s$  à  $t$  qui rencontrent  $\Omega^+(K^*)$  exactement une fois correspondent aux chemins de  $s$  à  $t$  dans le graphe  $\vec{G}'$  obtenu à partir de  $\vec{G}$  en supprimant les arcs entrant dans  $K^*$  (de type  $(i, j)$ ,  $i \in V \setminus K^*$ ,  $j \in K^*$ ). Ainsi, le problème  $DINV$  associé à  $K^*$  est :

$$DINV \left\{ \begin{array}{l} \min \langle c, y \rangle \\ \sum_{(i,j) \in P} y_{ji} \geq 1, \text{ pour tout } P \in C(\vec{G}, s, t) \\ y \leq 1 + x_{ij}^* \\ y \geq 0 \end{array} \right.$$

D'après la discussion précédente, une solution optimale de ce problème est à composantes 0,1 et correspond à une coupe de capacité minimum séparant  $s$  et  $t$  dans le graphe  $\vec{G}' = (V, \vec{E}')$ . Une solution optimale du dual est donnée par une décomposition d'un flot  $\tilde{\phi}^*$  de valeur maximum de  $s$  à  $t$  dans  $\vec{G}'$ . En notant  $\pi$  les variables duales associées aux contraintes  $\sum_{(i,j) \in P} y_{ji} \geq 1$  de  $DINV$ , la quantité  $c_{ij}^\pi$  associée à  $\pi$  pour l'arc  $(i, j)$  vaut exactement  $c_{ij} - \tilde{\phi}_{ij}^*$  où  $\phi_{ij}^*$  est la valeur de flot  $\phi^*$  sur l'arc  $(i, j)$ . Le flot étant compatible,  $c_{ij}^\pi \geq 0, \forall (i, j) \in \vec{E}'$ . D'après le théorème 1.1, on déduit le théorème 1.3 :

**THÉORÈME 1.3.**– [AHUJA 01] Soit  $\vec{G} = (V, E, c)$  un réseau de transport,  $s$  et  $t$  deux sommets et  $K^* \subset V$  une coupe séparant  $s$  à  $t$ . Alors une solution du problème inverse  $\text{INVCOUPEMIN}_{\mathbb{R}+m}$  associé à  $c$  et  $K^*$  pour la norme  $L_1$  peut être déterminée de la manière suivante :

1) on construit le graphe  $\vec{G}'$  obtenu à partir de  $\vec{G}$  en effaçant les arcs entrant dans  $K^*$  ;

2) on détermine un flot  $\tilde{\phi}^*$  de valeur maximum de  $s$  à  $t$  dans  $\vec{G}'$  ;

$$3) c_{ij}^* = \begin{cases} \tilde{\phi}_{ij}^*, & (i, j) \in \Omega^+(K^*) \\ c_{ij} & \text{sinon} \end{cases}$$

Notons que si toutes les capacités sont entières, l'algorithme de Ford et Fulkerson détermine un flot maximum  $\phi^*$  à composantes entières et dans ce cas, la solution déterminée par le théorème 1.3 est à composantes entières, d'où le corollaire 1.2 :

**COROLLAIRE 1.2.**– Si tous les arcs ont un coût entier,  $\text{INVCOUPEMIN}_{\mathbb{R}+m}$  admet une solution optimale à composantes entières déterminée par le théorème 1.3 ; il y a donc équivalence entre  $\text{INVCOUPEMIN}_{\mathbb{R}+m}$  et  $\text{INVCOUPEMIN}_{\mathbb{N}+m}$ .

### 1.3.2. Problème de flot maximum inverse

La démarche proposée au paragraphe 1.3.1 s'applique directement au problème de flots de coût minimum de  $s$  à  $t$  dans un réseau de transport. Sous sa version usuelle, ce problème peut en effet se formuler par un programme linéaire du type :

$$\left\{ \begin{array}{l} \min \langle p, x \rangle \\ \sum_{j, (j,i) \in \vec{E}} x_{ji} - \sum_{j, (i,j) \in \vec{E}} x_{ij} = b(i) \\ x \leq c_{ij} \\ x \geq 0 \end{array} \right. \quad [1.10]$$

où  $b(s) \leq 0$ ,  $b(t) = -b(s)$  et  $b(i) = 0$  en tout autre nœud.  $p_{ij}$  est le prix unitaire du flux sur l'arc  $(i, j)$  et  $c_{ij}$  est la capacité de l'arc  $(i, j)$ . Le problème inverse associé consiste à ajuster le prix pour rendre optimal un flot fixé. Ce problème entre pleinement dans le cadre du théorème 1.1 ; la résolution est très proche de ce que nous avons présenté pour le problème de cheminement (voir [AHUJA 01], et [HEUBERGER 04] pour des références détaillées), qui est un cas particulier de flot de coût minimum. De même, plusieurs généralisations peuvent être introduites, notamment la prise en compte de capacités inférieures.

Dans le cas du problème de flot maximum FLOTMAX, la problématique inverse est différente dans la mesure où il ne s'agit pas d'un problème pondéré. Ce problème a été étudié dans [YANG 97] ; nous proposons ici une analyse combinatoire dans le cas d'un réseau de transport, le cas général pourrait être traité de manière analogue. Etant donné un réseau de transport muni d'un sommet origine  $s$  et d'un sommet destination  $t$  et un flot  $\phi^*$  de  $s$  à  $t$ , il s'agit de modifier les capacités des arcs (c'est-à-dire les contraintes et non l'objectif comme dans le cadre inverse usuel) pour rendre  $\phi^*$  optimal, à savoir de flux maximum. L'objectif consiste à minimiser, au sens de la norme  $L_1$ , la modification du système de capacités. On note  $\phi_{ij}^*$  le flux de l'arc  $(i, j)$  et  $c_{ij} > 0$  sa capacité. On suppose  $\phi_{ij}^* \geq 0$  pour tout  $(i, j)$ . Pour ce type de problème inverse portant sur les contraintes, il n'y a pas de raison particulière de supposer la solution fixée réalisable. Nous supposons juste  $\phi^*$  positif pour rester dans le cadre d'un réseau de transport. Pour toute solution réalisable de INVFLOTMIN,  $\phi^*$  doit être un flot compatible de sorte que pour tout arc  $(i, j)$  tel que  $c_{ij} < \phi_{ij}^*$ , il faut au moins ajouter  $\phi_{ij}^* - c_{ij}$  à la capacité de  $(i, j)$ . Ainsi, en notant  $v_{ij} = \max\{0, \phi_{ij}^* - c_{ij}\}$ , toute solution réalisable  $\tilde{c}$  du problème inverse satisfait  $\|\tilde{c} - c\|_1 = \|(\tilde{c} - v) - c\|_1 + \|v\|_1$ . Il s'agit donc de minimiser  $\|\tilde{c} - (v + c)\|_1$  de sorte que  $\phi^*$  soit optimal dans le réseau muni des capacités  $\tilde{c}$ . Le problème initial se ramène donc au problème inverse pour le réseau muni des capacités  $c + v$ , réseau pour lequel  $\phi^*$  est réalisable. Par ailleurs, si  $K$  est une coupe de capacité minimum dans le graphe muni des capacités  $c^*$  (solution optimale du problème inverse), d'après le théorème du flot maximum et de la coupe

de capacité minimum (voir par exemple [GONDRAN 98]) les arcs  $(i, j)$  sortant de  $K$  vérifient  $\phi_{ij}^* = c_{ij}^*$ , de sorte que  $\|c^* - (v + c)\|_1 \geq \sum_{i \in K, j \notin K, (i, j) \in \vec{E}} v_{ij} + c_{ij} - \phi_{ij}^*$ .

En considérant le graphe  $\vec{G}$  muni de capacités  $v_{ij} + c_{ij} - \phi_{ij}^* \geq 0$  (on note  $\vec{G}'$  le graphe ainsi obtenu),  $\|c^* - (v + c)\|_1$  est au moins égal à la capacité de  $K$ , donc au moins égale à la capacité minimum d'une coupe séparant  $s$  et  $t$  dans  $\vec{G}'$ . Inversement, considérons  $K^*$  une coupe de capacité minimum dans  $\vec{G}'$  et en posant  $\tilde{c}_{ij} = \phi_{ij}^*$  pour tout arc  $(i, j)$  sortant de  $K^*$  et  $\tilde{c}_{ij} = v_{ij} + c_{ij}$  pour tout autre arc,  $\|\tilde{c} - (v + c)\|_1$  vaut exactement la capacité de  $K^*$  dans  $\vec{G}'$ ; par ailleurs, comme  $\phi^*$  sature tous les arcs sortant de  $K^*$ ,  $\phi^*$  est un flot de  $s$  à  $t$  de valeur maximum dans  $\vec{G}'$ . On a donc  $\tilde{c} = c^*$ , d'où le théorème 1.4 :

**THÉORÈME 1.4.**— Soit  $R = (\vec{G} = (V, \vec{E}), c, s, t)$  un réseau de transport, ( $c \geq 0$  est le système de capacités,  $s$  le sommet origine et  $t$  le sommet destination) et  $\phi^*$  un flot positif. Alors une solution  $c^*$  du problème inverse pour la norme  $L_1$  associé à  $R$  et  $\phi^*$  peut être déterminée de la manière suivante :

- construire le graphe  $\vec{G}'$  en affectant à tout arc  $(i, j)$  la capacité  $\max\{0; c_{ij} - \phi_{ij}^*\}$ ;
- déterminer  $K^*$ , coupe de capacité minimum séparant  $s$  et  $t$  dans  $\vec{G}'$ ;
- $c_{ij}^* = \begin{cases} \phi_{ij}^* & \text{si } i \in K^* \text{ et } j \notin K^* \\ \max\{c_{ij}; \phi_{ij}^*\} & \text{sinon} \end{cases}$

Un résultat du même type peut être obtenu sur un réseau général [YANG 97].

**REMARQUE 6.**— Remarquons que si  $c$  et  $\phi^*$  sont à composantes entières, la solution déterminée est à composantes entières et est donc solution de  $\text{INVFLotMAX}_{\mathbb{N}^m}$ . Si  $\phi^*$  n'est pas à composantes entières (mais  $c$  l'est), le problème  $\text{INVFLotMAX}_{\mathbb{N}^m}$  reste pertinent. On commence par poser  $c'_{ij} = \max\{c_{ij}, \lceil \phi_{ij}^* \rceil\}$ . On construit alors  $\vec{G}'$  en affectant à tout arc  $(i, j)$  la capacité  $c'_{ij} - \phi_{ij}^*$ ; il s'agit alors de déterminer dans  $\vec{G}'$  une coupe  $K^*$  de capacité minimum parmi les coupes dont les arcs sortant portent un flux entier. Dans  $\vec{G}'$ , on remplace toute capacité non entière par  $+\infty$ . Si le graphe  $\vec{G}''$  résultant admet une coupe de capacité minimum de capacité finie, on obtient une solution optimale pour  $\text{INVFLotMAX}_{\mathbb{N}^m}$  en ramenant à  $\phi_{ij}^*$  les capacités  $c'$  sur les arcs sortant d'une telle coupe; sinon, le problème  $\text{INVFLotMAX}_{\mathbb{N}^m}$  n'a pas de solution.

### 1.3.3. Une classe de problèmes inverses polynomiaux

Le théorème 1.1 fournit un résultat d'existence et une méthode de résolution polynomiale pour une large classe de problèmes. Dans le cas général, les problèmes de cette classe (problèmes linéaires) admettent une infinité de solutions réalisables.

Pour la plupart des problèmes combinatoires, le problème admet un nombre fini de solutions réalisables. Dans le cas de problèmes combinatoires pouvant se formuler comme un programme linéaire tel que le problème de cheminement ou le problème de coupe de capacité minimum, seules certaines solutions réalisables du problème linéaire sont des solutions du problème combinatoire (les solutions à composantes 0,1 dans le cas du problème de cheminement par exemple et les solutions minimales à composantes 0,1 dans le cas de la coupe). L'équivalence des deux formulations consiste à montrer que ces conditions peuvent être imposées à l'optimum.

