Inférence des accords économiques et des politiques de routage dans l'Internet

Mickael Meulle

Inférence des accords économiques et des politiques de routage dans l'Internet

Mickael Meulle

12 mars 2007

Thèse présentée pour obtenir le grade de docteur en informatique

Jean-Luc Lutton

Laboratoire CORE/CPN
France Télécom Division R&D
Issy-Les-Moulineaux, France

Philippe Mahey

Laboratoire LIMOS
Université Blaise Pascal
Clermont-Ferrand, France

Dédicacée à

ma mère Elisabeth

Inference of AS business relationships and routing policies in the Internet

Abstract

Internet supports public IP communications. It is composed of over 20 000 smaller networks called Autonomous Systems (AS). ASs belong to various administrative entities : network operators, universities, companies... Each of them has global connectivity to any other, thanks to interdomain routing handled by BGP (Border Gateway Protocol). Bilateral connections between ASs support business agreements to guarrantee Internet connectivity. During the negociation of an economical relationship between two ASs, the two administrative entities agree on network availability, price, bandwith, and ratio of the amount of trafic they send to each other. They may also restrict the set of reachable destinations they share. This economic market of network interconnections is not regulated. Typically, long-term actors take advantage of their market position to negociate their economic relationships, and large players such as content providers or access providers use providers in competition in multiple locations to minimize their cost and maximize their availability through redundancy. In interdomain topology, BGP routes are rather economical paths than shortest AS paths. Economic business agreements between ASs are private but they shape Internet routes as they define a great part of BGP route-redistribution filters.

In this Ph. D. thesis, we solve several difficult problems dealing with reverse engineering of the Internet. We start by an introduction on Internet economy and interdomain routing with the BGP protocol. Then, we successively review several works respectively on : measuring the Internet topology, determining economic properties of Internet and infering Internet routing policies. We study three problems in this thesis : the measurement of a stable AS-level topology, the inference of AS business relationships and the modeling of interdomain routing at the AS-level. We propose a set of tools that bring insights on Internet economics for an operator like France Telecom. Knowledge produced by our algorithms is of great importance in negociating business relationships and in studying economic aspects of Internet interconnection markets.

Inférence des accords économiques et des politiques de routage dans l'Internet

Résumé

L'Internet est le réseau de transport pour les communications IP publiques. Ce réseau est formé d'environ 20 000 systèmes autonomes (AS) interconnectés, appartenant chacun à une entité administrative (un opérateur, une université, une administration, une entreprise...). Chaque AS a une connectivité totale vers l'ensemble des autres AS grâce au fonctionnement distribué du routage BGP (Border Gateway Protocol). Pour garantir la connectivité totale, les administrations des AS négocient des contrats économiques d'interconnexion bilatéraux dans lesquels les deux parties spécifient les différentes destinations accessibles via la liaison, les tarifs et si nécessaire les conditions à respecter pour la qualité et la disponibilité du service de connectivité. Le marché économique des interconnexions entre AS est auto-régulé par la concurrence. Schématiquement, les opérateurs historiques et les réseaux de transport IP fortement interconnectés sont avantagés dans les négociations commerciales, et les fournisseurs de contenu cherchent à minimiser leurs coûts et maximiser la qualité des liaisons. Les accords commerciaux établis entre AS ne sont pas publics alors qu'ils constituent une information clé dans les routages Internet puisqu'ils sont à l'origine des filtres et des préférences pour les routages de chaque opérateur.

Dans cette thèse on propose de résoudre plusieurs problématiques de mesure et d'inférence liées à l'économie de l'Internet. Après une introduction à cette économie et à la technologie de routage BGP, on propose une revue de travaux existants dans lesquels les auteurs ont cherché à mesurer et inférer la topologie, le routage et l'économie de l'Internet. Nous traitons dans cette thèse trois problèmes d'inférence inter-dépendants : la mesure des politiques de routage BGP, l'inférence des accords d'interconnexion économiques entre AS et la modélisation des chemins de routage BGP. On propose un jeu d'outils qui apporte un soutien pour un opérateur comme France Télécom à la gestion opérationnelle des réseaux et une base de connaissance de l'économie de l'Internet. Cette base apporte des informations déterminantes lors des négociations d'interconnexion, pour l'étude des modèles économiques d'opérateurs et pour la régulation de certains marchés d'interconnexion.

