



**HAL**  
open science

## Quelques résultats de complexité pour un problème de conception de réseaux radio multi-sauts robustes.

Benoit Darties

► **To cite this version:**

Benoit Darties. Quelques résultats de complexité pour un problème de conception de réseaux radio multi-sauts robustes.. [Rapport de recherche] RR-07022, LIRMM. 2007. lirmm-00182422

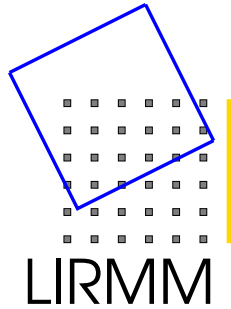
**HAL Id: lirmm-00182422**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00182422>**

Submitted on 25 Oct 2007

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

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



LABORATOIRE D'INFORMATIQUE, DE  
ROBOTIQUE ET DE MICROÉLECTRONIQUE DE  
MONTPELLIER

LIRMM, Univ. Montpellier 2, CNRS

## RAPPORT DE RECHERCHE

**Quelques résultats de complexité  
pour un problème de conception de  
réseaux radio multi-sauts robustes.**

**Benoît Darties**

octobre 2007

R.R.LIRMM 07022

# Table des matières

<b>1</b>	<b>Conception de réseaux radio multi-sauts robustes</b>	<b>2</b>
1.1	Le problème de conception . . . . .	2
1.2	Modélisation . . . . .	6
1.2.1	Modélisation de la donnée . . . . .	6
1.2.2	Formulation du problème . . . . .	7
<b>2</b>	<b>Étude de complexité</b>	<b>12</b>
2.1	NetworkDesign sans contrainte . . . . .	12
2.2	NetworkDesign avec un seul type de contraintes . . . . .	15
2.2.1	Seul le degré est borné . . . . .	15
2.2.2	Seul le nombre de sauts est borné . . . . .	16
2.2.3	Seule la robustesse est imposée. . . . .	20
2.3	NetworkDesign avec deux types de contraintes . . . . .	20
2.3.1	Degré et nombre de sauts bornés . . . . .	20
2.3.2	Degré borné et robustesse imposée . . . . .	24
2.3.3	Nombre de sauts borné et robustesse imposée . . . . .	25

## Résumé

Nous étudions dans ce rapport la complexité d'un problème de conception de réseaux radio multi-sauts, posé au départ par une entreprise de télécommunications du sud de la France. L'objectif de cette entreprise est d'assurer la distribution d'un flux internet haut-débit dans des zones rurales non couvertes en technologie ADSL. Ce flux internet est récupéré à partir de villes sources et doit être acheminé à destination de villages clients par un jeu de relais. Un village client peut également servir de relais. La technologie radio est utilisée pour établir un lien entre deux nœuds du réseau (villes, villages ou relais). Le déploiement d'un tel réseau est soumis à plusieurs contraintes. La difficulté du problème consiste à définir quels liens établir pour assurer une distribution efficace, et ce en respectant les contraintes posées.

Nous n'utilisons pas d'antennes omnidirectionnelles pour établir les communications entre nœuds. Un lien est rendu possible par la mise en place d'une antenne directionnelle sur chacune de ses extrémités. La communication est possible si et seulement si la qualité du signal est suffisamment élevée. L'utilisation de cette technologie offre une meilleure robustesse aux interférences. Nous supposons ces dernières inexistantes.

Ce rapport se décompose en 2 chapitres :

- Le premier chapitre précise les conditions dans lesquelles sont abordées le problème de conception. Nous y présentons notamment les différentes contraintes que doit satisfaire le réseau à concevoir, ainsi qu'une modélisation formelle du problème.
- Nous proposons dans le deuxième chapitre une étude de complexité du problème de conception. Nous distinguons trois paramètres régissant trois principales contraintes de déploiement. Notre étude établit la difficulté du problème selon si certains de ces trois paramètres sont considérés ou non. Nous montrons notamment dans quelles mesures le problème de conception peut se ramener à des problèmes algorithmiques bien connus, et quels résultats existent alors sur ces derniers.

# Chapitre 1

## Conception de réseaux radio multi-sauts robustes : objectifs et contraintes

*Ce chapitre introductif définit dans une première section le problème de conception d'un réseau radio multi-sauts, dont la résolution doit permettre l'acheminement d'un flux réseau vers un ensemble de destinations. Nous y présentons les contraintes que le réseau déployé doit satisfaire. La seconde section précise la modélisation du problème.*

### 1.1 Le problème de conception

L'objectif des travaux de cette partie consiste à concevoir un réseau radio assurant la distribution de bande passante depuis des nœuds sources vers des nœuds clients géographiquement proches. Cette distribution est soumise à plusieurs contraintes de divers ordres.

D'un point de vue fonctionnel, les nœuds du réseau représentent :

- des nœuds sources,
- des nœuds clients,
- des relais additionnels qu'il est possible de mettre en place pour assurer la distribution.

La dimension des réseaux considérés est à l'échelle pluri-urbaine, et peut s'étendre à un département ou une région :

- Les nœuds sources représentent des villes disposant d'un accès haut-débit à Internet et qu'il est possible d'exploiter. Un nœud source est donc l'association d'un routeur et d'un *point de collecte* fournissant un flux haut-débit.
- Les nœuds clients, ou *points de desserte*, sont des villages pour lesquels une demande de couverture haut-débit (une bande passante) a été définie,

de taille par exemple proportionnelle à la taille du village. D'un point de vue technique, ces nœuds désignent un routeur auquel est connecté un sous-réseau. Ce sous-réseau représente l'ensemble des utilisateurs qui vont bénéficier de la bande passante demandée par ce routeur. Remarquons qu'un nœud client peut également être un nœud source. Ce statut implique la connexion d'un point de collecte au routeur du nœud.

- Les nœuds *relais* sont des sites présentant un intérêt certain pour être envisagés comme relais potentiels dans le réseau à concevoir. Ils peuvent être des villes dont la demande de couverture est nulle, ou des points stratégiques (ex. structures ou sommets élevés, . . .) permettant d'atteindre des destinations plus éloignées. L'équipement déployé sur ces sites se limite à un routeur, mais nécessite une source électrique à proximité.

Nous utilisons le média radio comme support de communication. Une communication directe entre deux nœuds du réseau nécessite sur chacun d'entre eux la mise en place d'une antenne directionnelle. Ces antennes doivent être en vision directe l'une de l'autre, et sont par conséquent placées sur des sites situés en hauteur (clochers, châteaux d'eau, . . .).

La figure 1.1 présente les différents types de nœuds rencontrés. Nous limiterons notre étude au déploiement d'un réseau radio assurant la connexion inter-routeurs (nous n'étudierons pas la distribution du flux depuis un routeur aux utilisateurs finaux qui lui sont rattachés).

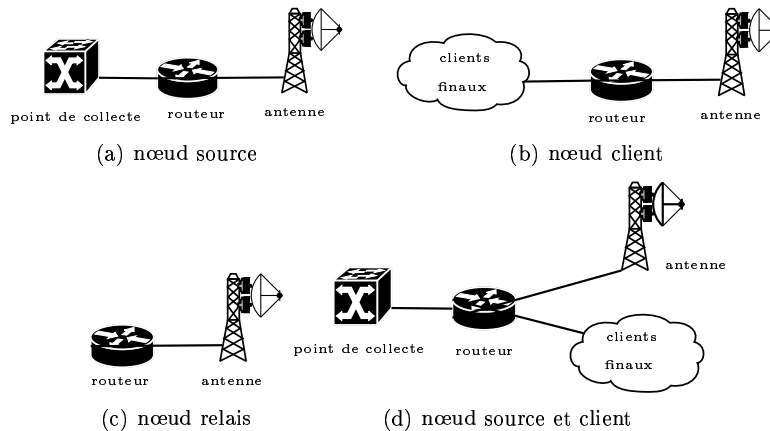


FIG. 1.1 – Types de nœuds envisagés.

### Établir une communication directe

L'utilisation d'antennes directionnelles s'opère par le déploiement suivant :

1. Les deux antennes doivent être orientées en direction l'une de l'autre, et en ligne de vue.

2. Une antenne est reliée à un routeur par une carte réseau qui sert d'interface entre ces deux entités.
3. Le fonctionnement d'une antenne est régi par un protocole issu de la norme 802.11. Une communication directe entre deux nœuds requiert que les deux antennes impliquées utilisent le même protocole.

**Remarque 1** *Le long de notre étude, nous limitons par soucis de clarté le nombre de protocoles disponibles, et ne considérons que les normes 802.11a et 802.11b. Néanmoins l'ajout de technologies supplémentaires n'affecte en rien la définition et la résolution du problème de conception.*

L'existence d'un lien radio entre deux nœuds est déterminée par la qualité du signal émis depuis l'un des nœuds vers l'autre, qui dépend notamment :

- de la puissance d'émission de la transmission,
- de facteurs environnementaux et géographiques.

Établir la qualité d'un lien entre deux sites n'est pas évident. Une méthode exacte consiste à positionner un émetteur sur un site, un récepteur sur l'autre site, et mesurer la qualité du lien. Cette mesure doit également être opérée dans l'autre sens, la mesure la plus faible est alors retenue. Cette démarche est coûteuse en temps et en moyens. Une solution à coût réduit consiste à utiliser un simulateur de qualité de liens, qui prend en entrée :

- les données topologiques et géographiques d'une zone,
- les coordonnées GPS des sites sélectionnés.

Il en ressort deux matrices :

1. une matrice de qualité des liens par paire de nœuds,
2. une matrice des distances par paire de nœuds.

Un premier filtrage des liens est opéré en supprimant les liens dont la qualité de signal se situe au dessous d'un seuil critique. Lorsqu'un lien radio possède un signal suffisamment fort, il convient d'en définir alors sa capacité, c'est à dire le débit maximal qu'il peut faire transiter. Les principaux facteurs qui peuvent influencer sur la capacité d'un lien sont :

1. le type d'antennes (matériel),
2. la distance entre les antennes,
3. le protocole de communication utilisé.

Les caractéristiques techniques d'une antenne directionnelle indiquent les débits atteignables en fonction de la distance et du protocole utilisé. Nous associons à chaque lien le protocole qui permet la plus grosse capacité (voir figure 1.2).