Considérons un problème combinatoire dont chaque instance admet un nombre fini de solutions et tel que l'objectif est linéaire. Un tel problème s'exprime donc :

$$PC(c) \left\{ \begin{array}{l} \min \\ x \in \mathcal{D} \end{array} \langle c, x \rangle \right. \quad [1.11]$$

où  $|\mathcal{D}|$  est fini. Notons que la plupart des problèmes combinatoires entrent dans ce cadre. Etant donnée une solution réalisable  $x^* \in \mathcal{D}$ , le problème inverse associé à  $x^*$  et  $c$  pour la norme  $L_1$  se formule :

$$INVPC(c, x^*) \left\{ \begin{array}{l} \min \\ x^* \text{ solution de } PC(\tilde{c}) \end{array} \|\tilde{c} - c\|_1 \right. \quad [1.12]$$

Ce problème admet la formulation linéaire suivante :

$$\left\{ \begin{array}{ll} \min & \sum_{i=1}^n z_i \\ & \langle \tilde{c}, x^* \rangle \leq \langle \tilde{c}, x \rangle, \forall x \in \mathcal{D} & [1.13.a] \\ & \tilde{c}_i - c_i \leq z_i & [1.13.b] \\ & c_i - \tilde{c}_i \leq z_i & [1.13.c] \end{array} \right. \quad [1.13]$$

Il s'agit bien d'un programme linéaire dans la mesure où  $\mathcal{D}$  est fini. Notons qu'en posant  $z_i = |c_i|$  et  $\tilde{c} = 0$  on définit une solution réalisable du programme [1.13]. Par ailleurs, les équations [1.13.b] et [1.13.c] impliquent que le problème est minoré ( $z \geq 0$ ). Le programme [1.13] admet donc toujours une solution optimale.

Pour mettre en évidence des cas où ce problème est polynomial, nous utilisons l'argument de l'équivalence entre optimisation et séparation [GRÖTSCHEL 81] (voir aussi [KORTE 02]) qui consiste à remarquer que la méthode des ellipsoïdes ne requiert pas la formulation complète d'un programme linéaire mais juste la connaissance d'un oracle séparateur. La méthode des ellipsoïdes peut être appliquée à des

programmes linéaires à coefficients rationnels. C'est en particulier le cas du programme [1.13] si le vecteur coût initial est rationnel et si  $\mathcal{D} \subset \mathbb{Q}^m$ , ce qui peut être supposé pour quasiment tous les problèmes.

Le problème de séparation associé consiste, étant donné un couple  $(z, \tilde{c})$ , à décider si il est réalisable pour le problème [1.13] et, dans le cas négatif, à déterminer  $(u, v)$  tel que  $\langle \tilde{z}', u \rangle + \langle c', v \rangle < \langle \tilde{z}, u \rangle + \langle c, v \rangle$  pour tout  $(z', c')$  réalisable.

Si la contrainte [1.13.b] (resp. [1.13.c]) n'est pas satisfaite, il suffit de poser  $u_i = v_i = 1$  et  $u_j = v_j = 0, j \neq i$ . Enfin, pour déterminer si la contrainte [1.13.a] est satisfaite, il suffit de résoudre le problème  $PC(\tilde{c})$  : en notant  $\hat{x}$  une solution optimale, si  $\langle \tilde{c}, \hat{x} \rangle = \langle \tilde{c}, x^* \rangle$ , la contrainte est satisfaite ; sinon,  $\langle \tilde{z}', x^* - \hat{x} \rangle \leq 0 < \langle \tilde{c}, x^* - \hat{x} \rangle$  et alors  $u = x^* - \hat{x}, v = 0$  définit un hyperplan séparateur. Ainsi, si  $PC(d)$  est polynomial pour tout  $d \in \mathbb{Q}^m$ , le problème de séparation associé au problème [1.13] est polynomial et alors, par la méthode des ellipsoïdes [GRÖTSCHHEL 81], le problème inverse [1.13] est polynomial.

Notons enfin, que le problème de séparation n'est pas plus difficile si l'on intègre des contraintes de bornes sur les coefficients de  $\tilde{c}$ . Le seul problème pouvant survenir est que le problème inverse contraint peut ne pas avoir de solution, ce qui sera détecté en temps polynomial.

THÉORÈME 1.5.– [AHUJA 01] Soit  $PC$  un problème combinatoire du type

$$PC \left\{ \begin{array}{l} \min \langle c, x \rangle \\ x \in \mathcal{D} \end{array} \right. \quad [1.14]$$

tel que  $|\mathcal{D}|$  est fini pour chaque instance et  $\{c\} \cup \mathcal{D} \subset \mathbb{Q}^m$ . Si  $PC$  est polynomial (l'objectif  $c$  faisant partie de l'instance et pouvant prendre toute valeur de  $\mathbb{Q}^m$ ), alors le problème inverse associé à  $PC$  est polynomial.

REMARQUE 7.– Réciproquement, si le problème inverse est polynomial, il offre immédiatement un test d'optimalité polynomial : une solution réalisable  $x$  est optimale si et seulement si l'instance associée du problème inverse a pour valeur 0. Pour la plupart des problèmes combinatoires **NP**-difficiles connus, il est très facile de montrer qu'un tel test ne peut exister si  $\mathbf{P} \neq \mathbf{NP}$ . Nous donnons, en introduction de la section 1.4, deux exemples d'application de cet argument.

REMARQUE 8.– Notons aussi que si  $\mathcal{C} \subset \mathbb{Z}^m$ , le programme résultant est un programme linéaire en nombres entiers et le résultat précédent ne s'applique plus. Nous montrerons notamment dans la section 1.4 des exemples pour lesquels le problème inverse devient difficile dans ce cas.

REMARQUE 9.– Notons enfin, que même si un problème de programmation linéaire admet une infinité de solutions, il suffit de considérer le problème consistant à déterminer une solution optimale de base. Les solutions de base sont en nombre fini et déterminer une solution optimale de base est polynomial (pour un programme rationnel) d’après le théorème de Khachiyan [KHACHIYAN 79]. En considérant le problème combinatoire dont les solutions réalisables sont la solution fixée  $x^*$  d’une part et les solutions de base d’autre part, le théorème 1.5 permet de retrouver le caractère polynomial de INVPL quelle que soit la forme initiale du programme linéaire dans la mesure où il existe des sommets extrémaux. Notons que  $x^*$  peut ne pas être une solution de base.

#### 1.3.4. Pistes à explorer : l’exemple du couplage de poids maximum inverse en variables bivalentes

Les problèmes inverses en variables entières ouvrent un champs très large de problèmes combinatoires. Nous en avons déjà vu deux exemples pour un problème de cheminement et un problème de coupe. Jusqu’à présent, presque tous les problèmes inverses étudiés sont des problèmes en variables continues :  $\mathcal{C}$  est le plus souvent  $\mathbb{R}^m$  ou  $\mathbb{R}^{+m}$ . Si le cadre inverse usuel ne concerne que des problèmes pondérés (lorsqu’il s’agit de modifier l’objectif) ou faisant au moins intervenir des paramètres réels dans l’expression d’une instance (voir le cas du flot maximum), le cadre inverse en variables 0,1, quant à lui, est déjà naturel lorsque le problème d’origine est non pondéré. Il permet aussi d’exprimer des situations où l’on modifie la structure de l’instance pour rendre une solution prédéterminée optimale. Enfin, le cas  $\mathcal{C} \subset \mathbb{N}^m$  apparaît souvent comme restriction d’un problème inverse usuel lorsque les coefficients de l’objectif correspondent à des grandeurs discrètes. Comme nous l’avons vu précédemment, si le problème d’origine a un objectif linéaire et un nombre fini de solutions réalisables ou encore des contraintes linéaires, le cas  $\mathcal{C} \subset \mathbb{N}^m$  se formule par un programme linéaire en variables entières.

Le dernier exemple de cas polynomial concerne le problème de couplage maximum (ou de poids maximum) inverse en variables 0,1, noté  $\text{INVCPM}_{\{0,1\}^m}$ . Le cas continu ( $\text{INVCPM}_{\mathbb{R}^m}$ ) est polynomial d’après le théorème 1.5. Nous montrons (voir [ALFANDARI 04]) dans la section 1.4 que  $\text{INVCPM}_{\mathbb{N}^m}$  est **NP**-difficile dans un graphe général (en fait le problème  $\text{INVCPM}_{\{0,1\}^m}$  est également **NP**-difficile). Nous montrons aussi qu’une version bivaluée d’un autre problème de couplage (le couplage parfait de poids maximum) reste difficile dans les graphes bipartis. Dans cette section, nous montrons que  $\text{INVCPM}_{\{0,1\}^m}$  devient polynomial dans des graphes bipartis lorsque les arêtes du couplage fixé  $M^*$  sont toutes valuées par 1. La preuve est à base de techniques de flots et ne se déduit pas directement du théorème 1.1 comme pour le cas des problèmes de cheminement et de coupe.

Dans un graphe  $G = (V, E)$  à  $n$  sommets, un *couplage* est un ensemble d'arêtes  $M \subseteq E$  deux à deux non adjacentes ; les sommets incidents à  $M$  sont dits *saturés* par  $M$ , tandis que les autres sont dits *insaturés*. Un couplage est *maximal* lorsqu'il n'existe pas de couplage le contenant strictement. Il est *maximum* s'il est de cardinalité maximum (de valeur maximum dans le cas pondéré). La détermination d'un couplage maximum dans un graphe peut se faire en  $O(\sqrt{|V|}|E|)$  [MICALI 80] (problème pondéré en  $O(|V|^3)$ , voir [GONDRAN 98]), et utilise le principe des chaînes alternées augmentantes. Rappelons qu'une chaîne alternée augmentante relativement à un couplage  $M$  est une chaîne alternant arêtes de  $E \setminus M$  et de  $M$ , dont les deux sommets extrémités sont insaturés par  $M$ . Un couplage est maximum si et seulement si il n'existe pas de chaîne alternée augmentante [BERGE 73]. Pour le cas pondéré (CPM), la notion de chaîne alternée augmentante se généralise et le problème reste polynomial, nous y revenons au paragraphe 1.4.4.

Etant donné un réseau  $(G = (V, E), w)$  et un couplage  $M^*$  de  $G$  avec  $|V| = n$ ,  $|E| = m$  et  $w(e) \in \{0, 1\}^m$ , le problème *du couplage maximum inverse* en variables  $0, 1$ ,  $\text{INVCPM}_{\{0,1\}^m}$ , consiste à trouver une fonction poids  $w^*$  telle que :

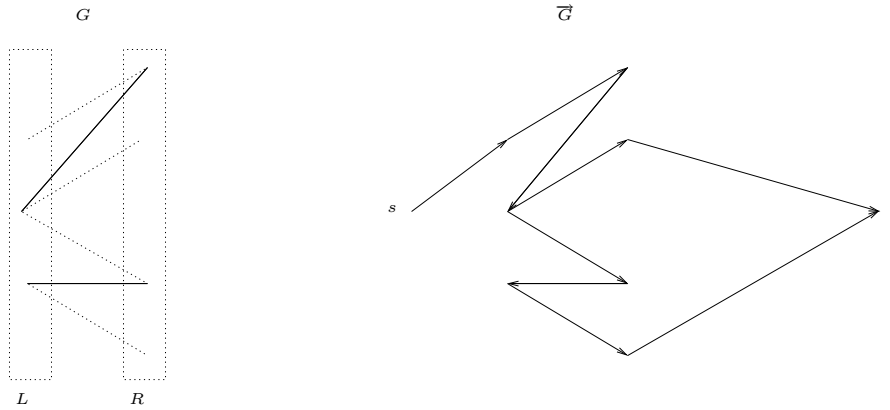
- i) le couplage  $M^*$  est de poids maximum dans  $(G, w^*)$  ;
- ii) la quantité  $\sum_{e \in E} |w(e) - w^*(e)|$  est minimum ;
- iii)  $w^* \in \{0, 1\}^m$ .

