

Sélection quasi-optimale de routes pour la surveillance de réseaux à l'aide de la génération de colonnes

Alice Burlats, Cristel Pelsser, Pierre Schaus

UCLouvain, ICTeam

2025

Résumé

Un système de gestion de réseau robuste doit permettre de détecter et de diagnostiquer rapidement les pannes. Avec la tomographie booléenne, la surveillance d'un ensemble suffisant de routes permet de rendre toute panne détectable et identifiable de manière unique (1-identifiable). Étant donné que ces sondes génèrent du trafic dans le réseau, il est crucial de sonder le plus petit ensemble de routes possible tout en garantissant une supervision efficace. Pour ce problème nous expérimentons une approche basée sur la génération de colonnes pour résoudre les problèmes de mémoire rencontrés par un modèle de programmation linéaire en nombres entiers tout en retournant de meilleures solutions qu'un algorithme glouton.

Mots-clés

Tomographie booléenne, supervision de réseaux, programmation linéaire en nombres entiers, génération de colonnes

1 Introduction

Les pannes dans les réseaux informatiques sont fréquentes. Elles surviennent suite à des erreurs de configurations, des câbles endommagés, des phénomènes naturels ou simplement l'usure. La tomographie booléenne consiste à déterminer l'état de chaque nœud à l'aide de mesures et de règles d'inférence. Lorsqu'un nœud tombe en panne, toutes les routes le traversant tombent également en panne. Une sonde (un paquet ICMP) peut donc être envoyée le long d'une route pour vérifier si elle atteint sa destination, et donc s'assurer que la route, et tous les nœuds qu'elle traverse, est fonctionnelle. En sondant régulièrement un ensemble suffisant de routes de mesure, où chaque nœud est traversé par au moins une route, toute panne dans le réseau peut être détectée.

Au-delà de la détection des défaillances, il est possible de les localiser. L'ensemble des routes de mesures qui traversent un nœud constitue son *symptôme*, ce qui signifie que sa défaillance entraînera la coupure de toutes ces routes. La défaillance d'un seul nœud peut être localisée si chaque nœud présente un symptôme unique. Cette condition est connue sous le nom de 1-identifiabilité.

La Figure 1 présente une petite topologie de réseau. Les styles de lignes et les couleurs représentent les différentes routes de mesure. Dans (1), chaque nœud est traversé par

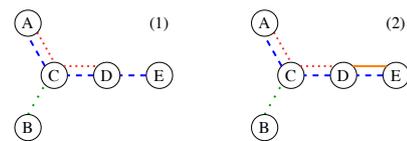


FIGURE 1 – Illustration de la 1-identifiabilité. Avec les routes mises en évidence : (1) tous les nœuds sont couverts mais pas 1-identifiables ; (2) tous les nœuds sont 1-identifiables.

au moins une route de mesure. Ainsi, lorsque l'un d'entre eux rencontre une défaillance, il est possible de la détecter. Mais A et D ont la même signature. Si l'un ou l'autre tombe en panne, dans les deux cas la route liant A à E et A à D ne fonctionne plus. La panne sera détectée mais il sera impossible de déterminer sans ambiguïté sur quel nœud elle se situe. Le réseau est couvert mais les défaillances ne sont pas 1-identifiables. Dans (2), une route liant E à D a été ajoutée. Avec cette route supplémentaire, le réseau est 1-identifiable car chaque nœud est traversé par un ensemble unique de routes de mesure.

Être réactif aux défaillances implique d'envoyer des sondes très fréquemment, ce qui peut perturber le réseau. Ainsi, pour limiter ce problème, il est nécessaire de minimiser le nombre de routes de mesures qui doivent être sondées.

Dans notre étude, nous considérons que nous n'avons pas le contrôle sur la définition des routes dans le réseau : le protocole de routage offre une route entre chaque paire de nœuds. Nous piochons un sous-ensemble de routes de mesure parmi ces routes pour mesurer le réseau.

Nos contributions peuvent être résumées comme suit :

- Nous formulons le problème de 1-identifiabilité par ensembles (**1IE**) (Section 3) ;
- Nous présentons un algorithme glouton et un modèle de programmation linéaire en nombres entiers (PLNE) pour résoudre ce problème (Sections 3.1 et 3.2) ;
- Nous proposons une approche basée sur la génération de colonnes (GC) pour résoudre un modèle utilisant moins de variables (Section 3.3) ;
- Nous montrons que la génération de colonnes est capable de trouver de meilleures solutions que le glouton tout en utilisant moins de 25% des variables du modèle PLNE complet. (Section 4).