**Remarque 2** *Nous supposons ici que le coût d'un lien n'est pas lié au choix d'un protocole. Dans le cas contraire, on peut considérer pour chaque paire de nœuds autant de liens que de protocoles disponibles, et associer à chacun de ces liens un coût en fonction de son protocole.*

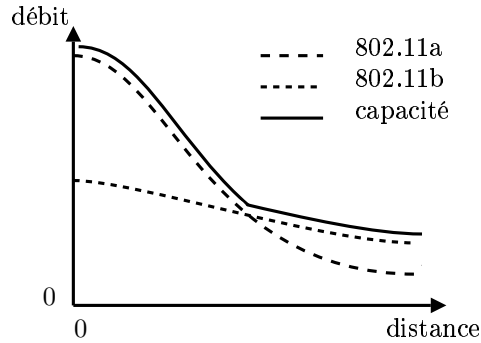


FIG. 1.2 – Capacité d'un lien en fonction de la distance et du protocole.

### Pannes envisageables

Trois types de pannes peuvent intervenir :

1. une panne sur un lien radio : la liaison entre deux nœuds est interrompue. Ce cas survient généralement lors de pannes matérielles sur une carte réseau ou une antenne, mais peut aussi avoir des raisons logicielles. Parmi les causes liées à ces pannes citons la défaillance d'une carte réseau suite à une surchauffe, ou un coup de foudre sur l'antenne correspondante. Le routeur peut cependant continuer à distribuer un flux sur le sous-réseau qu'il connecte, ou acheminer les paquets aux autres routeurs.
2. une panne sur un routeur : si une panne intervient sur le routeur d'un nœud, tous les liens assurant les communications vers les routeurs voisins deviennent hors d'usage. Aucun flux ne peut être acheminé vers le sous-réseau qu'il connecte. Ce type de panne intervient généralement si un impact de foudre sur une antenne s'est propagé à tout le routeur, ou en cas de coupure d'électricité.
3. une panne sur un point d'accès : en cas de panne sur un point d'accès, le routeur reste connecté au réseau et à son sous-réseau, mais perd sa capacité à délivrer un flux. Il se comporte comme un simple relai. Lorsque le nœud concerné est également un nœud client, le sous-réseau correspondant ne reçoit plus le flux associé au routeur du nœud.

### Contraintes de déploiement

Le déploiement du réseau est soumis à plusieurs contraintes de divers ordres :

**Contraintes de demande :** chaque nœud client exige un débit spécifique en bande passante, qui doit lui être assuré dans des conditions normales de fonctionnement.

**Contraintes de capacité des liens :** le flux circulant sur un lien est borné par la capacité de ce dernier.



**Contraintes de degré :** le degré d'un nœud du réseau est égal au nombre d'antennes directionnelles présentes sur ce dernier. La charge de travail d'un routeur est proportionnelle au nombre d'antennes qu'il commande. Ainsi un nombre élevé d'antennes peut entraîner des difficultés de fonctionnement (temps de traitement de paquets élevés, surchauffe du routeur, ...). De plus, le nombre de slots disponibles pour les cartes réseau est borné sur chaque routeur. Le nombre d'antennes associées au même routeur ne doit donc pas excéder une certaine valeur.

**Contraintes de nombre de sauts :** la distribution d'un nœud client est assurée par un ou plusieurs nœuds sources. Pour des raisons de temps d'accès et de réponse, le nombre de sauts entre un nœud client et la (les) source(s) dont il dépend ne peut excéder une certaine valeur.

**Contraintes de robustesse :** Dans la mesure du possible, le réseau doit être résistant aux pannes, c'est à dire assurer une bande passante non nulle à tous les nœuds clients si une anomalie survient, sur un nœud, un lien. Quelle que soit la localisation de la panne (routeur, lien, ou point d'accès), les nœuds du réseau <sup>1</sup> doivent restés connectés à au moins un nœud source, tout en respectant les contraintes de nombre de sauts énoncées précédemment. Nous nous autorisons en cas de panne à dégrader le débit reçu par certains nœuds clients, afin d'assurer une connexion avec l'extérieur à tous les nœuds.

Le coût pour déployer un tel réseau est surtout induit par l'achat et l'installation des antennes directionnelles. Nous chercherons donc à minimiser leur nombre.

Nous appelons *NetworkDesign* le problème consistant à définir un réseau entre les routeurs des nœuds qui répond à toutes les contraintes énoncées ci-avant. La définition formelle de ce problème est donnée en fin de section 1.2.

## 1.2 Modélisation

Cette section propose la formalisation du problème *NetworkDesign*. La première sous-section présente les éléments d'une instance de *NetworkDesign*. La seconde sous-section introduit le vocabulaire inhérent au problème, et se conclut par la définition formelle de *NetworkDesign*.

### 1.2.1 Modélisation de la donnée

Nous modélisons l'ensemble des informations nécessaires à la conception du réseau comme suit :

- Nous modélisons la donnée par un graphe non orienté  $G = (V, E)$ , où  $V$  désigne l'ensemble des nœuds du réseau, et  $E$  l'ensemble des liens déployables de signal suffisamment grand.

---

<sup>1</sup>à l'exception du nœud sur lequel intervient une panne

- Une fonction *capa* associe à chaque arête  $e \in E$  un réel dans  $R^{*+}$  correspondant à sa capacité.
- Nous nommons  $S \subseteq V$  et  $T \subseteq V$  les ensembles de sommets correspondant respectivement aux nœuds sources et aux nœuds clients.
- Une fonction *dem* associe pour chaque élément  $t \in T$  un réel dans  $R^{*+}$  désignant la demande de bande passante de  $t$ .

Répondre au problème NetworkDesign consiste à trouver un graphe  $H$ , sous-graphe partiel de  $G$ , et répondant aux contraintes présentées précédemment. Nous proposons dans la sous-section suivante une définition formelle de ce problème. Parmi les contraintes de conception énoncées, le degré des nœuds et le nombre de sauts entre une source et un client doivent être bornés. De plus le réseau doit supporter un certain nombre de pannes simultanées. Nous notons respectivement *deg*, *hop* et  $r$  le degré maximal, le nombre de sauts maximum, et le nombre de pannes simultanées que peut supporter le réseau.

### 1.2.2 Formulation du problème

Le vocabulaire suivant précède la définition du problème :

**Définition 1 (*r*-robustesse) :**

Soient un graphe  $G = (V, E)$ , un sous-ensemble  $S \subseteq V$ , un entier naturel  $dist \geq 1$ , et un sommet  $x \in V$ .

Un sommet  $x \notin S$  est *r*-robuste dans  $(G, S, d)$ , si et seulement s'il existe  $r + 1$  chaînes  $\{C_1, C_2, \dots, C_{r+1}\}$ ,  $C_i \subseteq G$ , telles que :

- les extrémités de  $C_i$  sont  $x$  et  $s_i$ , avec  $s_i \in S$ ,
- $V(C_i) \cap V(C_j) = \{x\}$ ,  $\forall 1 \leq i < j \leq r + 1$ .

Réciproquement, nous disons que l'ensemble de chaînes  $\{C_1, C_2, \dots, C_{r+1}\}$ ,  $C_i \subseteq G$ , assure la *r*-robustesse de  $x \notin S$  dans  $(G, S)$ .

Si  $x$  est inclus dans  $S$ ,  $x$  est *r*-robuste dans  $(G, S)$  si et seulement s'il existe  $r$  chaînes avec les conditions énoncées précédemment, puisque  $x$  est déjà connecté à un point de collecte.

Nous disons qu'un ensemble de sommets  $T$  est *r*-robuste dans  $(G, S)$  si chacun de ses sommets  $x \in T$  est *r*-robuste dans  $(G, S)$ . Réciproquement, nous disons qu'un ensemble de chaînes assure la *r*-robustesse de  $T$  dans  $(G, S)$  si et seulement s'il assure la *r*-robustesse de chacun des éléments de  $T$ .

**Remarque 3** La notion de résistance aux pannes est ici portée sur les nœuds du réseau, et se veut logiquement plus forte que la résistance aux pannes sur les liens. Le modèle de robustesse proposé et l'ensemble de nos travaux peuvent se décliner aisément pour ne considérer que des pannes intervenant sur les liens.

Dans le problème de conception, l'ensemble des nœuds clients  $T$  doit donc être *r*-robuste dans le réseau déployé.

Le terme de flot est couramment employé dans la terminologie des réseaux pour décrire la quantité de matière transitant sur chacun des liens du réseau. Les problèmes de flots cherchent généralement à satisfaire des demande de trafic

entre paires de nœuds selon des contraintes liées aux capacités des liens. Nous proposons ici une définition d'un flot chaînes-contraint :

**Définition 2 (flot chaînes-contraint) :**

Soient un graphe non orienté  $G = (V, E)$ , une valuation des arêtes  $capa : E \rightarrow R^{*+}$ , un ensemble  $S \subseteq V$  de nœuds sources, un ensemble  $T \subseteq V$  de nœuds clients, et une fonction  $dem : T \rightarrow R^{*+}$ .

Considérons alors un ensemble de chaînes  $C$  tel que  $\forall c \in C, ext(c) = \{s_c, t_c\} | s_c \in S, t_c \in T$ . Appelons  $P_C(e)$  l'ensemble des chaînes de  $C$  contenant l'arête  $e$ .

Un flot chaînes-contraint valide est une fonction  $f : C \rightarrow Z^+$  qui associe à chaque chaîne  $c$  de  $C$  un réel positif ou nul tel que, en notant  $h(e) = \sum_{c \in P_C(e)} f(c)$  avec  $e \in E$  :

- $h(e) \leq capa(e), \forall e \in E$
- $\sum_{c \in C | t_c \in ext(c)} f(c) \geq dem(t), \forall t \in T$

Nous formalisons NetworkDesign par le problème de décision 1.2.1.

**Donnée** :

- un graphe non orienté  $G = (V, E)$ ,
- une fonction  $capa : E \rightarrow R^{*+}$ ,
- deux sous-ensembles  $S$  et  $T$  de  $V$ ,
- une fonction  $dem : T \rightarrow R^{*+}$ ,
- les entiers naturels  $deg, hop, r, k$ .