Mentionnons ici un cas particulier intéressant. Soit  $G = (V, E)$  un graphe et  $M^*$  un couplage. On considère le graphe complet à  $n = |V|$  sommets (ou tout graphe contenant  $G$  comme graphe partiel) et on définit un système de poids  $w \in \{0, 1\}^{|V|}$  représentant les arêtes présentes dans  $G$  ( $w(e) = 1, \forall e \in E$ ). Le problème inverse à composantes  $0, 1$  associé correspond au cas où les arêtes du couplage fixé (ici  $M^*$ , couplage de  $G$ ) sont de poids 1. Il s'agit alors de modifier l'instance (ajouter ou retrancher des arêtes) pour rendre  $M^*$  de cardinal maximum dans le graphe modifié. Il est clair dans ce cas qu'il n'est jamais intéressant de modifier une valeur d'arête de 0 à 1 de sorte que l'on peut supposer  $w^* \leq w$ . De même,  $w^*(e) = 1, \forall e \in M^*$  car diminuer la valeur d'une arête de  $M^*$  diminue de 1 la valeur de tout couplage (dont  $M^*$ ) contenant cette arête. Dans ces circonstances, le problème consiste à déterminer un ensemble de cardinal minimum d'arêtes  $E^* \subseteq E \setminus M^*$  de sorte que  $M^*$  est un couplage maximum dans le graphe partiel  $G' = (V, E \setminus E^*)$  ( $w - w^*$  est le vecteur caractéristique de  $E^*$ ). Pour la suite de cette section, nous nous limitons à ce cas particulier.

**THÉORÈME 1.6.**– [ALFANDARI 04] *Le problème  $\text{INVCPM}_{\{0,1\}^m}$  ( $m$  est le nombre d'arêtes) pour lequel le couplage fixé ne contient que des arêtes de valeur 1 est polynomial dans les graphes bipartis.*

*Preuve.* soit une instance de  $\text{INVCPM}_{\{0,1\}^m}$  constituée d'un graphe biparti  $G = (L \cup R, E)$ ,  $|L \cup R| = m$ , et d'un couplage  $M^*$  sur  $G$ ; nous notons par  $V(M^*)$  les sommets insaturés par  $M^*$ .

Si  $L \cap V(M^*) = \emptyset$  ou  $R \cap V(M^*) = \emptyset$ , alors  $M^*$  est déjà un couplage maximum et  $E^* = \emptyset$ . Supposons donc  $L \cap V(M^*) \neq \emptyset$  ou  $R \cap V(M^*) \neq \emptyset$ ; nous construisons le graphe orienté  $\vec{G} = (V \cup \{s, t\}, \vec{E})$  de la manière suivante :  $\vec{e} = (x, y) \in \vec{E}$  si  $(x, y) \in M^*$ ,  $x \in R$  et  $y \in L$  ou  $(x, y) \in E \setminus M^*$ ,  $x \in L$  et  $y \in R$  ou  $x = s$  et  $y \in L \cap V(M^*)$  ou enfin  $x \in R \cap V(M^*)$  et  $y = t$ . La figure 1.1 illustre cette construction (les arêtes de  $M^*$  sont dessinées en trait continu, celles de  $E \setminus M^*$  en pointillés).



**Figure 1.1.** Exemple de la transformation de  $G$  en  $\vec{G}$

Nous vérifions aisément qu'à tout chemin de  $s$  à  $t$  dans  $\vec{G}$  correspond une unique chaîne alternée augmentante de  $G$  et réciproquement, qu'à toute chaîne alternée augmentante de  $G$ , il est possible d'associer un unique chemin de  $s$  à  $t$  dans  $\vec{G}$ .

Ainsi, en construisant le réseau  $(\vec{G}, c)$  avec  $c(\vec{e}) = 1$  si  $e \in E$  et  $c(\vec{e}) = |E| - |M^*| + 1$  si  $\vec{e} = (s, x)$  ou  $(y, t)$ , l'algorithme de *Ford et Fulkerson* permet de déterminer un flot maximum à composantes 0,1 et une *coupe de capacité minimum* séparant  $s$  de  $t$ . Une telle coupe consiste en un ensemble d'arcs de cardinal minimum coupant tous les chemins de  $s$  à  $t$ , c'est-à-dire à déterminer  $E^*$ , ensemble d'arêtes de cardinal minimum dans le graphe initial coupant toutes les chaînes alternées augmentantes par rapport au couplage  $M^*$ . ■

#### 1.4. Problèmes inverses difficiles

L'objectif de cette section est de mettre en évidence certains problèmes d'optimisation inverse difficiles. Nous commençons d'abord par illustrer à travers deux exemples (le problème de *stable de poids maximum* et le *voyageur de commerce maximum*) comment déduire un résultat de difficulté pour un problème inverse à partir d'un résultat de difficulté pour le problème initial. Même si une réciproque du théorème 1.5 ne semble pas évidente à montrer, il semble que de tels résultats peuvent être établis pour nombre de problèmes combinatoires **NP**-difficiles. Une autre question soulevée par le théorème 1.5 est de savoir s'il existe des problèmes naturels polynomiaux dont une version inverse (en variables continues et positives) serait difficile. L'existence d'un tel problème est longtemps restée ouverte et a été résolue dans [CAI 99] : le problème de *localisation d'un centre inverse* en variables continues est **NP**-difficile, alors même que le problème de localisation d'un centre est polynomial (et même trivial). C'est l'hypothèse de linéarité dans le théorème 1.5 qui n'est pas satisfaite dans ce cas.

Nous nous intéresserons ensuite à des problèmes d'optimisation inverse sous des hypothèses plus restrictives, notamment des problèmes inverses partiels et des problèmes inverses à variables entières ou binaires. Les problèmes inverses partiels ont fourni les premiers exemples de problèmes polynomiaux donnant lieu à un problème inverse difficile. Plus précisément, nous décrivons (voir [YANG 01]) un cas pour lequel le problème de *coupe de capacité minimum inverse* partiel est **NP**-difficile alors même que nous avons vu dans la section 1.3 plusieurs versions non partielles polynomiales. Le cadre des problèmes inverses en variables entières a été relativement peu étudié jusqu'à présent et fournirait sans doute de nombreux exemples de cas difficiles. Nous montrons notamment que le problème de *couplage de poids maximum inverse* en variables entières est difficile. Notons que nous avons proposé au paragraphe 1.3.4 un cas polynomial et que d'après le théorème 1.5, différentes versions en variables continues sont également polynomiales.

##### 1.4.1. Problèmes **NP**-difficiles inverses

Nous avons noté, après le théorème 1.5, qu'un problème admettant une version inverse polynomiale admet un test d'optimalité polynomial. Pour la plupart des problèmes **NP**-difficiles, on peut montrer qu'un tel test ne peut exister, si  $\mathbf{P} \neq \mathbf{NP}$ .

###### 1.4.1.1. L'exemple du *stable de poids maximum*

Le problème du *stable de poids maximum*, noté SPM, consiste, étant donné un graphe simple  $G = (V, E)$  et une fonction de poids  $w$  définie sur les sommets de  $V$ , à trouver un stable (sommets deux à deux non liés par une arête)  $S^*$  de  $G$  dont la somme  $w(S^*) = \sum_{v \in S^*} w(v)$  est maximum. Le *nombre de stabilité* de  $G$ , noté  $\alpha(G)$ , est la cardinalité maximum d'un stable ( $w(v) = 1$  pour tout sommet  $v \in V$ ). SPM est **NP**-difficile [GAREY 79]. On note  $n = |V|$  et  $m = |E|$ .

Le problème du *stable de poids maximum inverse*, noté INVSPM, consiste, étant donnée une instance  $I = (G, S^*, w)$ , à modifier les poids des sommets de sorte que :

- i) le stable  $S^*$  soit de poids maximum dans  $(G, w^*)$ ,
- ii) la quantité  $\sum_{v \in V} |w(v) - w^*(v)|$  soit minimum.

Dans le cas des variables binaires, on peut facilement montrer à l'aide d'arguments similaires à ceux exposés dans au paragraphe 1.3.4, que la restriction du problème  $\text{INVSPM}_{\{0,1\}^n}$  pour laquelle le stable  $S^*$  fixé ne contient que des sommets de valeur 1 est équivalente au problème consistant à retirer un nombre minimum de sommets pour rendre un stable fixé  $S^*$  optimal.

Il est facile de montrer qu'un test d'optimalité polynomial pour le problème de stable de cardinal maximum permettrait de déterminer le nombre de stabilité en temps polynomial. En effet, étant donné un graphe  $G = (V, E)$ , on construit un graphe  $G_k$  en ajoutant un stable  $S_k$  de taille  $k$  totalement lié aux sommets de  $V$ .  $S_k$  est optimal dans  $G_k$  si et seulement si le nombre de stabilité de  $G$  est inférieur ou égal à  $k$ . Il suffit donc, pour déterminer  $\alpha(G)$ , de considérer pour chaque  $k \in \{0, \dots, n\}$  l'instance  $G_k$  et de poser  $S^* = S_k$ .  $\alpha(G)$  est le plus grand  $k$  pour lequel ces instances de  $\text{INVSPM}_A$  ( $A \in \{\mathbb{N}^n, \mathbb{R}^{+n}, \{0, 1\}^n\}$ ), sont de valeur nulle.

Il est également intéressant de remarquer que le problème de stable maximum se réduit directement à sa version inverse par une réduction préservant la valeur optimale. Soit  $G = (V, E)$  un graphe simple, nous construisons une instance  $I = (G', S^*, w)$  de  $\text{INVSPM}_A$ ,  $A \in \{\mathbb{N}^n, \mathbb{R}^{+n}\}$  de la manière suivante :

- 1)  $G' = (V', E')$  où  $V' = V \cup \{s\}$  et  $E' = E \cup \{[v, s] | v \in V\}$ ;
- 2) La fonction de poids  $w$  est donnée par  $w(v) = 1$  pour tout  $v \in V$  et  $w(s) = 0$ ;
- 3)  $S^* = \{s\}$ .

Soit  $w'$  une solution réalisable pour le problème inverse. Si il existe  $v_0 \in V$ ,  $w'(v_0) > w(v_0)$ , alors la solution obtenue en diminuant le poids de  $v_0$  à  $w(v_0)$  reste réalisable et de valeur  $(\|w' - w\|_1)$  meilleure. On peut donc supposer  $w'(v) \leq w(v)$  pour tout sommet  $v \in V$ . Supposons qu'alors il existe  $v_0 \in V$ ,  $w'(v_0) = w(v_0) - \varepsilon$  avec  $\varepsilon > 0$ . Alors en posant  $w''(s) = w'(s) + \varepsilon$ ,  $w''(v_0) = w(v_0)$  et  $w''(v) = w'(v)$  sinon,  $w''$  reste réalisable (c'est-à-dire  $\{s\}$  reste un stable de valeur maximum) et  $\|w' - w\|_1 = \|w'' - w\|_1$ . En réitérant ce procédé, on montre que pour toute solution réalisable  $w'$  de  $\text{INVSPM}_A$  on peut construire en temps polynomial une solution réalisable  $\tilde{w}$  au moins aussi bonne telle que  $\tilde{w}(s) \geq w(s)$  et  $\tilde{w}(v) = w(v)$  pour  $v \in V$ . Une telle solution est réalisable pour  $\text{INVSPM}_A$  si et seulement si  $\tilde{w}(s) \geq \alpha(G)$ . Une solution optimale  $w^*$  de  $\text{INVSPM}_A$  est donc de valeur  $\alpha(G)$ . On a donc la proposition 1.1 :

**PROPOSITION 1.1.**– [ALFANDARI 04] *Aucun des problèmes  $\text{INVSPM}_A$  où  $A \in \{\mathbb{N}^n, \mathbb{R}^{+n}, \{0, 1\}^n\}$  n'est résoluble en temps polynomial à moins que  $P=NP$ .*

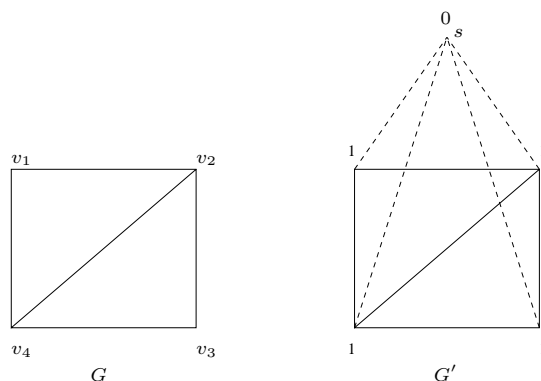


Figure 1.2. Exemple de la transformation de  $G$  en  $I = (G', \{s\}, w)$

#### 1.4.1.2. Une remarque sur le problème de voyageur de commerce maximum inverse

Le problème de voyageur de commerce TSPMAX consiste, dans un graphe complet arêtes-pondéré, à déterminer un cycle hamiltonien (passant une fois et une seule en chaque sommet) de valeur maximum. Ce problème est **NP**-difficile. Il ne semble pas si aisé que dans le cas précédent de le réduire à une version inverse. Toutefois, nous montrons que, si  $\mathbf{P} \neq \mathbf{NP}$ , TSPMAX n'admet pas de test d'optimalité polynomial et donc le problème inverse est non polynomial. Le problème de voyageur de commerce maximum inverse, noté INVTSPMAX, consiste étant donné  $(K_n, d)$  une instance de TSPMAX et un cycle hamiltonien  $C^*$ , à modifier aussi peu que possible les distances de sorte que  $C^*$  devienne un cycle hamiltonien de poids maximum dans le nouveau graphe.