2 Travaux antérieurs

Plusieurs approches pour la sélection de routes de mesure dans un réseau ont été explorées. Stanic et al. [11] présentent un algorithme glouton et un modèle PLNE. L'algorithme glouton commence par sélectionner toutes les routes possibles, puis les supprime une par une jusqu'à ce que la suppression d'une route supplémentaire entraîne une rupture de la couverture ou une violation de la propriété de 1-identifiabilité. Cette méthode est assez inefficace en termes de complexité temporelle car le nombre de routes nécessaires à la supervision est faible, comme montré dans la section 4. La conception de leur modèle PLNE est similaire à celle du nôtre. Pepe et al. [8] conçoivent également un algorithme glouton pour garantir la 1-identifiabilité des liens dans le réseau. Leur algorithme produit une solution avec autant de routes que d'éléments à identifier, ce qui est sans doute sous-optimal.

Plusieurs études [1, 2, 7, 9] traitent d'un problème connexe qui implique la sélection de moniteurs à partir des nœuds du réseau. Dans cette approche, tout chemin reliant deux moniteurs sélectionnés est automatiquement considéré comme un chemin de surveillance, ce qui élimine le besoin de choisir explicitement les routes. Cette approche conduit à la surveillance d'un plus grand nombre de routes.

3 Définition et approches

Le problème de 1-identifiabilité par ensemble (IIE) : Soit un univers U et une collection d'ensembles $S = \{S_1, \dots, S_{|S|}\}$, où $\forall S_i \in S, S_i \subseteq U$. Nous notons $S^i \subseteq S$ la collection contenant tous les ensembles contenant l'élément $i, \forall i \in U$. Le but de ce problème est de trouver le sous-ensemble $S^* \subseteq S$ de cardinalité minimale tel que :

- $\forall i \in U, S^i \cap S^* \neq \emptyset$ (couverture);
- $\forall i \in U, S^i \cap S^*$ est unique (1-identifiabilité).

Pour chaque nœud $i \in U$, nous appelons l'intersection $S^i \cap S^*$ son *symptôme*. Le but du IIE est de sélectionner la plus petite collection d'ensembles tel que chaque élément d'un univers dispose d'un symptôme unique.

Nous pouvons modéliser notre problème de sélection de routes comme un IIE : L'univers U représente l'ensemble de nœuds de la topologie, il contient un élément pour chaque nœud présent dans le réseau. Chaque route dans le réseau est représentée comme un ensemble non-ordonné contenant chacun des nœuds traversés par la route. Ainsi, la collection S représente l'ensemble des routes définies par le protocole de routage.

3.1 Algorithme glouton

Nous proposons l'algorithme 1, inspiré de l'algorithme MNMP [7], mais adapté pour travailler sur la sélection d'un ensemble de routes de mesure, plutôt qu'un ensemble de nœuds moniteurs.

Chaque ensemble S_i de la collection des ensembles candidats S est représenté par deux bitsets. Le premier, $\Omega(S_i)$, représente l'ensemble des éléments couverts par S_i . Il s'agit

Algorithm 1 Algorithme glouton

```

1:  $S^* \leftarrow \{\}$  ▷ Ensemble de routes de mesures
2:  $I \leftarrow U$  ▷ Nœuds non couverts
3:  $I' \leftarrow \{(i, j) \in U^2 \mid i < j\}$  ▷ Paires non distinguées
4: while  $I' \neq \emptyset$  do
5:    $S_{\text{choisi}} \leftarrow \arg \max_{S_i \in S \setminus S^*} |I' \cap \Omega(S_i)|$ 
6:    $S^* \leftarrow S^* \cup \{S_{\text{choisi}}\}$ 
7:    $I' \leftarrow I' \setminus \Omega(S_{\text{choisi}})$ 
8:    $I \leftarrow I \setminus \Psi(S_{\text{choisi}})$ 
9: end while
10: if  $I \neq \emptyset$  then ▷ Au plus un nœud non couvert
11:    $S_{\text{choisi}} \leftarrow \arg \max_{S_i \in S \setminus S^*} |I \cap \Psi(S_i)|$ 
12:    $S^* \leftarrow S^* \cup \{S_{\text{choisi}}\}$ 
13: end if
14: return  $S^*$ 