**Question** :

existe-t-il un ensemble de chaînes  $C$  de  $G$  avec  $ext(c) = \{s_c, t_c\} | s_c \in S, t_c \in T, \forall c \in C$ , et un flot chaînes-contraint  $f$  sur  $C$  tel que, en notant  $H = (V_H, E_H)$  le graphe pour lequel lequel  $V_H = \bigcup_{c \in C} V(c)$  et  $E_H = \bigcup_{c \in C} E(c)$ , on a :

1.  $|N_H(x)| \leq deg, \quad \forall x \in V(H)$ ,
2.  $\forall c \in C, |E(c)| \leq hop$ ,
3.  $f$  est valide pour l'instance  $(H, capa, S, T, dem, C)$ ,
4. l'ensemble des chaînes  $C$  assure la  $r$ -robustesse de  $T$  dans  $(H, S)$ ,
5.  $|E_H| \leq k$ .

**Problème de décision 1.2.1:** le problème NetworkDesign

Clairement, le problème NetworkDesign recherche à partir d'un réseau donné un sous-réseau dans lequel :

- les nœuds sont de degré borné, en accord avec la condition 1,

- en fonctionnement normal et d’après les condition 2 et 3, les demandes de nœuds clients sont satisfaites par des nœuds sources situés à un nombre de sauts borné,
- le sous-réseau peut supporter un nombre donné de pannes, en accord avec la condition 4.

Une instance de NetworkDesign est a priori décrite par un 9-uplet  $(G, capa, S, T, dem, k, deg, hop, r)$ . Nous cherchons une topologie dont le nombre de liens est borné par une constante  $k$ . La version optimisation de ce problème, appelée min-NetworkDesign, consiste à trouver le sous-réseau ayant un nombre minimum de liens. La donnée  $k$  est alors supprimée de la donnée.

Remarquons que la donnée des nœuds sources, des nœuds clients, des demandes qui leur sont associées, des liens et de leur capacité sont des données de terrain. À l’opposé, les contraintes de robustesse, de distance et de degré relèvent plutôt de choix techniques et de politiques de déploiement. Par souci de clarté, nous utiliserons parfois une notation consistant à extraire les paramètres techniques  $deg, hop$  et  $r$  de la donnée du problème. Ainsi le problème  $[deg, hop, r]$ -NetworkDesign réfère à une instance de NetworkDesign dans laquelle le degré maximum est borné par  $deg$ , le nombre de sauts entre une source et un client est d’au plus  $hop$ , et le réseau déployé doit être  $r$ -robuste. La donnée du problème est réduite à un 6-uplet  $(G, capa, S, T, dem, k)$ .

La figure 1.3(a) présente une instance de  $[3, 3, 1]$ -min-NetworkDesign. Les sommets  $A$  et  $F$  sont des sommets sources. Les sommets  $B$  à  $F$  sont des sommets clients. La demande de chacun des sommets est indiquée entre crochets. La capacité de chaque arête est ici fixée à 3. La figure 1.3(b) présente une solution de coût 13. Le tableau 1.1 détaille la solution, en énumérant pour chaque client la ou les chaînes assurant le transport du flot, et les chaînes de secours utilisées en cas de panne.

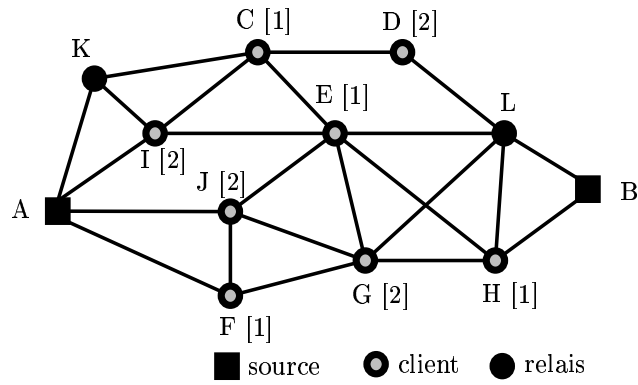
<i>client</i>	<i>chaînes [flot]</i>	<i>secours</i>	<i>client</i>	<i>chaînes [flot]</i>	<i>secours</i>
C	A-I-C [1]	B-L-D-C	G	B-H-G [2]	A-K-G
D	B-L-D [2]	A-I-C-D	H	B-H [1]	A-F-G
E	B-L-E [1]	A-I-E	I	A-I [2]	B-L-E-I
F	A-F [1]	B-H-G-F	J	A-J [2]	B-H-G-K

TAB. 1.1 – Listes des chaînes assurant la distribution du flot et la robustesse par sommet.

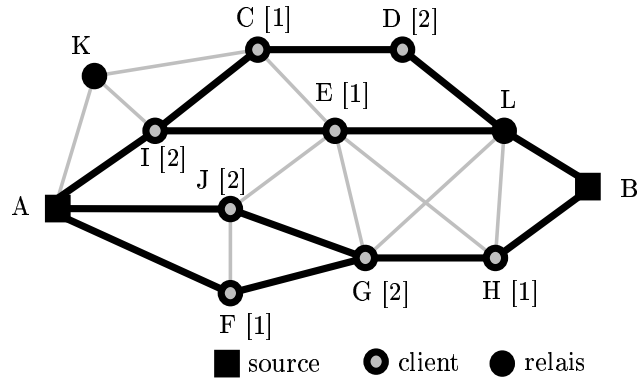
Nous terminons cette section par l’énoncé de la remarque suivante :

**Remarque 4** *Dans le problème tel que nous avons énoncé, les sommets du graphe résultat ont les mêmes contraintes de degré, de robustesse, et de distance. En adaptant notre formulation, nous pourrions définir en résultat un graphe dans lequel les paramètres critiques sont propres à chaque nœud. En effet :*

- *Au niveau du degré, certains sites peuvent accueillir plus d’antennes que d’autres.*



(a) Une instance ayant 2 sources et 8 clients



(b) Une solution de coût 13

FIG. 1.3 – Une instance de [3,3,1]-min-NetworkDesign et sa solution

- *Au niveau de la robustesse, on peut souhaiter que certains clients soient plus robustes que d'autres si la proportion d'utilisateurs finaux est supérieure.*
- *Enfin, augmenter la distance maximum entre une source et un client précis peut amener de nouvelles solutions moins coûteuses en nombre de liens.*

*Nous mentionnons l'existence de ces variantes qui peuvent se formaliser aisément par le vocabulaire que nous avons introduit. Cependant, nous restreindrons notre étude dans ce mémoire au problème présentant des contraintes identiques pour tous les nœuds.*

## Chapitre 2

# Étude de complexité

*Nous proposons une étude de complexité du problème `min-NetworkDesign` en fonction des valeurs des paramètres de degré, nombre de sauts et robustesse, respectivement désignées par les valeurs `deg`, `hop` et `r`. Ce chapitre regroupe à la fois un état de l'art sur des problèmes similaires, et instaure de nouvelles preuves de complexité. Par la suite, lorsqu'une contrainte est ignorée, nous lui attribuons une valeur correspondant à cet état : les affectations `deg = ∞` et `hop = ∞` impliquent respectivement que le degré maximum autorisé et le nombre maximum de sauts peuvent être arbitrairement grands (de l'ordre de  $n$ ), et l'affectation `r = 0` que le réseau ne tolère aucune panne.*

*Organiser les nombreux problèmes résultant des variations de ces paramètres n'est pas chose facile. Aussi nous découperons notre étude en trois sections. La première section décrit quelques résultats sur le problème `min-NetworkDesign` lorsqu'aucun des trois types de contraintes (degré, nombre de sauts, robustesse) n'est considéré. Les sections 2, et 3 s'intéressent aux problèmes pour lesquels respectivement 1 et 2 types de contraintes sont envisagés.*

### 2.1 NetworkDesign sans contrainte

Le problème  $[\infty, \infty, 0]$ -`NetworkDesign` décrit exactement le problème du `EDGE-COST FLOW (ECF)`, connu pour être NP-complet [GJ79]. Il reste NP-complet même sous l'hypothèse que les capacités des arêtes sont infinies. On le rencontre alors parfois sous le nom `INFINITE CAPACITY EDGE-COST FLOW (ICECF)` [EKS05].

Les deux propositions suivantes illustrent l'impact de la cardinalité des ensembles sources et clients sur la complexité de ces deux familles de problèmes.

**Proposition 1 :**

1. *Les deux précédents problèmes restent NP-complets, lorsque le nombre de sources est de 1.*

2. Lorsque le nombre de clients est de 1, ECF reste NP-complet.
3. En revanche, lorsque le nombre de clients est de 1, ICECF devient polynomial.

**Preuve :**

Soit une instance  $I$  de ECF contenant (entre autres) un ensemble de sommets sources  $S$  dans un graphe  $G = (V, E)$ . À partir de  $I$ , on peut construire polynomialement une instance  $I'$  du même problème dans laquelle le nombre de sources est de 1, en copiant  $I$  dans  $I'$  et en *fusionnant*<sup>1</sup> les sommets de  $S$  dans  $G$ . Une solution à  $I$  de coût  $k$  induit alors une solution à  $I$  de coût équivalent, et réciproquement.

Lorsque les arêtes sont de capacité infinies, et qu'il n'y a qu'une seule source, ce problème n'est autre que celui du STEINER TREE (StT), bien connu pour être NP-complet. Ceci conclut la preuve du point 1 de la proposition 1.

Soit une instance  $I$  de ECF contenant (entre autres) un ensemble de clients  $C = \{c_1, c_2, \dots, c_m\}$  dans un graphe  $G = (V, E)$ . On peut construire polynomialement une instance  $I'$  du même problème comme suit :

- $I'$  est initialisé par copie de  $I$ .
- Nous ajoutons à  $G$  un sommet  $c$ , et les arêtes  $\{c_i, c\} | c_i \in C$ .
- La capacité de  $\{c_i, c\}$  est égale à  $dem(c_i)$ .
- Le sommet  $c$  est désigné comme seul client, avec une demande  $dem(c) = \sum_{c_i \in C} dem(c_i)$ .

Par la suite, nous nommerons cette construction l'*ajout d'un super client*.

S'il existe une solution à  $I'$  de coût  $k$ , alors elle mobilise chacune des arêtes  $\{c_i, c\} | c_i \in C$  à pleine capacité. On peut en déduire une solution à  $I$  de coût  $k - |C|$ . réciproquement, toute solution à  $I$  de coût  $k$  implique une solution dans  $I'$  de coût  $k + |C|$ . La complexité du problème reste inchangée, comme l'indique le point 2 de la proposition 1.