Le problème suivant est prouvé **NP**-complet dans [PAPADIMITRIOU 94] : étant donné un graphe  $G = (V, E)$  et un cycle hamiltonien  $C$ , existe-t-il un second cycle hamiltonien ? Considérons un graphe instance de ce problème, affectons la valeur 1 à ses arêtes et complétons-le par des arêtes de valeur nulle en un graphe complet. Pour chaque arête  $e$  du cycle hamiltonien connu  $C$ , on pose à 0 la valeur de  $e$  et on considère le graphe  $G_e$  résultant muni du cycle  $C$  comme une instance de  $\text{INVTSPMAX}_A$ , pour  $A \in \{\mathbb{N}^m, \mathbb{R}^{+m}, \{0, 1\}^m\}$  ou  $m = |E|$ . On vérifie immédiatement que  $C$  est un cycle de valeur maximum dans  $G_e$  si et seulement si tous les cycles hamiltoniens contiennent  $e$ . Il existe donc un second cycle hamiltonien dans  $G$  si et seulement si pour au moins une arête  $e$  de  $C$ , le problème inverse associé est de valeur non nulle. On a donc la proposition 1.2 :

PROPOSITION 1.2. – [ALFANDARI 04] Si  $\mathbf{P} \neq \mathbf{NP}$ , le problème  $\text{INVTSPMAX}_A$ , pour  $A \in \{\mathbb{N}^m, \mathbb{R}^{+m}, \{0, 1\}^m\}$  n'admet pas d'algorithme polynomial.

### 1.4.2. Problème de localisation de centre

Etant donné un graphe orienté  $\vec{G} = (V, \vec{E})$  et une fonction distance  $d$  de  $\vec{E}$  à valeur dans  $\mathbb{R}^+$ , le problème de *localisation de centre*, noté 1-LC, consiste à trouver un sommet  $s \in V$  (appelé *centre du réseau*) de manière à minimiser la quantité  $val_d(s) = \max_{v \in V \setminus \{s\}} d(P_{s,v}^*)$  où  $d(P_{s,v}^*)$  désigne la valeur d'un plus court chemin de  $s$  à  $v$  (lorsqu'un tel chemin n'existe pas, on pose  $d(P_{s,v}^*) = +\infty$ ). Ce problème est polynomial car il suffit d'essayer chaque sommet et de calculer sa valeur (rappelons que le problème de cheminement de valeur minimum est polynomial) ; il le reste lorsque l'on cherche à placer un nombre fixé de centres (voir par exemple [CHRISTOFIDES 75]). En revanche, le problème devient **NP**-difficile si le nombre de centres fait partie de l'instance [GAREY 79].

Le problème de *localisation de centre inverse*, noté INV1-LC, consiste, étant donné un réseau  $(\vec{G}, d)$  et un sommet  $s \in V$ , à trouver une fonction distance  $d^*$  de  $\vec{E}$  à valeur dans  $\mathbb{R}^+$  telle que :

- i) le sommet  $s$  devient le centre optimal du réseau  $(\vec{G}, d^*)$ ,
- ii) la quantité  $\sum_{\vec{e} \in \vec{E}} |d(\vec{e}) - d^*(\vec{e})|$  est minimum.

Dans [CAI 99], INV1-LC $_{\mathbb{R}+m'}$  où  $m' = |E|$  est montré **NP**-difficile. La preuve établit une réduction polynomiale à partir du problème de satisfiabilité, noté SAT. Une instance est définie par un ensemble de  $n$  variables booléennes  $X = \{x_1, \dots, x_n\}$  et un ensemble de  $m$  clauses  $\mathcal{C} = \{C_1, \dots, C_m\}$  impliquant les variables de  $X$ . La *négation d'une variable*  $x_i$  est notée  $\bar{x}_i$  : elle vaut vrai lorsque  $x_i$  est faux et faux lorsque  $x_i$  est vrai ; une variable booléenne ou sa négation est appelée *littéral*. Une *clause* est une disjonction de littéraux et est *satisfaite par une affectation de valeurs de vérité* si au moins l'un de ses littéraux est vrai.

L'objectif de SAT consiste, étant donnée une instance  $(\mathcal{C}, X)$ , à savoir s'il existe une *fonction de vérité*  $f$ , c'est-à-dire une fonction de  $X$  dans  $\{\text{vrai}, \text{faux}\}$ , qui satisfait toutes les clauses de  $\mathcal{C}$ . Notons que ce problème fut le premier problème à avoir été montré **NP**-complet.

**THÉORÈME 1.7.**— [CAI 99] *Le problème INV1-LC $_{\mathbb{R}+m'}$  où  $m' = |\vec{E}|$  en variables continues est **NP**-difficile.*

*Preuve.* soit  $I = (\mathcal{C}, X)$  avec  $X = \{x_1, \dots, x_n\}$  et  $\mathcal{C} = \{C_1, \dots, C_m\}$  une instance de SAT. Nous construisons une instance  $I' = (\vec{G}, s, d)$  de INV1-LC $_{\mathbb{R}+m'}$  de la manière suivante : le graphe  $\vec{G} = (V, \vec{E})$ , où  $V = \{v_i, x_i, \bar{x}_i | i = 1, \dots, n\} \cup \{v_{n+1}, v_{n+2}, v_{n+3}\} \cup \{C_j | j = 1, \dots, m\}$ , se décompose en trois sous-graphes  $\vec{G}_1, \vec{G}_2$  et  $\vec{G}_3$ .

Le graphe  $\vec{G}_1$ , illustré par la figure 1.3, contient les liaisons entre les sommets  $v_i$  et les sommets  $x_i$  et  $\bar{x}_i$ . Les arcs dessinés en pointillés sont de distance nulle tandis que ceux dessinés en traits continus sont de distance 1.

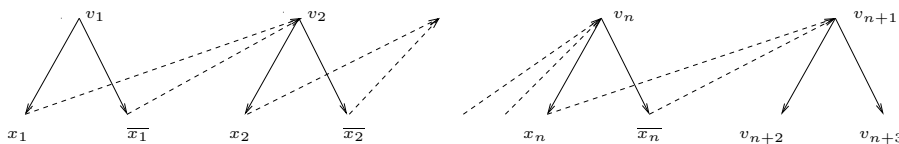


Figure 1.3. Le sous-graphe  $\vec{G}_1$

Le graphe  $\vec{G}_2$  est le graphe biparti caractéristique de l'instance  $I = (\mathcal{C}, X)$  dont les sommets représentent les littéraux  $x_i$  et  $\bar{x}_i$  pour  $i = 1, \dots, n$ , ainsi que les clauses  $C_j$  pour  $j = 1, \dots, m$ ; il existe alors un arc  $(l_i, C_j)$  si et seulement si le littéral  $l_i$  appartient à la clause  $C_j$ . Un exemple est illustré par la figure 1.4.

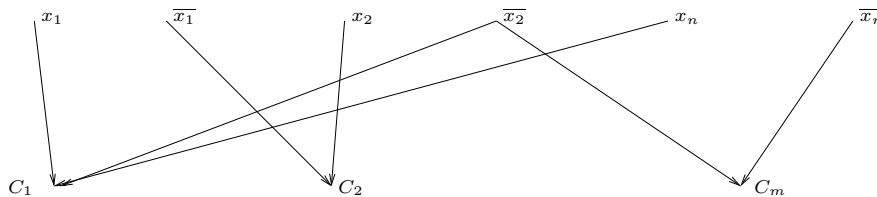


Figure 1.4. Exemple de sous-graphe  $\vec{G}_2$  contenant les clauses  $C_1 = x_1 \vee \bar{x}_2 \vee x_n$ ,  $C_2 = \bar{x}_1 \vee x_2$  et  $C_m = \bar{x}_2 \vee \bar{x}_n$

Le graphe  $\vec{G}_3$  contient les arcs d'extrémité initiale  $v_{n+2}$  ou  $v_{n+3}$  : ces deux sommets sont reliés à tous les autres sommets de  $V$  par des arcs partant de  $v_{n+2}$  ou  $v_{n+3}$  (en particulier, les arcs  $(v_{n+2}, v_{n+3})$  et  $(v_{n+3}, v_{n+2})$  existent dans  $\vec{G}$ ).

Le sommet  $v_1$  est centre (c'est-à-dire  $s = v_1$ ).

La fonction distance  $d$  vaut  $d(\vec{e}) = 0$  si  $\vec{e} = (x_i, v_{i+1})$  ou  $\vec{e} = (\bar{x}_i, v_{i+1})$  pour  $i = 1, \dots, n$ , vaut  $d(\vec{e}) = 1$  sinon.

Toute fonction distance  $d'$  doit vérifier  $d'(\vec{e}) \in \mathbb{R}^+$  pour tout arc  $\vec{e}$ .

Cette transformation s'exécute en temps polynomial et nous affirmons que  $I$  est satisfiable si et seulement si l'instance  $I'$  admet une fonction distance  $d^*$  de  $\vec{E}$  vers  $\mathbb{R}^+$  telle que  $v_1$  est centre de  $(\vec{G}, d^*)$  et telle que la somme des modifications des distances vaut au plus  $n$ , c'est-à-dire  $\sum_{\vec{e} \in \vec{E}} |d(\vec{e}) - d^*(\vec{e})| \leq n$ . Tout d'abord, remarquons que les sommets  $v_{n+2}$  et  $v_{n+3}$  sont centres optimaux du réseau  $(\vec{G}, d)$ , avec  $val_d(v_{n+2}) = val_d(v_{n+3}) = 1$  (voir le réseau  $(\vec{G}_3, d)$ ).

Supposons que  $I$  est satisfiable et soit  $f$  une fonction de vérité satisfaisant toutes les clauses de  $\mathcal{C}$  ; nous modifions la fonction de poids  $d$  de la façon suivante :

- 1)  $\forall i \in \{1, \dots, n\}$ ,  $d^*(v_i, z_i) = 0$  où  $z_i = x_i$  si  $f(x_i) = \text{vrai}$  et  $z_i = \bar{x}_i$  si  $f(x_i) = \text{faux}$ .
- 2)  $d^*(\vec{e}) = d(\vec{e})$  pour tous les autres arcs  $\vec{e}$ .