```

d'un bitset de longueur $|U|$, où le bit i' est vrai ssi l'ensemble S_i contient i' . Pour chaque paire d'éléments i', j' , si l'ensemble S_i contient soit i' soit j' mais pas les deux, alors choisir cet ensemble assure que i' et j' auront un symptôme différent. On dit alors que S_i *distingue* i' et j' . Pour chaque $S_i \in S$, $\Omega(S_i)$ représente l'ensemble des paires d'éléments distinguées par S_i . C'est un bitset de longueur $\frac{|U| \cdot (|U| - 1)}{2}$, contenant un bit pour chaque paire d'éléments qu'il est possible de constituer dans U , où le bit lié à la paire $(i'j')$ est vrai ssi l'ensemble S_i permet de les distinguer.

La boucle *while* des lignes 4-9 est exécutée jusqu'à ce que chaque paire d'éléments puisse être distinguée. À chaque itération, l'ensemble permettant de distinguer le plus de paires non distinguées est ajouté. En cas d'égalité, nous sélectionnons au hasard un ensemble parmi les meilleurs. À la fin de cette boucle, il est possible qu'un élément unique ne soit pas couvert. En effet s'il est le seul à avoir un symptôme unique, alors il est distingué de tous les autres nœuds. Dans ce cas, un ensemble couvrant l'élément non couvert est arbitrairement ajouté.

3.2 Formulation PLNE

Nous utilisons un ensemble de variables binaires x . Nous avons une variable x_s par ensemble candidat $S_s \in S$, qui vaut 1 ssi l'ensemble S_s est sélectionné. Pour chaque élément $i \in U$, la collection des ensembles candidats qui couvrent i est représenté par un vecteur binaire $C_i = \{C_{i1}, \dots, C_{i|S|}\}$ de taille $|S|$ où C_{is} vaut 1 ssi l'ensemble S_s contient l'élément i . Pour chaque paire d'éléments $i < j \in U^2$, il existe également un vecteur binaire $D_{(ij)} = \{D_{(ij)1}, \dots, D_{(ij)|S|}\}$ de taille $|S|$ représentant la collection des ensembles candidats qui contiennent soit i , soit j , mais pas les deux. Il contient la collection d'ensembles qui distinguent i et j ; si au moins l'un d'entre eux est sélectionné, les symptomes des deux éléments sont différents.

Le modèle PLNE est composé de plusieurs contraintes. L'équation (1) exprime la fonction objectif, c'est-à-dire le nombre d'ensembles sélectionnés. L'équation (2) garantit que chaque élément est contenu dans au moins un des ensembles sélectionnés, i.e., que chaque élément est couvert. L'équation (3) est utilisée pour la 1-identifiabilité : elle im-

pose que, pour chaque paire d'éléments, il existe au moins un ensemble qui contient l'un d'entre eux et pas l'autre, c'est-à-dire qu'ils aient des symptômes différents.

$$\text{minimiser } \sum_{S_s \in S} x_s \quad (1)$$

tel que :

$$\sum_{S_s \in S} C_{is} x_s \geq 1 \quad \forall i \in U \quad (2)$$

$$\sum_{S_s \in S} D_{(ij)s} x_s \geq 1 \quad \forall (i, j) \in U^2, i < j \quad (3)$$

$$x_s \in \{0, 1\} \quad \forall S_s \in S \quad (4)$$

3.3 Génération de colonnes

Pour résoudre les problèmes de mémoire rencontrés avec le modèle PLNE (voir section 4) sur les grandes instances, nous explorons la GC pour n'utiliser qu'un petit sous-ensemble des variables du modèle original et limiter ainsi la consommation de mémoire.