Nous terminons par la justification du point 3 : Lorsque les capacités des arêtes sont infinies et qu'il n'y a qu'un seul client  $c$ , il suffit de déterminer le plus court chemin entre  $s$  et une source quelconque pour obtenir un ensemble d'arêtes permettant le transit du flux. Cette opération est bien connue pour être en  $O(n^2)$  en terme de complexité.  $\square$

La preuve de la proposition 2 utilise une réduction vers le problème BI-PARTITION. Ce problème consiste à se demander s'il existe une bipartition  $\{N_A, N_B\}$  d'un ensemble  $S$  d'entiers, telle que la somme des éléments de  $N_1$  égale celle des éléments de  $N_2$ . Ce problème est connu pour être NP-complet.

---

<sup>1</sup>L'opération de fusion d'un ensemble de sommets  $S = \{s_1, s_2 \dots s_k\}$  dans un graphe  $G = (V, E)$  consiste à supprimer dans  $V$  les sommets de  $S$ , et à ajouter un nouveau sommet  $s$ . Chaque arête du type  $\{a, s_i\} | a \in V - S, s_i \in S$  est supprimée de  $E$ , et est remplacée par l'arête  $\{a, s\}$ . Les arêtes  $\{s_i, s_j\} | s_i, s_j \in S$  sont simplement supprimées.



**Proposition 2 :**

Lorsque tous les nœuds du réseau sont des clients :

1. *ECF* reste NP-complet.
2. *ICECF* est polynomial.

**Preuve :**

Nous prouvons le point 1 de la remarque par une réduction au problème BIPARTITION.

Le problème *ECF* est dans NP : pour toute solution potentielle de coût  $k$  décrire par un ensemble d'arêtes  $E_S$ , on peut vérifier en un temps polynomial que le flot est réalisable sur le graphe partiel induit par les arêtes de  $E_S$ . Nous construisons à partir de toute instance  $N = \{n_1, n_2, n_3, \dots, n_m\}$  du problème BIPARTITION une instance  $I_{ECF} = (G, capa, S, T, dem, k)$  de *ECF* comme suit : Le graphe  $G = (V, E)$  est défini par les ensembles :

$$V = \{s\} \cup N \cup \{N_1, N_2\}$$

et

$$E = big\{\{s, n_i\}, \{n_i, N_1\}, \{n_i, N_2\} | n_i \in N\}$$

Pour tout  $n_i \in N$ , on pose  $capa(\{s, n_i\}) = 2 \times n_i$ ,  $capa(\{n_i, N_1\}) = n_i$  et  $capa(\{n_i, N_2\}) = n_i$ . L'instance ne comporte qu'une source  $S = \{s\}$ , et  $T = V$ . La demande de chaque nœud  $n_i$  est égale à  $n_i$ . Nous posons également  $dem(N_1) = dem(N_2) = \frac{1}{2} \times \sum_{n_i \in N} n_i$ . Il convient dans notre réduction que tous les sommets soient des clients. Aussi la demande du nœud source  $s$  doit être non nulle, mais sa valeur n'est pas pertinente puisqu'auto-satisfaite. Enfin nous posons  $k = 2 \times |N|$ .

Nous affirmons que s'il existe une solution  $S$  à l'instance  $I_{ECF}$ , alors elle est nécessairement de coût égal à  $k$  et permet de déduire une solution à l'instance  $N$  du problème BIPARTITION. Le coût de  $S$  est ici le nombre d'arêtes de  $G$  transitant un flot de valeur non nulle. Dans toute solution réalisable, chacune des arêtes de la forme  $\{s, n_i\}$  doit obligatoirement faire transiter un flot égal à sa capacité. Ces arêtes sont au nombre de  $|N|$ . Pour chaque arête  $\{s, n_i\}$  et sur les  $2 \times n_i$  unités de flot qui transitent par elle, la moitié est consommée par le sommet  $n_i$ . S'il est possible de satisfaire les demandes de  $N_1$  et  $N_2$  avec  $k = 2 \times |N|$  arêtes, alors le flot non-consommé par chaque sommet  $n_i$  ne se divise pas, et est intégralement routé en direction de  $N_1$  ou  $N_2$ . On construit alors une bipartition  $N = \{N_A, N_B\}$  en ajoutant à  $N_A$  les éléments  $n_i$  de  $N$  pour lesquels l'arête  $\{n_i, N_1\}$  fait transiter un flot non nul (précisément de  $n_i$  unités), et à  $N_B$  sinon. La réciproque est immédiate.

Le point 2 de la proposition 2 est immédiat. Soit une instance  $I$  de *ICMECF*. S'il n'y a qu'une source  $s$ , n'importe quel arbre couvrant le graphe en entrée décrit une solution pour  $I$ . S'il y a plusieurs sources, nous les fusionnons pareillement à la preuve du point 1 de la proposition 1. Nous obtenons une instance  $I'$  dont la résolution en temps polynomial donne une solution de même coût pour  $I$ .

□

## 2.2 NetworkDesign avec un seul type de contraintes

Nous étudions la complexité de min-NetworkDesign lorsque seulement un type de contraintes est envisagé. Nous déterminons lorsqu'elle existe la limite entre la polynomialité et la NP-complétude du problème en fonction de la valeur du paramètre régissant la contrainte.

### 2.2.1 Seul le degré est borné

Le problème  $[deg, \infty, 0]$ -min-NetworkDesign est NP-difficile en général, puisqu'il constitue une généralisation du problème MECF présenté précédemment.

La première remarque énonce quelques résultats pour des instances particulières. La seconde remarque conclut sur la limite entre la polynomialité et la NP-complétude de  $[deg, \infty, 0]$ -NetworkDesign en fonction de la valeur de  $deg$ .

#### Proposition 3 :

*Considérons le problème  $[deg, \infty, 0]$ -min-NetworkDesign réduit aux instances où les arêtes sont de capacités infinies, et le nombre de sources de 1. Alors :*

1. *ce problème est NP-difficile,*
2. *il reste NP-difficile même en supposant que tous les nœuds sont des clients,*
3. *il est polynomial s'il n'y a qu'un client.*
4. *les trois premiers points vrais exacts lorsque l'instance possède plusieurs sources.*

#### Preuve :

1. Le problème est exactement le problème DEGREE-CONSTRAINED MINIMUM STEINER TREE (DCMStT) [BV95, Bau96], qui consiste à trouver un arbre de Steiner avec un nombre minimum d'arêtes et dont le degré des nœuds est borné. Ce problème est connu pour être NP-difficile.
2. Lorsque tous les nœuds sont des clients, l'objectif consiste à trouver un arbre couvrant de degré borné par  $deg$ . Déterminer l'existence d'un tel arbre correspond au problème DEGREE-CONSTRAINED SPANNING TREE (DCSpT) [GJ79], connu pour être NP-complet pour tout  $deg \geq 2$  (voir preuve de la proposition 4).
3. Lorsqu'il n'y a qu'un seul client, il suffit de rechercher un plus court chemin de la source vers le client.
4. S'il y a plusieurs sources dans une instance  $I$ , nous les fusionnons par paires à la preuve du point 1 de la proposition 1. Nous obtenons une instance  $I'$  construite en temps polynomial dont la résolution donne une solution de même coût sur  $I$ .

□

**Proposition 4 :**

Le problème de décision  $[deg, \infty, 0]$ -NetworkDesign est :

- polynomial lorsque  $deg = 1$ ,
- NP-complet pour tout  $hop \geq 2$ .

**Preuve :**

Soit  $I = (G, capa, S, T, dem, k)$  une instance de  $[deg, \infty, 0]$ -NetworkDesign.

Lorsque  $deg = 1$ , le problème consiste à trouver pour chaque client  $c_i$  une source dédiée  $s_j$  dans son voisinage, telle que la capacité de l'arête  $\{c_i, s_j\}$  soit supérieure à la demande  $dem(c_i)$ . Sa résolution consiste à trouver un couplage saturant  $T$  dans le graphe biparti  $H = (S, T, \{x, y\} | x \in S, y \in T, \{x, y\} \in E(G), capa(\{x, y\}) \geq dem(y))$ .

Comme annoncé dans la preuve de la proposition 3 et dans les instances avec une seule source, où tous les nœuds sont des clients et les arêtes sont de capacité infinie, le problème est exactement celui du DCSpT. En fait lorsque  $deg = 2$ , il s'agit du problème HAMILTONIAN PATH (HP).

Il existe également une réduction vers problème HP qui prouve la NP-complétude pour toute valeur  $deg \geq 2$  : Depuis  $G$  une instance de HP. On construit une instance  $(H, capa, S, T, dem, k)$  de  $[deg, \infty, 0]$ -NetworkDesign telle que :

- $V(H) = V(G) \cup \{(x, i)\}_{x \in V(G), i \in [1, deg-2]}$  et  $E(H) = E(G) \cup \{(x, (x, i))\}_{x \in V(G), i \in [1, deg-2]}$ ,
- $S = \{s\}$ , où  $s$  est un sommet quelconque de  $V(G)$ , et  $T = \{(x, i)\}_{x \in V(G), i \in [1, deg-2]}$ ,
- $dem(t) = 1 | t \in T, capa(e) = \infty | e \in E(H)$ ,
- $k \geq |V(H)|$ .

S'il existe une solution sur l'instance  $(H, S, T, dem, capa, k)$ , alors l'ensemble des arêtes de la forme  $\{(x, (x, i)) | x \in V(G), i \in [1, deg-2]\}$  appartient à la solution. Le degré des sommets de  $H$  ne peut excéder  $deg$  dans la solution. Les sommets de  $V(G)$  sont alors connectés entre eux dans la solution par un ensemble d'arêtes formant un chemin hamiltonien sur  $G$ .

□

**2.2.2 Seul le nombre de sauts est borné**

Le problème  $[\infty, hop, 0]$ -NetworkDesign s'intéresse à la recherche d'un flot satisfaisant un ensemble de demandes, sous la contrainte que le flot émis depuis chaque source n'effectue pas plus de  $hop$  sauts pour atteindre sa destination. L'objectif reste la minimisation du nombre d'arêtes mobilisées.

Le théorème 1 regroupe les principaux résultats de complexité de ce problème :

**Théorème 1 :**

Le problème  $[\infty, \text{hop}, 0]$ -NetworkDesign est :

1. *polynomial* lorsque  $\text{hop} = 1$ ,
2. *NP-complet* pour tout  $\text{hop} \geq 2$ .