Bien entendu, les contraintes imposées sur la fonction  $d^*$  sont vérifiées et l'on a  $val_{d^*}(v_1) = 1$ . Considérons en effet les chemins  $P_{v_1, v_i}^*$  de  $v_1$  à  $v_i$  (resp.,  $P_{v_1, z_i}^*$  de  $v_1$  à  $z_i$ ) dans le réseau  $(\vec{G}_1, d^*)$  pour  $i = 2, \dots, n+1$ , décrits par la séquence  $P_{v_1, v_i}^* = \{(v_1, z_1), (z_1, v_2), \dots, (v_{i-1}, z_i), (z_i, v_i)\}$  (resp.,  $P_{v_1, z_i}^* = P_{v_1, v_i}^* \setminus \{(z_i, v_i)\}$ ) : on a  $d^*(P_{v_1, v_i}^*) = 0$  et  $d^*(P_{v_1, z_i}^*) = 0$ . De plus, par construction, il existe pour toute clause  $C_j$  un littéral  $x_{i_j}$  (ou  $\bar{x}_{i_j}$ ) de cette clause tel que  $f(x_{i_j}) = \text{vrai}$  (ou  $f(x_{i_j}) = \text{faux}$ ), puisque  $f$  satisfait toutes les clauses ; dans le réseau  $(\vec{G}_2, d^*)$ , cela se traduit par  $d^*(z_{i_j}, C_j) = d(z_{i_j}, C_j) = 1$  et l'on déduit :  $d^*(P_{v_1, C_j}^*) = d^*(P_{v_1, z_{i_j}}^*) + d^*(z_{i_j}, C_j) = 1$ . Finalement,  $d^*(P_{v_1, y}^*) = d^*(P_{v_1, v_{n+1}}^*) + d^*(v_{n+1}, y) = 1$  pour  $y = v_{n+2}$  ou  $y = v_{n+3}$ . Ainsi, puisque  $\forall v \in V$ ,  $val_{d^*}(v) \geq 1$ ,  $v_1$  est centre optimal du réseau  $(\vec{G}, d^*)$  et la somme des modifications vaut  $\sum_{\vec{e} \in \vec{E}} |d(\vec{e}) - d^*(\vec{e})| = n$ .

Réciproquement, soit  $d^*$  une modification de la fonction distance à valeurs dans  $\mathbb{R}^+$  telle que le sommet  $v_1$  est centre optimal du réseau  $(\vec{G}, d^*)$  et qui vérifie  $\sum_{\vec{e} \in \vec{E}} |d(\vec{e}) - d^*(\vec{e})| \leq n$ . Montrons qu'alors on peut construire une fonction de vérité  $f$  satisfaisant toutes les clauses de  $\mathcal{C}$ . Remarquons d'abord que tous les chemins élémentaires de  $v_1$  à  $v_{n+2}$  (ou  $v_{n+3}$ ) appartiennent à  $\vec{G}_1$  et que ce sont tous des plus courts chemins de valeur  $d(P_{v_1, v_{n+2}}^*) = n+1$  ; ainsi, puisque la somme des modifications vaut au plus  $n$ , on déduit que :  $val_{d^*}(v_1) \geq 1$ . De plus, on doit au moins diminuer la valeur d'un de ces chemins d'une quantité  $p_1$ . Par construction du graphe  $\vec{G}_1$ , cela reviendra à réduire la distance de certains arcs  $(v_i, x_i)$  ou  $(v_i, \bar{x}_i)$  (les autres arcs ont une distance 0) ou de l'arc  $(v_{n+1}, v_{n+2})$  (nous montrons plus loin que la distance de cet arc ne doit pas être modifiée). Puisque  $d(P_{v_1, v_{n+2}}^*) = n+1$ , la somme  $p_1$  de ces pertes vérifiera :  $p_1 \geq n+1 - val_{d^*}(v_1)$ .

De même, puisque dans le réseau  $(\vec{G}, d)$ ,  $val_d(v_{n+2}) = 1$  (voir le graphe  $\vec{G}_3$ ), les modifications permettent d'ajouter au plus  $n$  à cette valeur dans  $(\vec{G}, d^*)$ . En d'autres termes  $val_{d^*}(v_{n+2}) \leq n+1$ . Le sommet  $v_1$  étant centre optimal pour  $d^*$ , nous concluons que :  $val_{d^*}(v_1) \leq val_{d^*}(v_{n+2}) \leq n+1$ . Ainsi, il faudra au moins augmenter la valeur d'un des chemins  $P_{v_{n+2}, y}^*$  vérifiant  $d(P_{v_{n+2}, y}^*) = val_d(v_{n+2})$  d'une quantité  $p_2$ . Par construction du graphe  $\vec{G}_3$ , chacun de ces chemins est réduit à un arc de la forme  $(v_{n+2}, v)$  pour  $v \in V \setminus \{v_{n+2}\}$ . Comme  $val_d(v_{n+2}) = 1$ , la quantité  $p_2$  ajoutée à la distance de cet arc vérifiera :  $p_2 \geq val_{d^*}(v_1) - 1$ .

Le sommet  $v_{n+3}$  étant lui aussi centre optimal du réseau  $(\vec{G}, d)$  et les chemins  $P_{v_{n+3}, v}^*$  qui vérifient  $d(P_{v_{n+3}, v}^*) = val_d(v_{n+3})$  étant disjoints des chemins optimaux

issus de  $v_{n+2}$ , le même raisonnement que précédemment s'applique. Ainsi, on doit au moins ajouter à un des arcs  $(v_{n+3}, v)$  une quantité  $p_3 : p_3 \geq \text{val}_{d^*}(v_1) - 1$ .

Naturellement,  $p_1 + p_2 + p_3 \leq \sum_{\vec{e} \in \vec{E}} |d(\vec{e}) - d^*(\vec{e})| \leq n$ ; la somme des trois dernières inégalités fournit :  $\text{val}_{d^*}(v_1) \leq 1$ . On en déduit  $\text{val}_{d^*}(v_1) = 1$ , de plus  $p_1 = n, p_2 = 0$  et  $p_3 = 0$ .

Montrons maintenant que  $d^*(v_{n+1}, v_{n+2}) = d(v_{n+1}, v_{n+2})$  et  $d^*(v_{n+1}, v_{n+3}) = d(v_{n+1}, v_{n+3})$ . Supposons le contraire et soit  $\varepsilon_i \geq 0$  pour  $i = 1, 2$ , la réduction de la distance de l'arc  $(v_{n+1}, v_{n+1+i})$ , c'est-à-dire  $d^*(v_{n+1}, v_{n+1+i}) = d(v_{n+1}, v_{n+1+i}) - \varepsilon_i$ . Sans perte de généralité, supposons  $\varepsilon_1 > 0$ . Puisque la somme des modifications  $p_1$  dans  $\vec{G}_1$  vaut  $n$ , cela indiquera que nous pourrions diminuer au maximum de  $(n - \varepsilon)$  avec  $\varepsilon = \varepsilon_1 + \varepsilon_2$ , la distance d'un plus court chemin de  $v_1$  à  $v_{n+1}$ . Autrement dit  $d^*(P_{v_1, v_{n+1}}^*) \geq d(P_{v_1, v_{n+3}}^*) - (n - \varepsilon) = \varepsilon$ . La répercussion de la diminution de cette distance sur le chemin  $P_{v_1, v_{n+3}}^*$  donnera  $d^*(P_{v_1, v_{n+3}}^*) \geq d(P_{v_1, v_{n+1}}^*) + d^*(v_{n+1}, v_{n+3}) = \varepsilon + 1 - \varepsilon_2 = 1 + \varepsilon_1$ , ce qui contredirait  $\text{val}_{d^*}(v_1) = 1$ . Ainsi, seules les distances des arcs de la forme  $(v_i, x_i)$  ou  $(v_i, \bar{x}_i)$  seront modifiées. Puisque leur distance vaut 1, qu'ils sont au nombre de  $n$  sur le chemin  $P_{v_1, v_{n+2}}^*$ , et que  $p_1 = n$ , on a pour tout  $i = 1, \dots, n$  soit  $d^*(v_i, x_i) = 0$  soit  $d^*(v_i, \bar{x}_i) = 0$ . Construisons la fonction de vérité  $f$  suivante :  $f(x_i) = \text{vrai}$  si  $d^*(v_i, x_i) = 0$  et  $f(x_i) = \text{faux}$  si  $d^*(v_i, \bar{x}_i) = 0$ . Par construction du graphe  $\vec{G}_2$  et en se servant de l'égalité  $\text{val}_{d^*}(v_1) = 1$ , on vérifie aisément que  $f$  satisfait toutes les clauses de  $\mathcal{C}$ . ■

### 1.4.3. Un problème inverse partiel : la coupe de capacité minimum

Nous avons vu au paragraphe 1.3.1.2 que le problème de la coupe de capacité minimum inverse en variables continues ou entières et positives est polynomial. En revanche, le problème partiel PINVCOUPEMIN en variables bornées est **NP**-difficile.

**THÉORÈME 1.8.**— [YANG 01] *Le problème partiel PINVCOUPEMIN<sub>[a\_i; b\_i]^m</sub> en variables bornées est NP-difficile.*

*Preuve.* la réduction polynomiale se fait à partir d'une restriction du problème de la bipartition appelée *bipartition paire* et notée BIPARTPAIRE où les éléments  $a_i$  sont en nombre pair. Il s'agit de décider s'il existe  $S \subset \{1, \dots, 2n\}$  tel que  $\sum_{i \in S} a_i = \sum_{i \in \{1, \dots, 2n\} \setminus S} a_i$ . Ce problème est **NP**-complet. En effet, le cas où les  $a_i$  sont en nombre impair se réduit polynomialement au cas où ils sont en nombre pair. Le principe de cette réduction est de transformer une instance  $I$  à  $2n + 1$  éléments en  $2n$  instances  $I_1, \dots, I_{2n}$  à  $2n$  éléments chacune où  $I_j$  est obtenue à partir de  $I$  en remplaçant les éléments  $a_1$  et  $a_{j+1}$  en l'élément  $a'_j = a_1 + a_{j+1}$ . Il est alors aisé de constater que la réponse à  $I$  est oui si et seulement si la réponse à au moins l'une des instances  $I_j$  est oui, puisque l'on peut supposer sans restriction que  $2 \leq |S| \leq 2(n - 1)$ .

Soit donc  $\{a_1, \dots, a_{2n}\}$  avec  $\sum_{i=1}^{2n} a_i = 2B$  une instance de BIPARTPAIR. On construit l'instance  $I = (\vec{H}, U, c)$  de PINVCOUPEMIN $_{[a_i; b_i]}$  de la manière suivante :

1) Le graphe  $\vec{G} = (V, \vec{E})$  a  $3n + 2$  sommets avec  $V = \{s, t\} \cup U \cup L \cup R$  où  $U = \{u_1, \dots, u_n\}$ ,  $L = \{l_1, \dots, l_n\}$ ,  $R = \{r_1, \dots, r_n\}$  et  $\vec{E} = \{(s, u_i), (u_i, l_i), (l_i, t), (u_i, r_i), (r_i, t) | i = 1, \dots, n\}$ ;

2) La fonction de capacité  $c$  est donnée par  $c(s, u_i) = 2B - \frac{1}{2n}B$ ,  $c(u_i, l_i) = c(u_i, r_i) = B$ ,  $c(l_i, t) = B - \frac{1}{2}a_{2i-1}$  et  $c(r_i, t) = B - \frac{1}{2}a_{2i}$  pour  $i = 1, \dots, n$ ;

3) On ne s'intéresse, dans la logique du problème inverse partiel, qu'aux coupes séparant  $s$  et  $t$  et contenant l'ensemble  $U$ ;

4) Toute fonction capacité  $c'$  doit vérifier  $c'(s, u_i) = 2B - \frac{1}{2n}B$  (l'intervalle est donc  $[2B - \frac{1}{2n}B; 2B - \frac{1}{2n}B]$ ),  $c'(u_i, l_i) = c'(u_i, r_i) = B$ ,  $c'(l_i, t) \in [B - \frac{1}{2}a_{2i-1}; B + \frac{1}{2}a_{2i-1}]$  et  $c'(r_i, t) \in [B - \frac{1}{2}a_{2i}; B + \frac{1}{2}a_{2i}]$ .

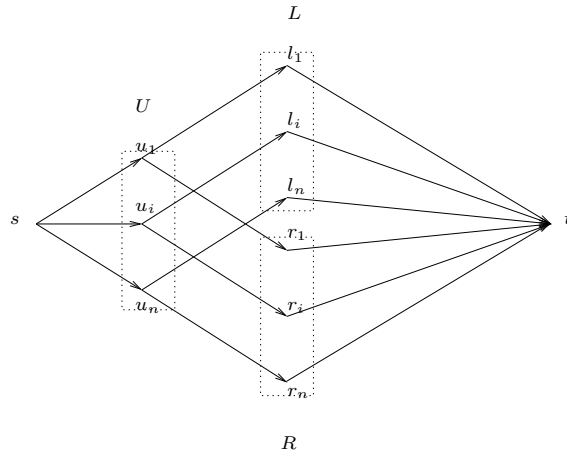


Figure 1.5. Exemple de la transformation de  $\{a_1, \dots, a_{2n}\}$  en  $I = (\vec{G}, U, c)$

La figure 1.5 illustre cette transformation. Nous montrons qu'il existe un ensemble  $S^* \subset \{1, \dots, 2n\}$  avec  $\sum_{i \in S^*} a_i = B$  si et seulement s'il existe une coupe  $V^*$  avec  $\{s\} \cup U \subseteq V^*$ ,  $t \notin V^*$  et une fonction capacité optimale  $c^*$  satisfaisant les contraintes d'intégrité telles que :  $V^*$  est une coupe de capacité minimum dans le réseau  $(\vec{G}, c^*)$  et la somme des modifications est au plus  $B$ .