Problème maître restreint : La GC commence par la résolution d'une relaxation linéaire réduite de notre problème, appelée problème maître restreint (PMR). Notre PMR est :

$$\text{minimiser } \sum_{S_s \in \tilde{S}} x_s \quad (5)$$

tel que :

$$\sum_{S_s \in \tilde{S}} C_{is} x_s \geq 1 \quad \forall i \in U \quad (6)$$

$$\sum_{S_s \in \tilde{S}} D_{(ij)s} x_s \geq 1 \quad \forall (i, j) \in U^2, i < j \quad (7)$$

$$x_s \in [0, 1] \quad \forall S_s \in \tilde{S} \quad (8)$$

où $\tilde{S} \subset S$ est une collection réduite d'ensembles et x est un vecteur de variables continues de taille $|\tilde{S}|$. Comme ce PMR est une relaxation linéaire sur un très petit sous-ensemble de variables, il est plus facile à résoudre pour un solveur de programmation linéaire (PL) que la formulation PLNE d'origine. Mais comme il manque de nombreuses variables, sa solution optimale peut ne pas être optimale pour le problème complet (dans sa version relaxée). Il faut donc prouver qu'il n'existe aucune colonne supplémentaire possible qui pourrait faire partie d'une solution de coût plus faible. Si une telle colonne existe, elle doit être ajoutée à \tilde{S} .

Pour déterminer quelles colonnes doivent être ajoutées pour réduire le coût de la solution, un *problème de tarification* est résolu. Ce dernier se base sur la version duale du problème.

Problème dual : Pour la relaxation linéaire du modèle PLNE, le problème dual est :

$$\text{maximiser } \sum_{i \in U} \pi_i + \sum_{i < j \in U^2} \pi'_{ij} \quad (9)$$

tel que :

$$\sum_{i \in U} \pi_i C_{is} + \sum_{i < j \in U^2} \pi'_{ij} D_{(ij)s} \leq 1 \quad \forall S_s \in S \quad (10)$$

où π est un vecteur de variables duales de taille $|U|$ liées aux contraintes de couverture (une variable pour chaque contrainte de couverture dans le PMR) et π' est un vecteur de variables duales de taille $\frac{|U| \cdot (|U|-1)}{2}$ liées aux contraintes de 1-identifiabilité (une variable pour chaque contrainte imposant la 1-identifiabilité). Ce problème dual nous permet d'en dériver le problème de tarification.

Problème de tarification : L'algorithme du simplexe utilisé pour trouver la solution optimale du PMR fournit également la solution optimale $(\pi, \pi')_{\tilde{S}}$ à son problème dual. Le coût réduit d'un ensemble $S_s \in S \setminus \tilde{S}$ est donné par :

$$1 - \sum_{i \in U} \pi_i C_{is} - \sum_{i < j \in U^2} \pi'_{ij} D_{(ij)s} \quad (11)$$

Si un ensemble S_s présente un coût réduit négatif, cela signifie que la contrainte duale associée à S_s est enfreinte et la solution trouvée $(x)_{\tilde{S}}$ peut donc être non optimale. Pour remédier à cela, S_s doit être ajouté à \tilde{S} . Mais il est important que \tilde{S} soit aussi petit que possible. Par conséquent, il n'est pas efficace d'ajouter tous les ensembles de $S \setminus \tilde{S}$ qui présentent un coût négatif. Au lieu de cela, nous sélectionnons les 10 variables présentant les coûts réduits les plus négatifs. Ce paramètre de 10 a été choisi comme compromis entre la vitesse de convergence et le nombre de colonnes ajoutées à chaque étape.

Initialisation : Pour obtenir les valeurs duales requises pour déterminer quelles colonnes doivent être ajoutées, le PMR initial doit être faisable. Il faut donc initialiser \tilde{S} avec une solution viable. Pour cela, il est initialisé avec la solution renvoyée par l'algorithme glouton 1.

Solution entière : Rappelons que la génération de colonnes, lorsqu'elle atteint son critère d'arrêt, fournit une solution optimale à la relaxation linéaire du problème. Elle peut attribuer des valeurs fractionnaires aux variables binaires et ne garantit pas que l'ensemble des colonnes candidates qui en résulte contienne la solution optimale au problème entier initial. Par conséquent, nous avons besoin d'une étape supplémentaire pour obtenir une solution entière. Pour cela nous utilisons le modèle PLNE sur la collection réduite \tilde{S} obtenue à la fin de la GC. Cette approche ne garantit pas de trouver la solution optimale du problème original mais permet tout de même de retourner des solutions de bonne qualité comme montré dans la section 4.