**Preuve :**

Posons  $\text{hop} = 1$ . Soit  $I = (G, \text{capa}, S, T, \text{dem}, k)$  une instance du problème  $[\infty, \text{hop}, 0]$ -NetworkDesign.

Lorsque le nombre de sauts autorisé  $\text{hop}$  est d'au plus 1, une solution optimale en nombre d'arêtes se calcule polynomialement comme suit : pour chaque client  $t \in T$ , nous sélectionnons un ensemble  $E_t$  d'arêtes incidentes à  $t$  et à n'importe quel sommet source assurant à  $t$  la distribution du flot. L'ensemble  $E_t$  de cardinalité minimale se calcule en sélectionnant d'abord les arêtes  $\{t, s\} | s \in S$  de plus grande capacité, et en les ajoutant successivement à  $E_t$  tant que la somme des capacités des arêtes de  $E_t$  reste inférieure à la demande de  $t$ . Les arêtes du graphe solution sont alors l'union des  $E_t | t \in T$ . La distribution du flot est évidente : il suffit de vérifier que la solution possède un coût inférieur ou égal à  $k$ . Ceci conclut la démonstration du point 1 du théorème 1<sup>2</sup>.

Lorsque  $\text{hop} \geq 2$  nous proposons une réduction au problème NP-complet bien connu SET-COVER (SC) :

Le problème  $[\infty, \text{hop}, 0]$ -NetworkDesign est dans NP : soit  $(G, \text{capa}, S, T, \text{dem}, k)$  une instance de ce problème. Compte tenu d'un ensemble de chaînes  $C$  de taille au plus  $\text{hop}$  et d'un flot chaînes-contraint  $f : C \rightarrow R^+$ , il est possible de vérifier en un temps polynomial que  $(C, f)$  est bien une solution pour l'instance  $(G, \text{capa}, S, T, \text{dem}, k)$ .

Soit une instance  $I_{SC} = (X, T, k)$  une instance de SC, où  $T$  est un ensemble de  $n$  éléments  $T = \{t_1, t_2, \dots, t_n\}$ , et  $X = \{X_1, X_2, \dots, X_m\}$  est composé de  $m$  sous-ensembles de  $T$ . Nous construisons depuis  $I_{SC}$  une instance  $I$  de  $[\infty, \text{hop}, 0]$ -NetworkDesign comme suit :

Nous définissons le graphe  $G = (V, E)$  tel que :

$$V = \{X_i\}_{0 \leq i \leq m} \cup \{s_i\}_{0 \leq i \leq \text{hop}-2}$$

et

$$E = \{\{X_i, t_j\} | t_j \in X_i\}_{0 \leq i \leq m, 0 \leq j \leq n} \cup \{s_i, s_{i+1}\}_{0 \leq i \leq \text{hop}-3}$$

Nous posons  $S = \{s_0\}$ ,  $\text{dem} : T \rightarrow R^+$  une fonction quelconque, et  $\text{capa} : E \rightarrow \infty$  la fonction qui associe une capacité infinie aux arêtes de  $G$ . Soit  $I = (G, \text{capa}, S, T, \text{dem}, k)$  une instance de  $[\infty, \text{hop}, 0]$ -NetworkDesign. La construction de  $I$  est clairement polynomiale en fonction de la taille de  $I_{SC}$ . La figure 2.1(b) présente l'instance  $I$  construite à partir de l'instance de SC de la figure 2.1(a), avec  $\text{hop} = 4$ .

---

<sup>2</sup>Cet algorithme est repris de manière plus détaillée dans la preuve du théorème 5, avec l'ajout de la contrainte de robustesse.

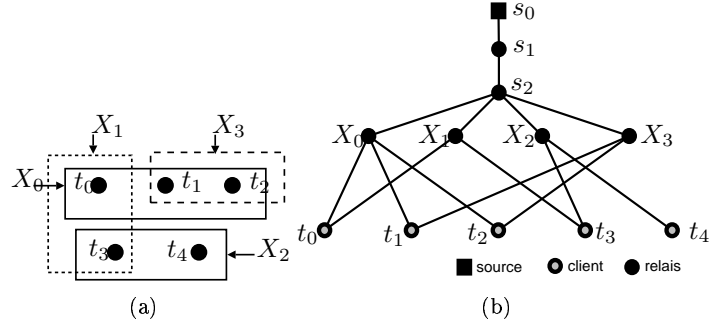


FIG. 2.1 – Construction d’une instance de  $[\infty, \text{hop}, 0]$ -NetworkDesign depuis une instance de SET-COVER et un entier  $\text{rob} = 4$

Considérons une solution  $(C, f)$  de coût  $a \leq k$  sur l’instance  $I$ , où  $C$  est l’ensemble des chaînes  $c$  reliant  $s_0$  clients de  $T$ , et telles que  $f(c) > 0$ . La construction proposée implique que toute chaîne  $c \in C$  est de taille  $\text{hop}$ . Appelons  $U$  l’ensemble des arêtes utilisées par les chaînes de  $C$ . Par définition  $|U| = a$ , il y a au moins  $n$  arêtes incidentes aux sommets de  $T$ , et au plus  $a - n$  arêtes relient  $s_0$  aux éléments de  $X$ . Soit  $W \subseteq U$  l’ensemble des arêtes de la forme  $\{s_{\text{hop}-2}, X_i\} | X_i \in X$ . Alors  $|W| \leq a - n - (\text{rob} - 2)$ . En sélectionnant les éléments de  $X$  incidents à une arête de  $W$ , nous obtenons une solution à l’instance  $I_{SC}$  de coût au plus  $a - n - (\text{rob} - 2)$ .

Réciproquement, si  $Y \subseteq X$  est une solution à  $I_{SC}$  de coût  $b$ , alors on peut en déduire une solution de coût  $b + n + (\text{hop} - 2)$  à  $I$  : nous posons  $U$  l’ensemble des arêtes  $\{s, X_i\} | X_i \in Y$ , et pour chaque client  $t_i$  nous ajoutons à  $U$  une arête  $\{t_i, X_j\} | X_j \in Y$ . Pour chaque client, il existe donc une seule chaîne de taille  $\text{hop}$  reliant tout client à a source, et que nous sélectionnons dans notre solution. Les capacités des arêtes sont infinies, aussi un seule chaîne suffit pour satisfaire une demande. Les arêtes de la forme  $\{s_i, s_{i+1}\} | 0 \leq i \leq \text{hop} - 3$  appartiennent nécessairement à la solution. La cardinalité de  $U$  est clairement de  $b + n + (\text{hop} - 2)$ .

Rappelons enfin que SC est NP-complet en général, ce qui conclut la démonstration du point 2. □

**Corollaire 1 :**

*La réduction de la preuve du théorème 1 conserve le rapport d’approximation entre les solutions. La version optimisation de SET-COVER, nommée MINIMUM-SET-COVER, est bien connue pour être NP-difficile et non approximable à un facteur constant près. Ainsi  $[\infty, \text{hop}, 0]$ -min-NetworkDesign est un problème NP-difficile et non approximable à un facteur constant près.*

**Corollaire 2 :**

Le problème  $[\infty, \text{hop}, 0]$ -NetworkDesign reste NP-complet pour tout  $\text{hop} \geq 3$ , même lorsque l'instance ne possède qu'une source et une destination.

**Preuve :**

Nous reprenons l'instance  $I$  construite dans la démonstration du point 2 du théorème précédent. Nous ajoutons un super client, par une construction déjà présentée dans la preuve de la proposition 1. Nous obtenons une instance  $I'$  n'ayant qu'une source  $s_0$  et un client  $t$ , et dans laquelle la distance entre ces deux nœuds a été augmentée de 1. Nous affirmons que résoudre  $[\infty, \text{hop} + 1, 0]$ -NetworkDesign sur l'instance  $I'$  est aussi difficile que résoudre  $[\infty, \text{hop}, 0]$ -NetworkDesign sur  $I$  : s'il existe une solution  $S'$  à  $I'$  de coût  $k$ , alors elle mobilise chacune des  $|T|$  arêtes ajoutées lors de l'ajout du super client, et ce à pleine capacité. On peut en déduire une solution à  $I$  de coût  $k - |T|$  en sélectionnant les arêtes de  $S'$  non incidentes au super client. La réciproque est immédiate. On en conclut la NP-complétude du problème lorsque le nombre de sauts maximum est 3, et que l'instance ne contient qu'un client.

La figure 2.2.2 présente l'ajout d'un super client  $t$  à partir de l'instance de la figure 2.1(b). Les arêtes  $\{c, t_i\}$  sont de capacité égale à la demande de  $t_i$  dans  $I$ .

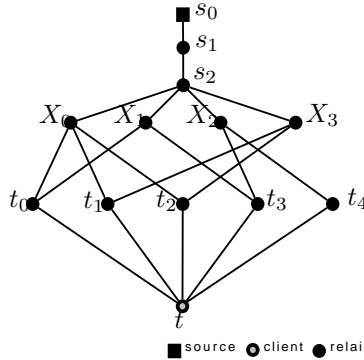


FIG. 2.2 – Ajout d'un super client  $t$  à la construction de la figure 2.1(b)

□

La preuve du dernier théorème annonce que le problème est NP-complet pour  $\text{hop} \geq 2$  même lorsque l'instance ne comporte qu'une source. Le corollaire de ce théorème prouve la NP-complétude pour  $\text{hop} \geq 3$  lorsqu'il n'y a qu'un client. La construction présentée dans la preuve du point 1 de la proposition 2 permet d'établir que le problème reste NP-complet lorsque tous les nœuds sont des clients. Signalons enfin que lorsque les arêtes sont de capacité infinie, et que tous les nœuds sont clients, le problème est exactement celui du MINIMUM DIAMETER SPANNING TREE (MDSpT), bien connu pour être polynomial lorsque le coût des arêtes est uniforme.

### 2.2.3 Seule la robustesse est imposée.

Même lorsque  $r = 0$ , le problème  $[\infty, \infty, r]$ -min-NetworkDesign est NP-difficile, puisqu'il décrit exactement le problème min-NetworkDesign sans contraintes, déjà étudié en 2.1. La variation de ce paramètre n'a qu'une influence limitée sur la complexité du problème. Le nombre de sources (au moins égal à la robustesse du réseau) et de clients ne change en rien la complexité du problème, puisque la preuve de NP-complétude présentée dans la section 2.1 s'applique pour des instances ne comportant qu'une source et un client.