Remerciements

Je tiens tout d'abord à remercier Jean-François Mallordy, Daniel Roux, Alain Quillot, Jean-Marc Petit et Patrice Laurencot pour mon orientation vers un double cursus ingénieur ISIMA (Institut Supérieur d'Informatique et de Mathématiques Appliquées) et DEA (Diplôme d'Etude Appliquées) de l'université Blaise Pascal de Clermont-Ferrand. Je tiens à remercier ensuite Michel Décima, Quang Nguyen, Jean-Luc Lutton et Jean-François Legros pour m'avoir offert l'opportunité d'intégrer en 2003 l'équipe MCT (Modélisation des coûts et trafics) au sein du laboratoire DAC (Direction de l'Architecture et de la Commande des réseaux) de la division Recherche et Développement du groupe France Télécom. Je remercie très chaleureusement toutes les personnes dont Philippe Mahey et Adam Ouorou qui m'ont permis ensuite de démarrer la thèse dont ce document fait l'objet. Je tiens aussi à remercier L'ANRT (Association Nationale pour la Recherche Technique) pour la bourse de recherche CIFRE qu'ils m'ont accordé et Lionel Levasseur, Prosper Chemouil, Vincent Martin, Yannick Le Louedec, Marie-hélène Briens et Florent Bersani entre autres pour leur soutien hiérarchique au sein du groupe France Télécom. Pendant ces trois dernières années, j'ai eut la chance de rencontrer de nombreux chercheurs du monde industriel et académique français et international. Tous ces échanges m'ont beaucoup apporté d'un point de vue humain, scientifique et pédagogique. Je pense par exemple, pour les personnes que je n'ai pas déjà mentionné, à Ehoud Ahronovitz, Jean-Sebastien Bedo, Benoît Besset, Marc-Olivier Buob, Daniel Bienstock, Olivier Bonaventure, Mathieu Chardy, Vincent Gillet, Eric Gourdin, Olivier Klopfenstein, Bartosz Kozlowski, Dimitri Krioukov, Anthony Lambert, Ridha Mahjoub, Dritan Nace, Mustapaha Bouhtou, Timothy Griffin, James Roberts, Jin-Kao Hao, Sebastien Tandel et Steve Uhlig.

De nombreuses personnes m'ont aidé à relire et corriger ce document de thèse et je les remercie chaleureusement. Sans vouloir citer tout le monde, je pense à Florent Bersani, Marie-Hélène Briens, Marc-Olivier Buob, Anthony Lambert et Jean-Luc Lutton.

Enfin je souhaiterai remercier ma famille et mes amis qui m'ont toujours soutenu dans mes démarches professionelles.

Table des matières

Table des figures

Liste des tableaux

Glossaire

Peering relation économique d'interconnexion qui stipule que deux opérateurs établissent un libre échange entre leurs clients. Voir aussi *transit*, *sibling* et *backup*
page(s) : 6, 9, 10, 13, 63, 67, 91

Transit relation économique d'interconnexion établie entre un réseau client et un réseau fournisseur. Le client paye pour recevoir et envoyer du trafic. Voir aussi *peering*, *sibling* et *backup*
page(s) : 6, 9, 91

ARD Système de routage autonome. Cela correspond à l'ensemble des AS d'une administration qui sont souvent en relation de *sibling*
page(s) : 9

Autonomous System (AS) un domaine IP qui regroupe un ensemble de routeurs BGP et des réseaux
page(s) : 7, 14, 16, 20, 24, 31, 45

AS source AS ou AS voisin d'un routeur sonde BGP
page(s) : 48

AS stub AS client final qui n'assure aucun transit entre deux AS
page(s) : 52

AS_PATH attribut d'une route BGP qui désigne le chemin d'AS à parcourir pour atteindre le NLRI indiqué dans la route. L'attribut AS_PATH correspond à une suite de segments. Chaque segment est composé d'un numéro d'AS éventuellement répété plusieurs fois, ou alors de plusieurs AS dont on ne connaît pas l'ordre de parcours exact (*AS Set*). Lorsqu'il n'y a aucun AS Set, la séquence de segments qui composent un AS_PATH correspond au chemin d'AS inversement parcouru par le message BGP
page(s) : 18, 20, 31, 45

Atome de politique de routage d'annonce triplet (X, Y, p) où X et Y sont des AS et p est un préfixe. Il Identifie le parcours d'une route BGP pour un préfixe p de l'AS X vers l'AS Y. On peut associer un poids de répétition entier à l'atome qui correspond au nombre de répétitions de X (1 par défaut)
page(s) : 49