4 Résultats expérimentaux

Dans cette section, nous évaluons nos différentes approches.¹ Nous utilisons Gurobi 11.0.2 [5] pour résoudre le modèle PLNE et le PRM de la génération de colonnes. Toutes les expériences tournent sur des processeurs AMD Epyc Genoa at CPU 2.4 GHz avec jusqu'à 766 Go de mémoire. La limite de temps est fixée à 30 minutes et chaque solveur peut utiliser jusqu'à 20 GB de mémoire.

Dataset : Pour évaluer nos modèles, nous utilisons des topologies du CAIDA ITDK (topologies IPv4 de Février 2022) [3], Rocketfuel [10] et du Internet Topology Zoo [6].

1. Le code source est disponible à <https://github.com/BurlatsAlice/PathSelection>

Nous supposons que les plus courtes routes sont basées sur le nombre de hops. Pour avoir un ensemble de routes plus réaliste, nous utilisons aussi des topologies provenant de *Repetita* [4], elles contiennent des poids IGP réalistes. Parmi elles, nous retrouvons des évolutions du *Internet Topology Zoo* et de *Rocketfuel*. Ainsi, pour certaines instances nous avons deux versions. Dans ces cas-là nous utilisons celle contenant les poids IGP. Pour les topologies qui ne sont pas connexes, nous gardons seulement la composante la plus grande pour s’assurer que l’instance peut être résolue. Au total, notre dataset contient 365 topologies, avec jusqu’à 960 nœuds et 920640 routes.

Solveur	Meilleure solution trouvée	Optimalité prouvée
PLNE	335 (91.78%)	195 (53.42%)
Glouton	55 (15.07%)	None
GC	262 (71.78%)	None

TABLE 1 – Nombre d’instances résolues pour chaque modèle.

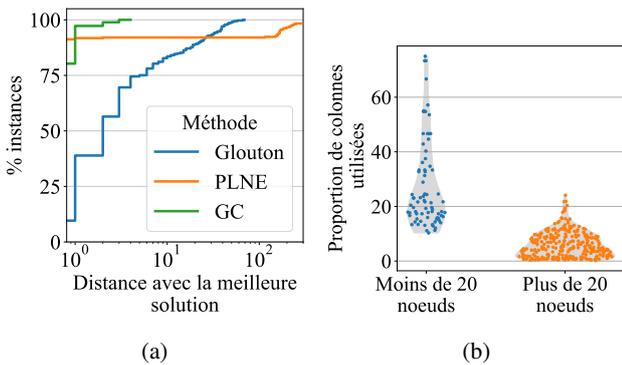


FIGURE 2 – (a) Proportion d’instances résolues par rapport à la distance sur l’objectif avec la meilleure solution trouvée. (b) Distribution de la proportions de routes sélectionnées par rapport au nombre total de routes candidates, selon le nombre de nœuds dans la topologie.

Résultats : La Table 1 montre pour chaque solveur la proportion d’instances pour lesquelles ils renvoient la meilleure solution trouvée, et la proportion d’instances pour lesquelles ils prouvent l’optimalité de leur solution. Comme le glouton et la GC ne sont pas en mesure de prouver l’optimalité de leurs solutions, la dernière valeur est None. La figure 2a montre pour chaque solveur la différence de coût entre la solution du solveur et la meilleure solution trouvée. Les résultats montrent que le modèle PLNE est prometteur : dans la Table 1 nous observons que le glouton est capable de retourner la meilleure solution trouvée pour seulement 15.07% des instances alors que cette proportion est de 91.78% pour le solveur PLNE. Pour 10,68% des instances (39 instances), le glouton renvoie une solution utilisant au moins 10 routes de plus que la meilleure solution trouvée. Ainsi, l’utilisation d’un solveur PLNE exact comme Gurobi réduit généralement le coût de surveillance.

Mais le solveur PLNE manque de mémoire pour 30 instances : parmi elles, la plus petite topologie contient 372 nœuds et 138012 routes, mais la plus grande instance qui est résolue sans problème de mémoire contient 413 nœuds et 170156 routes. Cette limitation de mémoire lui fait retourner des solutions loin d’être optimales (avec jusqu’à 283 routes supplémentaires), comme le montre la figure 2a. Dans nos instances, la plupart des routes ont des contributions similaires. Ainsi, lorsqu’une route est sélectionnée, de nombreuses autres routes voient leurs contributions devenir très limitées et deviennent donc inutiles. Le problème peut donc être résolu en n’utilisant qu’un petit sous-ensemble de variables, et donc en réduisant la mémoire consommée.