Soit  $c^*$  une fonction capacité optimale ; remarquons que toute coupe  $V'$  contenant  $s$  et  $U$  peut s'écrire  $V' = \{s\} \cup U \cup L_0 \cup R_0$  avec  $L_0 \subseteq L$  et  $R_0 \subseteq R$ . Si  $l_i \in L_0$  (resp.,  $l_i \notin V'$ ), alors  $(l_i, t) \in \Omega(V')$  (resp.,  $(u_i, l_i) \in \Omega(V')$ ); nous rappelons que  $\Omega(V')$  désigne le cocycle sortant de  $V'$ , c'est-à-dire l'ensemble des arcs d'extrémité initiale dans  $V'$  et d'extrémité terminale dans  $V \setminus V'$ ; de même, si  $r_i \in R_0$  (resp.,  $r_i \notin V'$ ),

alors  $(r_i, t) \in \Omega(V')$  (resp.,  $(u_i, r_i) \in \Omega(V')$ ) : en fait, on a «  $(u_i, l_i) \in \Omega(V') \Leftrightarrow (l_i, t) \notin \Omega(V')$  » et «  $(u_i, r_i) \in \Omega(V') \Leftrightarrow (r_i, t) \notin \Omega(V')$  » et ainsi,  $|\Omega(V')| = 2n$ . De plus, en notant  $S = \{2i-1 | l_i \in L_0\} \cup \{2i | r_i \in R_0\}$ ,  $p_{2i-1} = c(l_i, t) - c^*(l_i, t)$  et  $p_{2i} = c(r_i, t) - c^*(r_i, t)$  pour  $i = 1, \dots, n$ , la capacité de la coupe  $V'$  vaut  $c^*(V') = 2nB + \frac{1}{2} \sum_{i \in S} a_i - \frac{1}{2} \sum_{i \in S} p_i$ .

Soit  $V^*$  une coupe optimale du réseau  $(\vec{G}, c^*)$  contenant  $s$  et  $U$  ; on a  $c^*(V^*) \leq c^*(\{s\}) = 2nB + \frac{1}{2}B$  et on en déduit :  $\sum_{i \in S^*} p_i \geq \frac{1}{2}B + \frac{1}{2} \sum_{i \in S^*} a_i$ . Les contraintes sur les capacités imposent  $p_i \leq a_i$  et donc,  $\sum_{i \in S^*} p_i \leq \sum_{i \in S^*} a_i$ . Des deux dernières inégalités, on déduit  $\sum_{i \in S^*} a_i \geq B$ . Par ailleurs, comme  $\sum_{i \in S^*} p_i \leq \sum_{\vec{e} \in \vec{E}} |c(\vec{e}) - c^*(\vec{e})| \leq B$  et d'après la première inégalité, on a :  $\sum_{i \in S^*} a_i \leq B$ .  $S^*$  est donc une bipartition paire.

Réciproquement, soit  $S^*$  une bipartition de  $\{a_1, \dots, a_{2n}\}$ . Modifions les capacités par  $c^*(l_i, t) = B - \frac{1}{2}a_{2i-1}$  si  $2i-1 \in S^*$ ,  $c^*(r_i, t) = B - \frac{1}{2}a_{2i}$  si  $2i \in S^*$  et  $c^*(\vec{e}) = c(\vec{e})$  pour tous les autres arcs. La fonction capacité  $c^*$  vérifie les contraintes de bornes et de plus  $\sum_{\vec{e} \in \vec{E}} |c(\vec{e}) - c^*(\vec{e})| = B$ . Posons  $V^* = \{s\} \cup U \cup \{l_i | 2i-1 \in S^*\} \cup \{r_i | 2i \in S^*\}$ . On vérifie aisément que  $V^*$  est une coupe optimale dans le réseau  $(\vec{G}, c^*)$ . ■

#### 1.4.4. Problème du couplage de poids maximum

Nous reprenons le problème du couplage introduit au paragraphe 1.3.4. Dans un graphe  $G = (V, E)$  à  $n$  sommets, un couplage  $M^*$  est dit *parfait* s'il est de taille  $|M^*| = \frac{n}{2}$  ; notons qu'un couplage parfait est nécessairement maximum. Lorsque le graphe  $G$  est muni d'une fonction poids  $w$  sur ses arêtes (on parle en ce cas de réseau  $(G, w)$ ), le poids d'un couplage  $M$  est donné par  $w(M) = \sum_{e \in M} w(e)$ . Le problème qui nous intéresse alors est de déterminer un couplage de poids maximum, noté CPM. Lorsque tous les poids valent 1, il s'agit du problème du couplage maximum. On s'intéresse parfois à rechercher un *couplage parfait de poids maximum* (qui ne coïncide pas nécessairement avec un couplage de poids maximum) parmi les couplages parfaits du graphe ; cette restriction sera notée CPPM. Notons que lorsque le graphe est biparti, CPPM est équivalent au *problème d'affection*. La détermination d'un couplage (parfait ou non) de poids maximum est de complexité polynomiale et peut se faire en  $O(n^3)$  (voir [GONDRAN 98, PAPADIMITRIOU 98]). Comme pour le cas non pondéré (voir le paragraphe 1.3.4), il existe plusieurs caractérisations des couplages de poids maximum. La plus célèbre, due à Berge, généralise au cas pondéré le concept de *chaînes ou cycles alternés*<sup>2</sup> et s'énonce comme suit : un couplage  $M^*$  est de poids

2. Une chaîne (resp., un cycle) alternée relativement à un couplage  $M$  est une chaîne (resp., un cycle) alternant arêtes de  $M$  et de  $E \setminus M$  ; dans le cas d'une chaîne, chacune de ses extrémités doit être soit une arête de  $M$ , soit une arête de  $E \setminus M$  incidente à un sommet insaturé par  $M$ .

maximum si et seulement si pour toute chaîne ou cycle alterné  $E'$ , le coût réduit  $\bar{w}(E') = w(M^* \cap E') - w(E' \setminus M^*)$  est positif ou nul (voir [BERGE 73]). Une chaîne (resp., cycle) alternée  $E'$  dont le coût réduit est strictement négatif est appelée chaîne (resp., cycle) *augmentante* (c'est-à-dire que le couplage  $M' = (M^* \setminus E') \cup (E' \setminus M^*)$  à un poids strictement plus grand que  $M^*$ ).

Le problème *du couplage (resp., parfait) de poids maximum inverse*, noté INVCPM (resp., INVCPPM), consiste donc, étant donné un réseau  $(G, w)$  et un couplage (resp., parfait)  $M^*$ , à trouver une fonction poids  $w^*$  telle que :

i) le couplage  $M^*$  est de poids maximum (resp., parmi les couplages parfaits) dans  $(G, w^*)$  ;

ii) la quantité  $\sum_{e \in E} |w(e) - w^*(e)|$  est minimum (relativement à l'ensemble des fonctions de poids pour lesquelles  $M^*$  est (resp., parfait) de poids maximum).

Le lemme 1.1 est prouvé dans [ALFANDARI 04] :

LEMME 1.1.– [ALFANDARI 04] *Pour les problèmes INVCPM<sub>A</sub> et INVCPPM<sub>A</sub> où  $A \in \{\mathbb{N}^m; \mathbb{R}^{+m}\}$  avec  $m = |E|$ , on peut réduire en temps polynomial toute instance à une instance satisfaisant les propriétés suivantes :*

i) *le couplage  $M^*$  est maximal dans  $G$  ;*

ii) *chaque arête  $e \in E$  appartient à une chaîne ou cycle augmentant relativement à  $M^*$  dans  $(G, w)$  ;*

iii) *toute fonction poids  $w^*$  optimale vérifie :  $\forall e \in M^*, w^*(e) \geq w(e)$  et  $\forall e \in E \setminus M^*, w^*(e) \leq w(e)$ . De plus, Il existe une fonction poids  $w^*$  optimale telle que :  $\forall e \in E \setminus M^*, w^*(e) = w(e)$ .*

*Preuve.* nous démontrons uniquement ces résultats pour INVCPM.

Pour i) : supposons le contraire et soit  $e_i$  une arête telle que  $M^* \cup \{e_i\}$  reste un couplage. Dans ce cas, on a nécessairement  $w^*(e_i) = 0$  et  $e_i$  peut être supprimé de  $G$ .

Pour ii) : soit une arête  $e = [v_i, v_j] \in E$  n'appartenant à aucune chaîne ou cycle augmentant. On prouve facilement que pour toute fonction poids optimale  $w^*$  on a  $w^*(e) = w(e)$ . Aussi, si  $e \in M^*$  alors on peut remplacer  $G$  par le sous-graphe engendré par  $V \setminus \{v_i, v_j\}$  et  $M^*$  par  $M^* \setminus \{e\}$ , tandis que si  $e \notin M^*$ , on peut supprimer  $e$  de  $G$ .

Pour iii) : la première partie du résultat s'obtient sans peine. Supposons la propriété (i) vérifiée et soit  $w^*$  une fonction de poids optimale pour laquelle il existe  $e' \notin M^*$  telle que  $w^*(e') = w(e') - \varepsilon$  ; puisque  $M^*$  est maximal, il existe  $e^* \in M^*$  adjacent à  $e'$ . En posant  $w'(e) = w^*(e)$  pour  $e \notin \{e', e^*\}$ ,  $w'(e') = w(e)$  et  $w'(e^*) = w^*(e^*) + \varepsilon$ , on obtient une nouvelle fonction de poids optimale. Soit en effet  $M$  un

couplage maximal quelconque : si  $e^* \in M$  (alors  $e' \notin M$ ),  $w'(M) = w^*(M) + \varepsilon \leq w^*(M^*) + \varepsilon = w'(M^*)$ ; sinon,  $w'(M) \leq w^*(M) \leq w^*(M^*) + \varepsilon = w'(M^*)$ .

Dans chacun des cas *i*), *ii*) et *iii*), en répétant tant que possible ces opérations, nous obtenons le résultat souhaité. ■

REMARQUE 10.– Notons que la résolution de  $\text{INVCPM}_A$  où  $A \in \{\mathbb{N}^m; \mathbb{R}^{+m}\}$  permet celle du problème  $\text{INVCPM}_A$  lorsque le couplage  $M^*$  est parfait. En d'autres termes, il existe une réduction polynomiale de  $\text{INVCPM}_A$  vers  $\text{INVCPM}_A$ . Pour le montrer, partons d'une instance  $I = (G, M^*, w)$  de  $\text{INVCPM}_A$  avec  $M^*$  couplage parfait de  $G = (V, E)$ ; construisons alors l'instance  $I' = (G', M^*, w')$  de  $\text{INVCPM}$  où  $G' = (V, E')$  est le graphe complet issu de  $G$  et la fonction  $w'$  est définie par :  $w'(e) = w(e)$  si  $e \in E$ ,  $w'(e) = 0$  sinon. On remarque qu'une fonction de poids  $w^*$  vérifiant le lemme 1.1 est optimale pour  $\text{INVCPM}_A$  si et seulement si sa restriction  $w^*$  à  $G$  est optimale pour  $\text{INVCPM}_A$ . En d'autres termes,  $M^*$  est un couplage de poids maximum dans  $(G, w^*)$  si et seulement si  $M^*$  est un couplage de poids maximum parmi les couplages parfaits dans  $(G', w^*)$ . Ainsi, puisque  $\sum_{e \in E} |w(e) - w^*(e)| = \sum_{e \in E'} |w'(e) - w^*(e)|$ , le résultat est démontré.