Atome de politique de routage de transit triplet d'AS (X, Y, Z). Il Identifie le parcours d'une route BGP de l'AS X vers l'AS Z en passant par l'AS Y. On peut associer un poids de répétition entier à l'atome qui correspond au nombre de répétitions de Y (1 par défaut). Le triplet d'AS permet de conclure que l'AS Y est un AS de transit
page(s) : 50

Atome de politique de routage de transit-préfixe atome de politique de routage avec un préfixe p. Il Identifie le parcours d'une route BGP de l'AS X vers l'AS Z en passant par l'AS Y pour le préfixe p. On peut associer un poids de répétition entier à l'atome qui correspond au nombre de répétitions de Y (1 par défaut). Le triplet d'AS permet de conclure que l'AS Y est un AS de transit
page(s) : 49

Backup relation économique d'interconnexion établie entre un réseau client et un réseau fournisseur. Le client utilise le réseau du fournisseur en cas de pannes de ses liaisons primaires. Voir aussi *peering*, *transit* et *sibling*
page(s) : 11

Border Gateway Protocol (BGP) Protocole de routage interdomaine utilisé depuis 1989 pour le réseau Internet
page(s) : 7, 14, 16, 22, 24, 31, 35, 39, 45

External BGP (eBGP) correspond au protocole de routage BGP lorsqu'il est utilisé entre deux routeurs appartenant à des AS différents
page(s) : 16, 18, 20, 22, 24, 38

Forwarding Information Base (FIB) base de donnée d'un routeur répertoriant les différentes routes utilisées par le routeur pour l'acheminement des paquets
page(s) : 19, 24

Gadget BGP configuration du routage non désirée entre plusieurs AS
page(s) : 16, 25

Internal BGP (iBGP) protocole de routage BGP utilisé à l'intérieur d'un même AS. Permet aux routeurs BGP d'un même AS de s'échanger des routes BGP en provenance de l'extérieur
page(s) : 16, 18, 20, 25, 38, 45

Inter-Domain Trafic Engineering (IDTE) méthode d'ingénierie de trafic interdomaine qui vise à optimiser la configuration du plan de contrôle BGP d'un AS
page(s) : 24, 25

Interior Gateway Protocol (IGP) protocole de routage utilisé dans un même AS ou ARD. On peut citer par exemple OSPF (Open Shortest Path First) ou IS-IS (Intermediate-System Intermediate System)
page(s) : 16, 20, 22, 45

Internet Protocol (IP) protocole réseau utilisé dans l'Internet
page(s) : 6, 9

Internet Service Provider (ISP) opérateur du réseau Internet offrant des services de connectivité
page(s) : 7, 28

Local Preference attribut BGP qui permet à un routeur de préférer une route avec un Local Preference maximal à la première étape du processus de décision BGP
page(s) : 20, 45

Matrice de politique de routage d'annonce ensemble d'atomes de politique d'annonces pour tous les préfixes d'un même AS
page(s) : 51

Matrice de politique de routage de transit ensemble d'atomes de politique de transit où un même AS est au milieu du triplet d'AS
page(s) : 51

Motif de répétition tableau d'entiers indiquant le nombre de fois qu'un AS est répété dans un chemin
page(s) : 46, 49

Next hop BGP attribut d'une route BGP dont la valeur est une adresse IP. Cette adresse est celle d'un routeur en bordure de l'AS pour les routes BGP externes (ou celle du routeur avec lequel le routeur de bordure établit la session eBGP) ou celle d'un routeur de l'AS s'il a donné origine à l'annonce de la route ou si il a activé l'option next-hop-self lors de la transmission de la route BGP
page(s) : 18, 45, 46, 48

Network Layer reachability Information (NLRI) attribut du protocole BGP qui représente une plage d'adresses IP. Un NLRI est aussi appelé préfixe
page(s) : 16, 18, 19, 25, 32, 37, 38, 45, 47, 63, 83–85

Registre de routage Internet (Internet Routing Registry) Entité administrative responsable de l'allocation des plages d'adresses IP pour des zones géographiques définies au préalable. Par exemple : RIPE (Réseaux IP européens), APNIC (Asia Pacific Network Information Center)... Les registres offrent un service d'accès publique à leur données grâce à un protocole whois
page(s) : 37