Nous citerons quelques résultats connus lorsque la capacité des arêtes est infinie. En pareil cas, une solution consiste à définir un sous-graphe du graphe de départ dans lequel chaque client peut être connecté à l'aide de  $r + 1$  chaînes sommet-disjoints à  $r + 1$  sources différentes. Une solution optimale requiert alors un nombre d'arêtes minimal. Le problème de l'existence même d'une solution (sans qu'elle soit nécessairement optimale en nombre d'arêtes) s'apparente un peu à celui du UNDIRECTED VERTEX-DISJOINT PATHS (UVDP) consistant à se demander s'il existe des chaînes sommets-disjoints entre  $k$  paires de nœuds sources et clients dans un graphe non orienté. Le problème a été montré NP-complet [Kar75] en général, mais polynomial lorsque  $k$  est borné [RS95] ou si chaque chemin recherché possède une extrémité commune. Nous mentionnerons enfin le théorème énoncé par Menger en 1927, qui a ouvert la voie à de nombreux travaux sur la connexité :

**Définition 3 (un vw-séparateur) :**

*Soient  $v$  et  $w$  deux sommets d'un graphe  $G$ .*

*Un vw-séparateur de  $G$  est un ensemble  $S$  de sommets distincts de  $v$  et  $w$  tel que toute chaîne entre  $v$  et  $w$  passe par un des sommets de  $S$ .*

**Théorème 2 (Menger, 1927) :**

*Le nombre maximal de chaînes sommets-disjoints connectant deux sommets distincts non adjacents  $v$  et  $w$  est égal à la taille minimum d'un vw-séparateur.*

Par une réduction appropriée, ces résultats permettent de définir l'existence ou non de chaînes disjointes dans un graphe entre un client et plusieurs sources pour une robustesse donnée  $r$ . Aucun ne détermine en revanche la solution mobilisant un nombre minimum d'arêtes.

## 2.3 NetworkDesign avec deux types de contraintes

Nous poursuivons l'étude de la complexité de NetworkDesign lorsque deux types de contraintes sont envisagés.

### 2.3.1 Degré et nombre de sauts bornés

Résoudre  $[deg, hop, 0]$ -NetworkDesign est ici encore un problème NP-complet en général. Rappelons tout de même que  $[1, hop, 0]$ -min-NetworkDesign est polynomial pour toute valeur de  $hop$ , en accord avec la proposition 4. Ce problème à

fait l'objet de nombreux travaux sous l'hypothèse que les arêtes sont de capacité infinie. Signalons par exemple qu'il est très voisin du DEGREE-BOUNDED MINIMUM DIAMETER SPANNING TREE (DBMDSpT) [KJL<sup>+</sup>04] lorsque tous les nœuds sont clients, qu'il n'existe qu'une seule source et que les arêtes sont de capacité infinie.

Dans la suite de cette sous-section, les résultats énoncés supposent tous que les arêtes sont de capacité infinie. Malgré cette hypothèse réductrice, nous allons voir que certains résultats de complexité restent encore à trouver.

Les deux théorèmes suivants annoncent la complexité de  $[deg, hop, 0]$ -NetworkDesign pour certaines valeurs de  $deg$  et  $hop$ .

**Théorème 3 :**

*Le problème  $[deg, hop, 0]$ -min-NetworkDesign est polynomial pour tout  $deg \geq 1$ , lorsque  $hop = 1$  et les arêtes sont de capacité infinie.*

**Preuve :**

Soit  $I = (G, capa, S, T, dem)$  une instance de  $[deg, hop, 0]$ -min-NetworkDesign. Le nombre de sauts maximum est de 1, et les capacités des arêtes sont infinies. La demande de débit de chaque client est satisfaite à la condition qu'une arête le relie directement à une source. Toute solution se peut se réduire à trouver un ensemble d'arêtes connectant tout client à une source, de sorte qu'une source ne connecte pas plus de  $deg$  clients. Le coût d'une solution lorsqu'elle existe est toujours de  $|T|$ .

Le graphe  $G = (V, E)$  de l'instance peut se réduire ici à un graphe biparti  $G_B = (S, T, E')$  où  $E' = \{\{x, y\} \in E \mid x \in S, y \in T\}$  puisque seules les arêtes reliant une source et un client sont exploitables.

nous définissons alors le graphe biparti  $H = (X_H, Y_H, E_H)$  tel que :

$$X_H = E', \quad Y_H = T \cup \{(x, i) \mid x \in S, d(x) > deg, i \in [1, d(x) - deg]\}$$

et

$$E_H = \{\{\{x, y\}, y\} \mid x \in S, y \in T\} \cup \{\{x, (x, i)\} \mid x \in S, i \in \mathbb{N}^*\}$$

La figure 2.3(b) présente un graphe biparti  $H$  construit à partir du graphe biparti  $G_B$  de la figure 2.3(a), lorsque le paramètre  $deg$  est fixé à 2.

Soit  $S$  un ensemble d'arêtes formant un couplage maximum dans  $H$ . Si  $S$  sature l'ensemble des sommets de  $Y_H$ , alors on peut en déduire une solution à l'instance  $I$  de  $[deg, 1, \emptyset]$ -min-NetworkDesign : Nous sélectionnons dans  $G_B = (S, T, E)$  les éléments de  $E$  associés à des éléments de  $T$  dans le couplage  $S$  (par exemple dans la figure 2.3(b), si l'arête  $\{\{a, t_3\}, t_3\}$  appartient au couplage, alors  $\{a, t_3\}$  fait partie de la solution de l'instance  $I$  sur la figure 2.3(a)).

Le fait que le couplage sature les sommets du type  $(x, i), x \in S, i \in \mathbb{N}^*$  permet de maintenir le degré de la source  $x$  inférieur ou égal à  $deg$  dans la solution de l'instance  $I$ .  $\square$



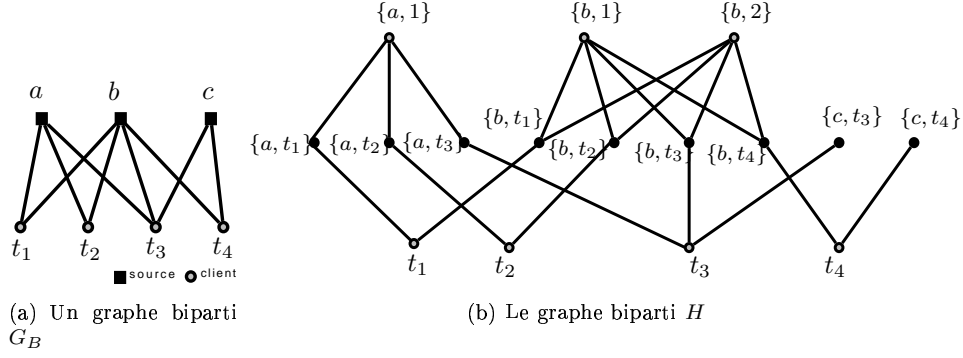


FIG. 2.3 – Construction du biparti  $H$  (b) à partir du biparti  $G_B$  de (a). Ici  $deg = 2$ .

**Théorème 4 :**

Le problème  $[deg, hop, 0]$ -min-NetworkDesign est NP-complet pour tout  $deg \geq 4$  et tout  $hop \geq 2$  lorsque les arêtes sont de capacité infinie.

**Preuve :**

Nous proposons une réduction au problème K-SET-COVER (KSC) qui décrit les instances de SC dans lesquelles la cardinalité de chaque sous-ensemble est d'au plus  $K$ . Le problème est connu pour être NP-complet à partir de  $K = 3$  :

Le problème  $[deg, hop, 0]$ -NetworkDesign est dans NP : soit  $(G, capa, S, T, dem, k)$  une instance de ce problème. Compte tenu d'un ensemble de chaînes  $C$  de taille au plus  $hop$  et d'un flot chaînes-contraint  $f : C \rightarrow R^+$ , il est possible de vérifier en un temps polynomial que  $(C, f)$  est bien une solution pour l'instance  $(G, capa, S, T, dem, k)$  dans laquelle le degré des nœuds est bien inférieur à  $deg$ .

Soit une instance  $I_{KSC} = (X, T, k)$  une instance du problème K-SET-COVER, où  $T$  est un ensemble de  $n$  éléments  $T = \{t_1, t_2, \dots, t_n\}$ , et  $X = \{X_1, X_2, \dots, X_m\}$  est composé de  $m$  sous-ensembles de  $T$ , de cardinalité au plus  $K$ .

Nous construisons depuis  $I_{KSC}$  une instance  $I$  du problème  $[deg, hop, 0]$ -NetworkDesign comme suit :

Nous définissons le graphe  $G = (V, E)$  tel que :

$$V = X \cup T \cup \{s_j^{X_i}\}_{1 \leq i \leq m, 0 \leq j \leq hop-2}$$

et

$$E = \{\{X_i, t_j\} | t_j \in X_i\}_{0 \leq i \leq m, 0 \leq j \leq n} \cup \{s_j^{X_i}, s_{j+1}^{X_i}\}_{1 \leq i \leq m, 0 \leq j \leq hop-3}$$

La construction de  $G$  est polynomiale en fonction de la taille de  $I_{KSC}$ , et le degré dans  $G$  de tous les sommets de  $V - T$  est clairement inférieur ou égal à  $K + 1$ . Nous considérons l'instance  $I = (G, capa, S, T, dem, k)$ , dans laquelle

$S = \{s_0^{X_i}\}_{1 \leq i \leq m}$  et  $\text{capa}(e) = \infty, \forall e \in E$ . La figure 2.4(b) présente l'instance  $I$  construite depuis l'instance de 3-SET-COVER de la figure 2.4(a).