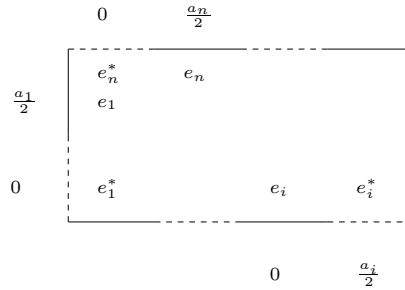
Nous montrons que  $\text{INVCPM}$  en variables bivaluées est **NP**-difficile dans les graphes bipartis alors que, pour ces graphes, la version la plus générale  $\text{INVCPM}_{\mathbb{R}^{+m}}$  (voir [AHUJA 01, HUANG 99, ZHANG 96a]) et une restriction en variables  $\{0, 1\}$  (voir le paragraphe 1.3.4) sont polynomiales.

THÉORÈME 1.9.– [ALFANDARI 04] *Le problème  $\text{INVCPM}_{\{a_i, b_i\}^m}$  est **NP**-difficile, même dans les graphes bipartis.*

*Preuve.* la réduction polynomiale s'effectue à partir du *problème de la bipartition*, connu comme étant **NP**-complet [GAREY 79]; ce problème, noté **BIPART**, consiste à décider si, étant donnés  $n$  entiers  $a_1, \dots, a_n$  dont la somme vaut  $\sum_{i=1}^n a_i = 2B$ , il existe  $S \subset \{1, \dots, n\}$  tel que  $\sum_{i \in S} a_i = B$ . Soit donc,  $\{a_1, \dots, a_n\}$  avec  $\sum_{i=1}^n a_i = 2B$  une instance de **BIPART**, nous construisons une instance  $I = (G, M^*, w)$  de  $\text{INVCPM}$  2-équilibré de la manière suivante :

- 1) le graphe  $G$  a  $2n$  sommets et  $2n$  arêtes et est réduit à un cycle alternant les arêtes  $e_i$  et  $e_i^*$  pour  $i = 1, \dots, n$ ;
- 2) La fonction de poids  $w$  est donnée par  $w(e_i) = \frac{a_i}{2}$  et  $w(e_i^*) = 0$ ;
- 3) Le couplage est  $M^* = \{e_1^* \dots, e_n^*\}$ ;
- 4) Toute fonction de poids  $w'$  admissible doit vérifier  $w'(e_i) \in \{\frac{a_i}{2}, 3\frac{a_i}{2}\}$  et  $w'(e_i^*) \in \{0, a_i\}$ .

On vérifie aisément que  $G$  est biparti et que les contraintes de poids sont satisfaites. Un exemple de cette transformation est illustrée par la figure 1.6 (les arêtes en pointillés sont celles de  $M^*$ ). Nous affirmons que  $\{a_1, \dots, a_n\}$  contient une bipartition si



**Figure 1.6.** L'instance  $I = (G, M^*, w)$

et seulement s'il existe une fonction poids  $w^*$  satisfaisant les contraintes d'intégrité avec  $\sum_{e \in E} |w^*(e) - w(e)| \leq B$ .

Par construction du graphe  $G$ , il n'y a que 2 couplages parfaits :  $M^*$  et  $M = \{e_1, \dots, e_n\}$  avec  $w(M^*) = 0$  et  $w(M) = \sum_{i=1}^n \frac{a_i}{2} = B$ . Soit une bipartition  $S \subset \{1, \dots, n\}$  de  $\{a_1, \dots, a_n\}$ , on pose  $w^*(e_i^*) = a_i$  si  $i \in S$ ,  $w^*(e) = w(e)$  sinon. Nous avons  $w^*(M) = w(M) = B$  et  $w^*(M^*) = \sum_{i \in S} a_i = B$ . Ainsi,  $M^*$  est un couplage de poids maximum dans  $(G, w^*)$  et  $\sum_{e \in E} |w^*(e) - w(e)| = B$ .

Réciproquement, soit  $w^*$  une fonction de poids optimale vérifiant  $\sum_{e \in E} |w^*(e) - w(e)| \leq B$ . Nous savons que  $w^*(e_i) = \frac{a_i}{2}$  pour  $i = 1, \dots, n$  puisque les contraintes d'intégrité doivent vérifier  $w^*(e_i) \geq \frac{a_i}{2}$ . Ainsi, en posant  $S = \{i | w^*(e_i^*) \neq w(e_i^*)\}$ , nous avons  $w^*(M^*) = \sum_{i \in S} a_i = \sum_{i \in S} w^*(e_i^*) = \sum_{e \in E} |w^*(e) - w(e)| \leq B$ . D'un autre côté,  $M^*$  est un couplage parfait de poids maximum et donc  $\sum_{i \in S} a_i = w^*(M^*) \geq w^*(M) = w(M) = B$ . En rapprochant ces 2 inégalités, on déduit que  $S$  est une bipartition. ■

Pour le problème INVCPM, nous montrons ci-après qu'il est difficile en variables entières, alors que, comme nous l'avons vu dans le paragraphe 1.3.4, il est polynomial dans les graphes bipartis lorsque les variables sont binaires.

**THÉORÈME 1.10.**– [ALFANDARI 04] *Le problème INVCPM $_{\mathbb{N}^m}$  est NP-difficile, même si  $M^*$  est un couplage parfait et si  $\Delta(G) = 4$ .*

*Preuve.* nous établissons une réduction polynomiale à partir du problème de la couverture des arêtes par des sommets, appelé plus simplement *couverture par des sommets* et noté VC. Etant donné un graphe simple connexe  $G = (V, E)$ , il s'agit de trouver un sous-ensemble  $V^* \subseteq V$  de sommets de taille minimum vérifiant que chaque arête de  $G$  a au moins une de ses extrémités dans  $V^*$ . Le problème de décision associé

consiste, étant donné un graphe  $G$  et un entier  $k$ , à déterminer s'il existe une couverture satisfaisant  $|V^*| \leq k$ . Ce problème est **NP-complet**, même dans la classe des graphes de degré maximum 3 [GAREY 79].

Soit  $G = (V, E)$  un graphe connexe de degré maximum 3 où  $V = \{v_1, \dots, v_n\}$ ,  $E = \{e_1, \dots, e_m\}$  et  $k$  forment une instance de VC ; nous construisons polynomialement une instance  $I = (H, M^*, w)$  de  $\text{INVCPM}_{\mathbb{N}}$  de la manière suivante :

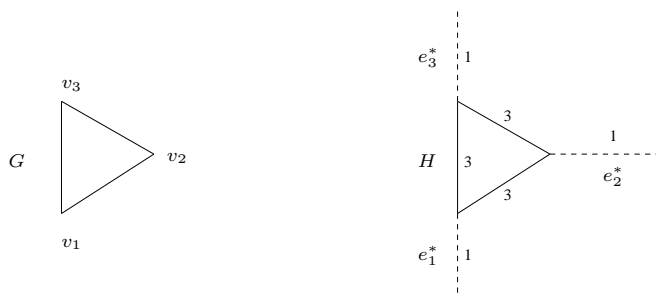
1) le graphe  $H = (V', E')$  a  $2n$  sommets avec  $V' = \{v_1, \dots, v_n, v_1^*, \dots, v_n^*\}$  et  $E' = \{e_1, \dots, e_m, e_1^*, \dots, e_n^*\}$ . Ainsi  $H$  contient  $G$  et possède  $n$  nouveaux sommets, ainsi que  $n$  arêtes définies par :  $e_i^* = [v_i, v_i^*]$  pour  $i = 1, \dots, n$  ;

2) la fonction de poids  $w$  est donnée par  $w(e_i) = 3$  et  $w(e_i^*) = 1$  ;

3) le couplage est  $M^* = \{e_1^* \dots, e_n^*\}$  ;

4) toute fonction de poids  $w'$  doit vérifier  $w'(e) \in \mathbb{N}$  pour  $e \in E'$ .

Un exemple de cette transformation est donné avec la figure 1.7. Remarquons trois faits : d'une part, le graphe  $H$  a un degré maximum égal à 4 (c'est-à-dire  $\Delta(H) = \Delta(G) + 1 = 4$ ) ; d'autre part,  $M^*$  est un couplage parfait de  $H$  ; enfin, les chaînes et cycles alternants augmentent tous  $M^*$  et sont les chaînes  $E_e$  de la forme  $E_e = \{e_i^*, e, e_j^*\}$  pour une arête arbitraire  $e = [v_i, v_j] \in E$ , de coût réduit  $\bar{w}(E_e) = -1$ .



**Figure 1.7.** Exemple de la transformation de  $G$  en  $I = (H, M^*, w)$

Nous affirmons qu'il existe une couverture par des sommets de  $G$  de taille au plus  $k$  si et seulement s'il existe une fonction de poids  $w^*$  à valeur entière telle que  $M^*$  est un couplage de poids maximum dans le réseau  $(H, w^*)$  et dont la somme des modifications  $\sum_{e \in E'} |w^*(e) - w(e)|$  n'excède pas  $k$ .

Soit  $V^*$  une couverture par des sommets avec  $|V^*| \leq k$ . On pose  $w^*(e_p^*) = 2$  si  $v_p \in V^*$  et  $w^*(e) = w(e)$  sinon. On vérifie aisément que  $\forall e \in E, \bar{w}^*(E_e) \geq 0$  puisque si  $e = [v_i, v_j]$  alors au moins une des 2 arêtes  $e_i^*$  ou  $e_j^*$  a un nouveau poids

de 2 ; ainsi, on déduit que  $M^*$  est un couplage de poids maximum dans  $(H, w^*)$ . De plus, on a  $\sum_{e \in E'} |w^*(e) - w(e)| = |V^*| \leq k$ .

Réciproquement, soit  $w^*$  une fonction de poids optimale qui rend le couplage  $M^*$  maximum dans  $(H, w^*)$  et qui vérifie :  $\sum_{e \in E'} |w^*(e) - w(e)| \leq k$ . De plus, supposons que  $w^*$  vérifie le lemme 1.1 ; sans perte de généralité, nous pouvons supposer que  $w(e_i^*) \leq w^*(e_i^*) \leq w(e_i^*) + 1$  pour  $i = 1, \dots, n$ , puisque nous n'augmenterons jamais le poids d'une arête de plus du maximum des coûts réduits des chaînes augmentantes contenant cette arête ; de même, on sait que  $w^*(e_j) = w(e_j)$  pour  $j = 1, \dots, m$ . Soit alors  $V^* = \{v_i | w^*(e_i^*) = w(e_i^*) + 1\}$ , on vérifie aisément que  $V^*$  est une couverture des arêtes par au plus  $k$  sommets de  $G$ . En effet, si  $V^*$  n'est pas une couverture, c'est qu'il existe  $e = [v_i, v_j] \in E$  avec  $v_i \notin V^*$  et  $v_j \notin V^*$  ; mais alors,  $\bar{w}^*(E_e) = \bar{w}(E_e) < 0$  et l'optimalité de  $M^*$  est contredite. Enfin,  $|V^*| = \sum_{e \in E'} |w^*(e) - w(e)| \leq k$ . ■

Dans [ALFANDARI 04], nous montrons par une réduction du même type que  $\text{INVCPM}_{\{0,1\}^{m'}}$  est **NP**-difficile.

REMARQUE 11.— Notons que ces résultats renforcent le résultat du paragraphe 1.4.1.1 sur la difficulté du stable maximum inverse  $\text{INVSPM}_A$  où  $A \in \{\mathbb{N}^n, \{0, 1\}^n\}$ . Les restrictions de ces deux problèmes à la classe des *line-graphes* [BERGE 73] restent difficiles.

REMARQUE 12.— Achevons l'étude des couplages inverses en observant que le résultat de difficulté tient également pour  $\text{INVCPM}_{\mathbb{N}^{m'}}$ , puisqu'il existe une réduction polynomiale à partir du problème  $\text{INVCPM}_{\mathbb{N}^{m'}}$  lorsque  $M^*$  est un couplage parfait (voir la remarque 11). Cela constitue un saut de difficulté entre les versions  $\text{INVCPM}_{\mathbb{N}^{m'}}$  et  $\text{INVCPM}_{\mathbb{R}^{m'}}$  puisque cette dernière version est polynomiale d'après [LIU 03]. Remarquons à ce propos que la fonction de poids optimale  $w^*$  de l'instance  $I = (H, M^*, w)$  décrite dans l'exemple issu de la figure 1.7 vérifie  $w^*(e_1^*) = w^*(e_2^*) = w^*(e_3^*) = \frac{3}{2}$  lorsque les contraintes d'intégrité sont réelles.