La GC permet d’exploiter cette caractéristique de nos instances : La figure 2b montre la distribution de la proportion de colonnes utilisées pour résoudre une instance avec la GC. Nous pouvons observer que, la plupart du temps, moins de 15% des routes sont ajoutées au modèle lors de la GC. Cela montre deux choses : d’une part que la GC est appropriée à ce problème, car la plupart des variables peuvent ne pas être ajoutées au problème ; d’autre part très peu de routes (parmi celles fournies par le protocole de routage) sont nécessaires pour assurer la 1-identifiabilité du réseau. Pour aucune instance le solveur GC ne rencontre de problèmes de mémoire. En ce qui concerne la qualité des solutions obtenues, le tableau 1 montre qu’en utilisant la GC, nous trouvons une solution avec un coût égal à la meilleure solution trouvée pour 71,78% des instances. La GC permet d’améliorer la solution du glouton dans 87,67% des cas. Pour 26,30% des instances, la limite de temps a été atteinte lors de la résolution du modèle PLNE final, ce qui indique que le modèle PLNE réduit reste difficile à résoudre malgré l’ensemble restreint de colonnes. Pour 1,36% des instances, la GC ne termine pas dans le temps imparti et la solution rapportée est celle utilisée pour initialiser le PMR. Au final, la GC retourne des solutions de meilleures qualités que le glouton tout en se préservant des problèmes de mémoire du modèle PLNE. Selon nous, elle représente le meilleur compromis pour résoudre ce problème.

5 Conclusion

Dans ce travail, nous nous sommes concentrés sur la sélection des routes de mesure pour garantir la 1-identifiabilité de chaque nœud dans un réseau. Nous avons introduit le problème de 1-identifiabilité par ensembles pour sélectionner l’ensemble minimal de routes de mesure. Notre comparaison d’une approche de GC avec un modèle PLNE et un algorithme glouton a révélé que la GC évite les limitations de mémoire du modèle PLNE tout en fournissant des solutions supérieures à l’algorithme glouton et des solutions de meilleure qualité que la PLNE sur les instances difficiles et de grande taille. Pour les travaux futurs, le développement d’un algorithme de « branch-and-price » en complément de la GC pourrait être une direction prometteuse pour garantir l’optimalité des solutions trouvées.

Références

- [1] Pamela Bezerra, Po-Yu Chen, Julie A. McCann, and Weiren Yu. Adaptive Monitor Placement for Near Real-time Node Failure Localisation in Wireless Sensor Networks. *ACM Transactions on Sensor Networks*, 18(1) :2 :1–2 :41, October 2021.
- [2] Auguste Burlats, Pierre Schaus, and Cristel Pelsser. An exploration of exact methods for effective network failure detection and diagnosis. In Bistra Dilkina, editor, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 153–169, Cham, 2024. Springer Nature Switzerland.
- [3] CAIDA. The CAIDA Macroscopic Internet Topology Data Kit, February 2022.
- [4] Steven Gay, Pierre Schaus, and Stefano Vissicchio. *Repetita : Repeatable experiments for performance evaluation of traffic-engineering algorithms*, 2017.
- [5] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2023.
- [6] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9) :1765 –1775, october 2011.
- [7] Liang Ma, Ting He, Ananthram Swami, Don Towsley, and Kin K. Leung. On optimal monitor placement for localizing node failures via network tomography. *Performance Evaluation*, 91 :16–37, September 2015.
- [8] Teresa Pepe and Marzio Puleri. Network Tomography : A novel algorithm for probing path selection. In *2015 IEEE International Conference on Communications (ICC)*, pages 5337–5341, June 2015. ISSN : 1938-1883.
- [9] Wei Ren and Wei Dong. Robust network tomography : K-Identifiability and monitor assignment. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.
- [10] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4) :133–145, August 2002.
- [11] Sava Stanic, Suresh Subramaniam, Gokhan Sahin, Hongsik Choi, and Hyeong-Ah Choi. Active monitoring and alarm management for fault localization in transparent all-optical networks. *IEEE Transactions on Network and Service Management*, 7(2) :118–131, June 2010. Conference Name : IEEE Transactions on Network and Service Management.