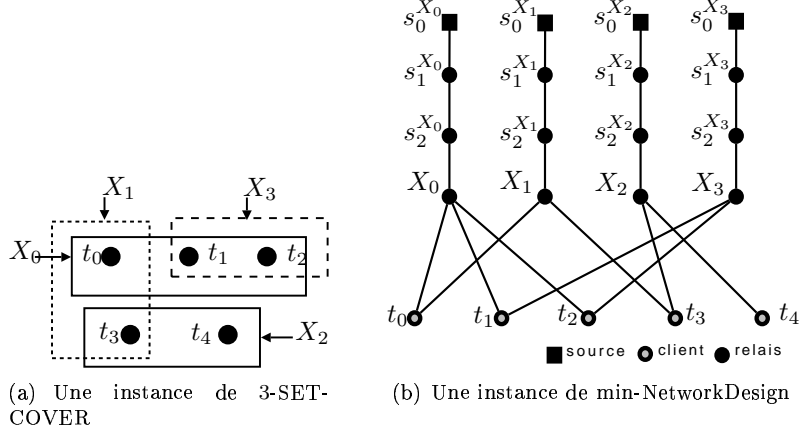


FIG. 2.4 – Construction d'une instance de  $[deg, hop, 0]$ -min-NetworkDesign depuis une instance de 3-SET-COVER

Nous affirmons que s'il existe une solution à l'instance  $I$  sur  $[deg, hop, 0]$ -NetworkDesign, alors il existe une solution de coût proportionnel sur l'instance  $I_{KSC}$  de K-SET-COVER et réciproquement.

Soit  $(C, f)$  une solution de coût  $a$  sur l'instance  $I$ , où  $C$  est l'ensemble des chaînes  $c$  reliant une source à un client de  $T$ , et telles que  $f(c) > 0$ . D'après la construction proposée et les conditions sur les capacités des arêtes, on peut supposer que toute chaîne  $c \in C$  est de taille  $hop$ , et qu'il n'y a qu'une seule chaîne par client (dans le cas contraire on peut se ramener à une solution de coût inférieur). Appelons  $U$  l'ensemble des arêtes utilisées par les chaînes de  $C$ . Alors exactement  $|T|$  arêtes de  $U$  sont incidentes aux sommets clients. Les autres arêtes assurent la connexion entre les différentes sources et les sommets de  $X$ . On peut en déduire une solution de coût  $(a - |T|)/(hop - 2)$  sur l'instance  $I_{KSC}$ , en sélectionnant les ensembles incidents aux arêtes de  $U$ . Rappelons que le degré des clients dans le graphe induit par les arêtes  $U$  est de 1, et que celui des autres sommets est d'au plus  $K + 1$ .

Réciproquement, s'il existe une solution de coût  $b$  pour l'instance  $I_{SC}$ , on en déduit une solution de coût  $(hop - 2) \times b + |T|$  sur  $I$ , en sélectionnant comme nœuds relais les ensembles de la solution, et en assurant la connexité avec la source la plus proche.

Rappelons enfin que le problème est dans NP, et que K-SET-COVER est NP-complet pour  $K \geq 3$ .  $\square$

Le tableau 2.1 récapitule les résultats de cette sous-section, selon les valeurs de  $deg$  et  $hop$ .

À la vue de ce dernier, certains résultats demeurent encore à trouver. On

nombre de sauts ( <i>hop</i> )							
$\geq 6$	P			NP-C	NP-C	NP-C	
5	P			NP-C	NP-C	NP-C	
4	P			NP-C	NP-C	NP-C	
3	P			NP-C	NP-C	NP-C	
2	P			NP-C	NP-C	NP-C	
1	P	P	P	P	P	P	
		1	2	3	4	5	$\geq 6$
		degré ( <i>deg</i> )					

TAB. 2.1 – complexité de  $[deg, hop, 0]$ -NetworkDesign selon le nombre de sauts et le degré maximum autorisés, lorsque les arêtes sont de capacité infinie

peut se demander quelle est la complexité de  $[deg, hop, 0]$ -min-NetworkDesign lorsque  $deg \in [2, 3]$ ,  $hop \geq 2$  et que les arêtes sont de capacité infinie.

Il est par ailleurs admis que  $[deg, hop, 0]$ -min-NetworkDesign avec arêtes de capacité infinie est au moins aussi difficile à résoudre que  $[deg, hop, 0]$ -min-NetworkDesign lorsque les arêtes sont de capacité finie. La NP-complétude du premier problème annonce la NP-complétude du second. En revanche, rien ne permet d'affirmer que  $[deg, hop, 0]$ -min-NetworkDesign est bien polynomial lorsque  $deg \geq 2$  et  $hop = 1$  : le fait qu'un client puisse recevoir un flot depuis plusieurs sources ne permet pas d'étendre la preuve du théorème 3 à cette configuration. La complexité de ce problème est ici ouverte.

### 2.3.2 Degré borné et robustesse imposée

La prochaine configuration étudiée consiste à assurer une  $r$ -robustesse aux pannes lorsque le degré du graphe est borné par  $deg$ . Il ne fait aucun doute de la NP-complétude du problème, même pour  $r = 0$  puisque cette valeur a été considérée par défaut dans les travaux précédents. La remarque suivante est immédiate :

**Remarque 5 :**

*Si  $deg \leq r$  ou si  $|S| \leq r$ , alors  $[deg, \emptyset, r]$ -NetworkDesign est trivialement polynomial puisque sans solution réalisable.*

En accord avec la proposition 4, le problème est polynomial lorsque  $deg = 1$  et  $r = 0$ , et NP-complet pour tout  $deg \geq 2$ . Nous proposons la proposition suivante :

**Proposition 5 :**

*Le problème  $[2, \emptyset, 1]$ -NetworkDesign est NP-complet.*

**Preuve :**

Nous proposons une réduction depuis le problème HAMILTONIAN PATH (HP), connu pour être NP-complet même lorsque les extrémités du chemin recherché sont fixées.

Le problème  $[2, \emptyset, 1]$ -NetworkDesign est dans NP : compte tenu d'une instance de ce problème, on peut vérifier en temps polynomial que toute solution à cette instance assure une robustesse de 1 à chaque client, sans que le degré des nœuds n'excède 2.

Soit  $G$  une instance de HP avec extrémités fixées, ici  $s_1, s_2 \in V(G)$ . Nous construisons une instance  $(H, capa, S, T, dem)$  de  $[2, \emptyset, 1]$ -NetworkDesign dans laquelle :

- $V(H) = V(G) \cup \{z\}$ ,
- $E(H) = E(G) \cup \{\{s_1, z\}, \{s_2, z\}\}$ ,
- $S = \{s_1, s_2\} | s_1, s_2 \in V(G)$ ,
- $T = V(G)$ ,
- et  $capa(e) = \infty | e \in E(G)$ .

Résoudre l'instance de NetworkDesign revient exactement à résoudre l'instance de HP avec extrémités fixées :

Dans toute solution à  $(H, capa, S, T, dem)$ , le degré de chaque client est de 2 exactement (au moins 2 pour assurer la robustesse, au plus 2 par la contrainte de limitation de degré). Soit  $U$  l'ensemble des arêtes utilisées dans toute solution. Alors  $\{\{s_1, z\}, \{s_2, z\}\} \subseteq U$ . De plus, chaque client doit être connecté aux deux sources  $s_1$  et  $s_2$  par deux chemins disjoints. Le degré de ces sources ne peut excéder 2, et les arêtes  $\{s_1, z\}, \{s_2, z\}$  appartiennent à la solution. Il en résulte que les arêtes de  $U - \{\{s_1, z\}, \{s_2, z\}\}$  décrivent un chemin hamiltonien dont les extrémités sont  $s_1$  et  $s_2$ . La réciproque est immédiate.  $\square$

Le tableau 2.2 résume les résultats de complexité de  $[deg, \emptyset, r]$ -NetworkDesign selon la valeur des variables  $deg$  et  $r$ .

**2.3.3 Nombre de sauts borné et robustesse imposée**

Depuis les résultats énoncés dans la section précédente, il est possible de conclure immédiatement à la NP-complétude du problème  $[\infty, hop, r]$ -NetworkDesign lorsque  $hop$  et  $r$  sont quelconques. Les deux théorèmes suivants étudient la frontière entre la polynomialité et la NP-complétude en fonction de ces deux paramètres. Le premier théorème annonce la polynomialité du problème pour tout couple  $(hop, r)$  tel que  $hop = 1$  et  $r \geq 0$ . Le second précise la complexité du problème lorsque  $hop \geq 2$  et  $r \geq 0$ .

**Théorème 5 :**

*Le problème de décision  $[\infty, hop, r]$ -NetworkDesign est polynomial pour tout  $hop = 1$  et  $r \geq 0$ .*

degré ( $deg$ )	0	1	2	3	4	$\geq 5$
$\geq 6$	NP-C					
5	NP-C					P
4	NP-C				P	P
3	NP-C			P	P	P
2	NP-C	NP-C	P	P	P	P
1	P	P	P	P	P	P

TAB. 2.2 – complexité de  $[deg, \emptyset, r]$ -NetworkDesign selon la robustesse minimale et le degré maximum autorisé.

**Preuve :**

La preuve est une extension de la démonstration du point 1 du théorème 1.

Nous proposons un algorithme polynomial qui, depuis une instance  $I$  du problème  $[\infty, hop, r]$ -NetworkDesign, renvoie une solution avec un nombre minimum d'arêtes lorsque  $hop = 1$ .

Le nombre de sauts maximum étant 1, le graphe de l'instance peut se limiter au graphe biparti  $G_B$ , sous-graphe de  $G$  induit par les sommets  $S \cup T$ . Notons que dans une solution, chaque arête sélectionnée est alors dédiée à un couple  $(s_i, t_j)$ ,  $s_i \in S, t_j \in T$  et ne fera passer de flux qu'à destination de  $t_j$ . Les deux conditions sont nécessaires et suffisantes pour obtenir une solution à  $I$  :

1. chaque client  $t \in T$  doit être incident à un ensemble d'arêtes  $E_s(t)$  tel que  $\sum_{e \in E_s(t)} capa(e) \geq dem(t)$ ,
2. l'ensemble  $E_s(t)$  doit être de cardinalité au moins  $r$  pour chaque client  $t \in T$ .

Pour chaque client  $t$ , nous définissons un ensemble  $E^*(t)$  de cardinalité minimum comme suit :

- nous ajoutons successivement à  $E^*(t)$  les arêtes  $\{t, s_i\} | s_i \in S$  en privilégiant celles de plus grande capacité, et tant que la somme des capacités des arêtes de  $E_t$  reste inférieure à la demande de  $t$ .
- Si  $|E^*(t)| < r$ , nous ajoutons des arêtes incidentes à  $t$  jusqu'à ce que  $|E^*(t)| = r$ .

Appelons  $E^* = \bigcup_{t \in T} E^*(t)$ . Soit  $(C, f)$ , avec  $C$  l'ensemble des chaînes  $c$  de longueur 1 telles que  $E(c) \in E^*$ , et  $f : C \rightarrow R$  une fonction telle que  $f(c) = capa(E(c))$ . Alors  $(C, f)$  constitue une solution de coût minimum pour  $I$ .