## 1.5. Conclusion

Ce chapitre se veut une introduction aux problèmes combinatoires inverses. Plus qu'une liste des résultats actuellement connus, notre objectif a été de sélectionner une palette représentative des différents types de problèmes et de résultats dans le cas de la norme  $L_1$ . Cette présentation permet de mettre en évidence les enjeux et les potentialités de ce domaine mais ne respecte pas, en revanche, les proportions, en quantités de travaux dédiés à chaque problématique, de la littérature existante. Une très grande majorité de travaux se sont focalisés jusqu'à présent sur les cas des variables continues. De même, nous avons préféré nous limiter à une norme et avons privilégié une problématique inverse. Le cas de la norme  $L_\infty$  donne lieu à des résultats et des techniques assez similaires et les travaux sur les autres normes restent encore marginaux

dans la littérature et constituent de vastes champs inexplorés. Le lecteur intéressé trouvera dans [HEUBERGER 04] un panorama complet des problèmes combinatoires inverses en variables continues déjà étudiés et un résumé des principaux résultats.

Parmi les potentialités encore très peu explorées, nous avons insisté sur quelques exemples de problèmes inverses en variables discrètes qui permettent, entre autres, de modéliser des situations où il s'agit de modifier la structure de l'instance plutôt que ses paramètres.

Outre l'étude de nouveaux problèmes sous ce point de vue, les cas difficiles soulèvent la question de l'approximation. De ce point de vue, l'intérêt du cadre inverse est qu'il induit de manière naturelle deux problématiques d'approximation : l'approximation de problèmes inverses difficiles et des versions inverses de problèmes d'approximation. La première est assez commune : il s'agit de considérer du point de vue de l'approximation de nouveaux problèmes. Quelques premiers résultats sont obtenus dans [ALFANDARI 04]. En revanche, la seconde problématique est radicalement différente : il s'agit de modifier l'instance (structure ou paramètres) pour qu'une solution fixée devienne une bonne solution au sens de l'approximation. Nous ne connaissons aucun résultat de ce type et, au premier abord, cette problématique semble assez difficile.

Une autre problématique est de faire la distinction entre deux objectifs : rendre la solution fixée  $x^*$  optimale ou la rendre optimale unique ? Imposer à  $x^*$  de devenir solution unique n'a pas toujours de sens (ou en tout cas pas toujours de solution) dans le cadre continu lorsque l'on modifie l'objectif. Par contre, elle se pose plus naturellement pour des problèmes inverses en variables discrètes.

Le dernier champs d'investigation que nous souhaitons mentionner est la notion de problème inverse par rapport à un algorithme fixé ou une classe d'algorithmes. L'idée est de relaxer la condition d'optimalité pour la solution fixée dans l'instance modifiée. Une manière de décrire un problème combinatoire inverse est de dire que l'on souhaite modifier à minima l'instance pour qu'un algorithme optimal quelconque puisse rendre  $x^*$  comme solution dans l'instance modifiée ou encore le forcer à choisir  $x^*$  lorsque l'on souhaite rendre  $x^*$  solution optimale unique. C'est d'ailleurs cette dernière formulation qui semble la plus pertinente dans ce contexte. Mais alors, une question naturelle est de poser le même problème, non pas contre tout algorithme optimal mais contre un algorithme spécifié (ou une classe d'algorithmes), optimal ou pas. Dans [ALFANDARI 04], nous envisageons cette problématique pour le problème de couplage et de stable maximum.

Nous considérons notamment le problème  $k$ -INVCPM $_{\{0,1\}^m}$  suivant qui utilise la notion de  $k$ -optimalité. Un couplage  $M$  est dit  $k$ -optimal si tout couplage obtenu à partir de  $M$  en retirant  $p$  arêtes et en en ajoutant  $p + 1$  pour  $p \leq k - 1$  n'est pas meilleur que  $M$ .  $k$ -INVCPM $_{0,1}^m$  consiste, étant donné un graphe et un couplage  $M^*$ ,

à modifier le graphe (ajout ou retrait d'arêtes) pour rendre  $M^*$   $k$ -optimal. Dans l'instance transformée,  $M^*$  peut être choisi par tout algorithme déterminant une solution  $k$ -optimale. Nous montrons que ce problème est polynomial pour  $k \leq 2$  et que pour tout  $k \geq 3$ , il est **APX**-complet (pour plus de détails sur cette notion, on se reportera au chapitre sur l'approximation et celui sur les réductions). Une conséquence est qu'il admet un algorithme polynomial garantissant un rapport constant et n'admet pas de schéma d'approximation polynomiale, si  $\mathbf{P} \neq \mathbf{NP}$ . Dans le même esprit, nous montrons que le problème consistant à retirer un nombre minimum de sommets d'un graphe pour rendre un stable fixé 2-optimal est **NP**-difficile. Enfin, nous considérons le problème de stable maximum inverse contre l'algorithme glouton pour le stable maximum : retirer un nombre minimum de sommets pour qu'un stable fixé soit choisi par l'algorithme glouton dans le graphe modifié est aussi un problème **NP**-difficile.

## 1.6. Bibliographie

- [AHUJA 01] AHUJA R. K., ORLIN J. B., « Inverse optimization », *Op. Res.*, vol. 49, p. 771-783, 2001.
- [ALFANDARI 04] ALFANDARI L., DEMANGE M., MONNOT J., « A general framework for inverse optimization problems », Working paper, Lamsade, University of Paris-Dauphine, 2005.
- [BERGE 73] BERGE C., *Graphs and hypergraphs*, North Holland, Amsterdam, 1973.
- [BURTON 92] BURTON D., TOINT P. L., « On an instance of the inverse shortest paths problem », *Math. Programming*, vol. 53, p. 45-61, 1992.
- [CAI 97] CAI M., YANG X., ZHANG J., Inverse problems with partial given solution, Working paper, Department of Mathematics, City University of Hong Kong, 1997.
- [CAI 99] CAI M., YANG X., ZHANG J., « The complexity analysis of the inverse center location problem », *J. Global Optim.*, vol. 15, p. 213-218, 1999.
- [CHRISTOFIDES 75] CHRISTOFIDES N., *Graph Theory : An Algorithmic Approach*, Academic Press, New York, 1975.
- [GAREY 79] GAREY M. R., JOHNSON D. S., *Computers and intractability. A guide to the theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.
- [GONDRAN 98] GONDRAN M., MINOUX M., *Graphes et Algorithmes*, Eyrolles, Paris, 1998.
- [GRÖTSCHEL 81] GRÖTSCHEL M., LOVÁSZ L., SCHRIJVER A., « The ellipsoid method and its consequences in combinatorial optimization », *Combinatorica*, vol. 1, p. 169-197, 1981.
- [HEUBERGER 04] HEUBERGER C., « Inverse optimization : a survey on problems, methods, and results », *J. Global Optim.*, 2004.
- [HU 98] HU Z., LIU Z., « A strongly polynomial algorithm for the inverse shortest arborescence problem », *Discrete Applied Mathematics*, vol. 82, p. 135-154, 1998.

- [HUANG 99] HUANG S., LIU Z., « On the inverse problem of linear programming and its application to minimum weight perfect  $k$ -matching », *European Journal of Operational Research*, vol. 112, p. 421-426, 1999.
- [KHACHIYAN 79] KHACHIYAN L., « A polynomial algorithm in linear programming », *Soviet Mathematics Doklady*, vol. 20, p. 191-194, 1979.
- [KORTE 02] KORTE B., VYGEN J., *Combinatorial optimization : Theory and Algorithms*, Springer, Berlin, 2002.
- [LIU 03] LIU Z., ZHANG J., « On Inverse Problems of Optimum Perfect Matching », *J. Comb. Optim.*, vol. 7, p. 215-228, 2003.
- [MICALI 80] MICALI S., VAZIRANI V. V., « An  $O(\sqrt{|V|}|E|)$  Algorithm for Finding Maximum Matching in General Graphs », *FOCS*, p. 17-27, 1980.
- [MOSER 91] MOSER T. J., « Shortest paths calculation of seismic rays », *Geophysics*, vol. 56, p. 59-67, 1991.
- [PAPADIMITRIOU 94] PAPADIMITRIOU C. H., *Computational Complexity*, Addison-Wesley, Reading, MA, 1994.
- [PAPADIMITRIOU 98] PAPADIMITRIOU C. H., STEIGLITZ K., *Combinatorial optimization : algorithms and complexity*, Dover, New York, 1998.
- [TARANTOLA 87] TARANTOLA A., *Inverse Problem Theory : Methods for Data Fitting and Model Parameter Estimation*, Elsevier, Amsterdam, 1987.
- [YANG 97] YANG C., ZHANG J., MA Z., « Inverse maximum flow and minimum cut problems », *Optimization*, vol. 40, p. 147-170, 1997.
- [YANG 99] YANG C., ZHANG J., « Two general methods for inverse optimization problems », *Applied Mathematics Letters*, vol. 12, p. 69-72, 1999.
- [YANG 01] YANG X., « Complexity of Partial Inverse Assignment Problem and Partial Inverse Cut Problem », *RAIRO Oper. Res.*, vol. 35, p. 117-126, 2001.
- [ZHANG 96a] ZHANG J., LIU Z., « Calculating some inverse linear programming problems », *J. Comp. and App. Math.*, vol. 72, p. 261-273, 1996.
- [ZHANG 96b] ZHANG J., LIU Z., MA Z., « On inverse problem of minimum spanning tree with partition constraints », *Mathematical methods of Op. Res.*, vol. 44, p. 347-358, 1996.
- [ZHANG 97] ZHANG J., XU S., MA Z., « An algorithm for inverse minimum spanning tree problem », *Optimization methods and software*, vol. 8, p. 69-84, 1997.
- [ZHANG 99] ZHANG J., LIU Z., « A further study on inverse linear programming problems », *J. Comp. and App. Math.*, vol. 106, p. 345-359, 1999.
- [ZHANG 00] ZHANG J., LIU Z., MA Z., « Some reverse location problems », *European J. Oper. Res.*, vol. 124, p. 77-88, 2000.

## Chapitre 2

# Index

- Bipartition (BIPART), 35
- Bipartition paire (BIPARTPAIRE), 31
- Centre d'un réseau, 28
- Chaîne augmentante, 23, 34
- Clause, 28
- cocycle sortant ( $\Omega^+(K)$ ), 15
- Contraintes du problème inverse, 5
- Coupe de capacité minimum inverse (INVCOUPEMIN), 15
- Coupe de capacité minimum (COUPEMIN), 15
- Couplage, 23
  - de poids maximum (CPM), 33
  - maximal, 23
  - maximum, 23
  - parfait, 33
  - parfait de poids maximum (CPPM), 33
- Couplage de poids maximum inverse (INVCPM), 22, 23, 34
- Couplage parfait de poids maximum inverse (INVCPPM), 25, 34
- Couverture par des sommets (VC), 36
- Cycle Hamiltonien, 27
- Fonction objective, 4
- Littéral, 28
- Localisation d'un centre (1-LC), 28
- Localisation de centre inverse (INV1-LC), 25, 28
- Nombre de stabilité ( $\alpha(G)$ ), 25
- Norme  $L_p$  ( $\|\cdot\|_p$ ), 4
- Problème évaluatif inverse, 6
- Problème d'affectation, 33
- Problème d'optimisation combinatoire, 4
- Problème d'optimisation combinatoire inverse (INV $\pi_C$ ), 4
- Problème inverse partiel, 5
- Satisfiabilité (SAT), 28
- Solution réalisable, 4
- Stable, 25
- Stable de poids maximum (SPM), 25
- Stable de poids maximum inverse (INVSPM), 26
- Variable booléenne, 28
- Voyageur de commerce maximum (TSPMAX), 27