Routing Information Base (Loc-RIB) base de données d'un routeur BGP répertoriant les différentes routes pouvant être sélectionnées dans la table de forwarding FIB du routeur
page(s) : 19, 39, 45

Route canonique séquence d'AS sans répétition correspondant à un chemin d'AS
page(s) : 46, 48, 49

Routeur équipement qui permet de relayer les paquets IP. Voir *commutation* et *protocole de routage*
page(s) : 9, 16

Sibling relation économique d'interconnexion qui stipule que deux opérateurs établissent un libre échange de trafic. Voir aussi *peering*, *transit* et *backup*
page(s) : 11

Sondage actif Le sondage actif d'un réseau IP consiste à interroger en temps réel les équipements et en particulier les routeurs afin d'obtenir différentes informations qui varient au cours du temps
page(s) : 26

Sondage passif Le sondage passif du réseau Internet consiste à analyser des informations qui varient au cours du temps mais qui sont collectées sur des équipements à intervalles fixes
page(s) : 26

Sonde routeur qui permet d'obtenir des routes BGP avec des attributs *AS_PATH*. Une sonde est un routeur BGP recevant toutes parfois les tables BGP de multiples autres

routeurs

page(s) : 47, 48

Tiers niveau hiérarchique d'un opérateur de transit

page(s) : 12

Tomographie BGP ensemble de chemins d'AS extraits des attributs *AS_PATH* de routes BGP publiques

page(s) : 47

Introduction

L'interconnexion Internet aujourd'hui

L'Internet est aujourd'hui le réseau de transport de nombreux services de télécommunication[1]. Sa viabilité économique repose sur deux grands types d'activités qui sont complémentaires et indissociables : tandis que des opérateurs de services proposent des offres de contenu en supposant transparent le médium de transport, des milliers de fournisseurs d'accès (ou opérateurs de transit) vendent la connectivité Internet de leur infrastructure assurant alors la connectivité globale pour chaque réseau. La connectivité proposée par un opérateur de transit fonctionne avec le protocole IP[2] et permet d'utiliser de multiples protocoles applicatifs et services de communication sans tenir compte du support physique de transmission[3]. Le routage IP entre les réseaux fonctionne de manière distribuée et il n'est pas possible d'avoir le contrôle des communications de bout en bout ni de garantir la qualité du service de connectivité[4]. L'interconnexion des différents domaines IP (systèmes autonomes ou AS[5]) de chaque opérateur de transit est nécessaire pour la connectivité globale de chaque réseau. Elle s'effectue dans un environnement économique non régulé et n'est soumise qu'aux seules règles de la concurrence. Chaque opérateur est libre d'appliquer ses tarifs et de s'interconnecter à n'importe quel(s) autre(s) réseau(x). Cette situation engendre des jeux de forces entre les acteurs de l'Internet qui cherchent chacun à maximiser leur intérêt. Généralement, la négociation d'un accord d'intercon-

[1] par exemple l'email, le chat, la vente en ligne, la téléphonie, la vidéo-conférence, les échanges Peer-To-Peer...

[2] Internet Protocol

[3] A la couche IP, un réseau est formé d'un ensemble de noeuds inter-connectés par des liaisons adressés dans les deux sens.

[4] Les informations échangées entre deux hôtes IP sont acheminées dans le réseau Internet par paquets de données. Les paquets de données empruntent simultanément différents chemins asymétrique qui sont décidés dynamiquement par les routeurs de tous les domaines. Les paquets peuvent arriver en plusieurs exemplaires et dans le désordre ou pas du tout.

[5] Autonomous System

nexion bilatéral tourne à l'avantage du plus fort. Depuis dix 10 ans, un oligopole d'une dizaine de grands opérateurs se charge de garantir la connectivité Internet globale. Le routage Internet est resté jusqu'à aujourd'hui hiérarchique.