□

**Théorème 6 :**

Le problème de décision  $[\infty, \text{hop}, r]$ -NetworkDesign est NP-complet pour tout  $\text{hop} \geq 2$  et  $r \geq 0$ .

**Preuve :**

Nous montrons que  $[\infty, \text{hop}, r]$ -NetworkDesign est NP-complet, par une réduction au problème de décision bien connu SET COVER (SC), pour tout  $\text{hop} \geq 2$  et  $r \geq 0$ .

Le problème est dans NP : soit  $(G, \text{capa}, S, T, \text{dem}, k)$  une instance de ce problème. Compte tenu d'un ensemble de chaînes  $C$  et d'un flot chaînes-contraint  $f : C \rightarrow R^+$ , il est possible de vérifier en un temps polynomial que  $(C, f)$  est bien une solution pour l'instance  $(G, \text{capa}, S, T, \text{dem}, k)$ .

Soit une instance  $I_{SC} = (X, T, k)$  une instance de SC, où  $T$  est un ensemble de  $n$  éléments  $T = \{t_1, t_2, \dots, t_n\}$ , et  $X = \{X_1, X_2, \dots, X_m\}$  est composé de  $m$  sous-ensembles de  $T$ . Nous construisons depuis  $I_{SC}$  une instance  $I$  de  $[\infty, \text{hop}, r]$ -NetworkDesign comme suit :

Nous construisons  $r + 1$  copies distinctes de  $X$  nommées  $X^0, X^1, \dots, X^r$ , en posant  $\forall 0 \leq i \leq r, X^i = \{X_1^i, X_2^i, \dots, X_m^i\}$ , où  $X_k^i$  est la copie de  $X_k$  dans  $X^i$ . Notons  $s_i^j = (j, i), \forall (j, i) \in [0, r] \times [0, \text{hop} - 2]$ .

Nous définissons le graphe  $G = (V, E)$  tel que :

$$V = \{X_i^j\}_{0 \leq j \leq r, 0 \leq i \leq m} \cup \{s_i^j\}_{0 \leq j \leq r, 0 \leq i \leq \text{hop} - 2}$$

et

$$E = \{\{X_i^j, t_k\} | t_k \in X_i\}_{0 \leq j \leq r, 0 \leq i \leq m, 0 \leq k \leq n} \cup \{s_i^j, s_{i+1}^j\}_{0 \leq j \leq r, 0 \leq i \leq \text{hop} - 3}$$

Soient  $S = \{s_0^j\}_{0 \leq j \leq r}$ ,  $\text{dem} : T \rightarrow R^+$  une fonction quelconque, et  $\text{capa} : E \rightarrow \infty$  la fonction qui associe une capacité infinie aux arêtes de  $G$ . Posons  $I = (G, \text{capa}, S, T, \text{dem}, l)$  une instance au problème  $[\infty, \text{hop}, r]$ -NetworkDesign. La construction de  $I$  est clairement polynomiale en fonction de la taille de  $I_{SC}$ . La figure 2.3.3 présente l'instance  $I$  construite à partir de l'instance SC de la figure 2.5 et des paramètres  $\text{hop} = 4$  et  $r = 2$ . L'instance comporte  $r + 1$  sources auxquelles tout client doit être connecté.

Considérons une solution  $(C, f)$  de coût inférieur ou égal à  $l$  sur l'instance  $I$ , où  $C$  est l'ensemble des chaînes  $c$  reliant une source et un client, telles que  $f(c) > 0$ . La construction proposée implique que toute chaîne  $c \in C$  est de taille  $\text{hop}$ , et que deux chaînes contenant 2 sources différentes ne partagent aucune arête.

Pour une source  $s_0^i$  donnée, appelons  $p_i = |\bigcup E(\{c\}), c \in C | s_0^i \in \text{ext}(c)|$  le nombre d'arêtes utilisées pour connecter  $s_0^i$  à l'ensemble des clients  $T$ . Soit  $p = \min_{0 \leq i \leq r} p_i$ . Alors,  $p$  arêtes suffisent pour connecter n'importe quelle source à  $T$  (évident par construction), et on peut se ramener à une solution de coût  $a$  avec  $a = (r + 1) \times p$ .

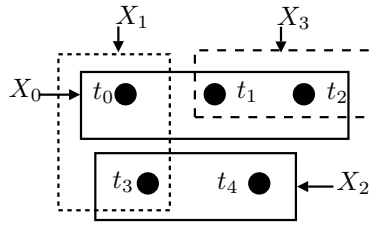


FIG. 2.5 – Une instance de SET-COVER

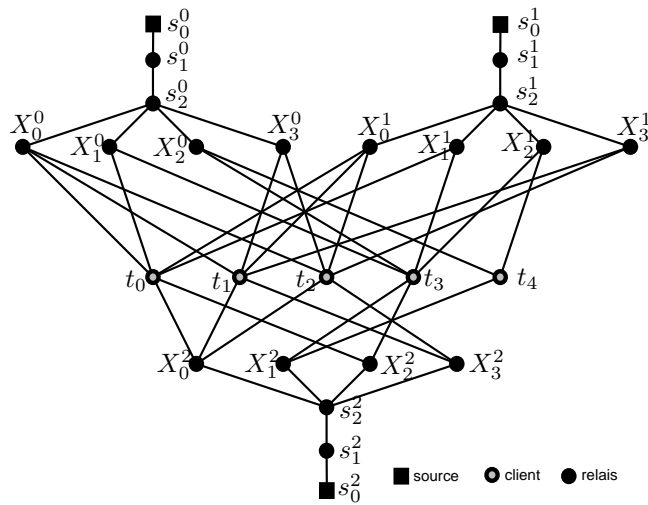


FIG. 2.6 – Une instance construite à partir de l'instance SET-COVER de la figure 2.5 et des paramètres  $hop = 4$  et  $r = 2$ .

Considérons alors une source  $s_0^i$ , et l'ensemble des arêtes  $U = \bigcup E(\{c\}), c \in C | s_0^i \in ext(c)$  la connectant à  $T$ , et notons  $p = |U|$ . Alors exactement  $p - |T| - (hop - 2)$  sommets de  $X_i$  sont utilisés pour connecter  $s_0^i$  à  $T$ . Ces sommets sont les ensembles de  $X^i \cap \{ \bigcup V(\{c\}), c \in C | s_0^i \in ext(c) \}$ , qui définissent alors une solution au problème SC sur  $I_{SC}$  de coût  $p - |T| - (hop - 2)$ , encore égal à  $a/(r + 1) - |T| - (hop - 2)$ .

Réciproquement, soit  $X^S \subseteq X$  une solution sur l'instance  $I_{SC}$  de coût  $b = |X^S| \leq k$ . On peut raisonnablement penser que  $X^S$  est une couverture minimale, c'est à dire qu'il existe pour chaque ensemble  $X_i$  de  $X^S$  un élément de  $T$  qui n'appartient à aucun autre ensemble de  $X^S$ . Pour chaque couple  $(s_0^i, t_j), 0 \leq i \leq r, 0 \leq j \leq n$ , nous considérons la chaîne  $c$  telle que  $V(c) = \{s_0^i, s_1^i, \dots, s_{hop-2}^i, X_\alpha^i, t_j\}$  avec  $X_\alpha^i \in X^S$  et  $t_j \in X_\alpha^i$ , et posons  $f(c) = \infty$ . Soit  $C$  l'union de ces chaînes, et  $H$  le graphe pour lequel  $V_H = \bigcup_{c \in C} V(c)$  et  $E_H = \bigcup_{c \in C} E(c)$ . Alors  $(C, f)$  est bien une solution de  $I$ , puisque  $C$  assure la robustesse de  $T$  dans  $(G, S)$ , et que les chaînes de  $C$  sont de longueur  $hop$ . Le nombre d'arêtes de  $H$  est alors de  $((hop - 2) + b + |T|) \times (r + 1)$ .

Ainsi pour toute solution sur  $I$  de coût  $a$ , on peut déduire une solution sur  $I_{SC}$  de coût  $b$ , où  $b = a/(r + 1) - |T| - (hop - 2)$ , et réciproquement. Rappelons enfin que SC est un problème NP-complet.  $\square$

**Corollaire 3 :**

*La réduction présentée conserve le rapport d'approximation. Rappelons que MSC est bien connu pour être NP-difficile et non approximable à un facteur constant près. Ainsi  $[\infty, hop, r]$ -min-NetworkDesign est un problème NP-difficile et non approximable à un facteur constant près pour tout  $hop \geq 2$  et  $r \geq 0$ .*

Le tableau 2.3 résume les résultats présentés dans cette sous-section.

nombre de sauts ( $hop$ )							
$\geq 6$		NP-C	NP-C	NP-C	NP-C	NP-C	NP-C
5		NP-C	NP-C	NP-C	NP-C	NP-C	NP-C
4		NP-C	NP-C	NP-C	NP-C	NP-C	NP-C
3		NP-C	NP-C	NP-C	NP-C	NP-C	NP-C
2		NP-C	NP-C	NP-C	NP-C	NP-C	NP-C
1		P	P	P	P	P	P
		0	1	2	3	4	$\geq 5$
		robustesse ( $r$ )					

TAB. 2.3 – Complexité de  $[\infty, hop, r]$ -NetworkDesign en fonction de  $hop$  et  $r$



# Bibliographie

- [Bau96] F. Bauer. Multicast routing in point-to-point networks under constraints, 1996.
- [BV95] F. Bauer and A. Varma. Degree-constrained multicasting in point-to-point networks, 1995.
- [EKS05] Guy Even, Guy Kortsarz, and Wolfgang Slany. On network design problems : fixed cost flows and the covering steiner problem. *ACM Trans. Algorithms*, 1(1) :74–101, 2005.
- [GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [Kar75] Richard M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5 :45–68, 1975.
- [KJL<sup>+</sup>04] Konemann, Jochen, Levin, Asaf, Sinha, and Amitabh. Approximating the degree-bounded minimum diameter spanning tree problem. *Algorithmica*, 41(2) :117–129, November 2004.
- [RS95] Neil Robertson and P. D. Seymour. Graph minors. xiii : the disjoint paths problem. *J. Comb. Theory Ser. B*, 63(1) :65–110, 1995.