Motivations

Le routage IP entre les domaines (routage inter-AS) est assuré par le protocole de routage BGP[6]. Ce protocole est implanté sur les routeurs interconnectant les AS pour mettre à jour les tables de routage inter-AS de chacun. Le mécanisme distribué d'échange de messages BGP permet aux routeurs de connaître au moins une route pour chacune des plages d'adresses IP[7]. Dans la pratique, un opérateur modifie librement la politique de routage de ses AS par le biais des configurations des routeurs BGP[8]. Le plan de routage BGP avec politiques[9] constitue la marche de manoeuvre technologique d'un opérateur : tout est permis tant que les engagements de connectivité stipulés dans les contrats d'interconnexion sont respectés. D'une part, la connaissance du réseau et des politiques de routage déployées est une information essentielle pour un opérateur afin qu'il négocie ses propres accords d'interconnexion commerciaux, qu'il fixe les règles de sécurité de ses AS et qu'il optimise la configuration du routage BGP dans ses AS. D'autre part les organismes de régulation, les opérateurs et les fournisseurs de contenus cherchent à obtenir une meilleure connaissance de l'économie de l'Internet pour optimiser leurs coûts. Finalement les équimentiers, les opérateurs et la communauté scientifique veulent connaître et modéliser les routages interdomaines pour contribuer à la recherche sur l'amélioration des performances et de la qualité de service du routage interdomaine. L'inférence de la topologie de l'Internet, des politiques de routage BGP, de la structure économique du réseau des AS et des routages interdomaines constituent les problèmes qui seront abordés dans cette thèse.

[6]Border Gateway Protocol

[7]Pour l'instant, les adresses IPv4 forment la majeure partie de l'espace Internet publique. Les plages d'adresses IPv4 disponibles étant bientôt épuisées, les opérateurs commencent à utiliser IPv6. On se restreint dans cette thèse à l'Internet IPv4, sachant que l'extrapolation vers IPv6 est immédiate.

[8]Chaque opérateur implante des règles de sécurité et de préférence pour la sélection et la redistribution des informations de routage BGP.

[9]Le routage avec politique consiste à utiliser des règles de préférence, de refus et de redistribution des chemins de routage échangés dans les messages protocolaires.

Difficultés

L'information nécessaire pour reconstituer les routages interdomaine dans l'Internet est indisponible car le routage fonctionne sur un mode décentralisé et les opérateurs ne divulguent pas leurs accords d'interconnexion économiques. Les configurations BGP et les informations de routage intra-domaine sont propres à chaque opérateur. Elles sont soumises à de nombreuses évolutions. De plus, le nombre important de domaines et de routeurs BGP apporte des difficultés supplémentaires pour la récupération, le stockage, la manipulation, et l'analyse de ces informations de routage. Même en disposant de l'intégralité des configuration de tous les domaines de l'Internet, la reconstitution des décisions de routage des routeurs, nécessiterait de simuler le processus d'échange et de sélection des messages BGP. Ceci est actuellement impossible du fait de la lourdeur de cette simulation et des capacités des ordinateurs. La convergence incrémentale du protocole BGP et le manque de connaissance des politiques de routage s'ajoutent à ces contraintes pour montrer l'impossibilité pratique de simuler le routage exact de bout en bout dans l'Internet. En plus des difficultés de mise en oeuvre d'une plate-forme de mesure distribuée et d'accès aux données économiques et BGP des opérateurs, les problèmes inverses d'inférence des accords d'interconnexion entre AS et de calcul des chemins inter-AS sont difficiles compte tenu de leur grande taille et de la combinatoire des possibilités.

Objectifs de la thèse

On s'intéresse dans cette thèse à définir la structure économique de l'Internet : déterminer et étudier la topologie du réseau Internet à l'échelle des AS, inférer les accords d'interconnexion entre AS et les chemins inter-AS qui respectent les politiques économiques de routage. On propose de modéliser les décisions de routage BGP en identifiant la nature économique des interconnexions et des politiques de routage à l'échelle des AS. On exploite des bases de données de routage BGP publiques pour inférer la structure économique de l'Internet. Les algorithmes d'inférence proposés exploitent un modèle algébrique des routages interdomaines respectant les contraintes des politiques de routage de chaque opérateur. Les hypothèses formulées pour nos modèles et nos algorithmes, prennent en compte l'échelle "AS" à laquelle le réseau interdomaine est étudié.

Contributions et organisation du document

Tout d'abord, au chapitre 1 on propose de montrer pourquoi la connaissance des politiques de routage est importante, et comment ce problème a été traité de façon directe ou indirecte dans la littérature. Puis, au chapitre 2 on procède à la mesure de l'Internet interdomaine par tomographie BGP. En utilisant des données de routages BGP à intervalles réguliers, on montre comment obtenir au moyen de filtres une vue stable du graphe des AS et des routages BGP correspondant à une vision statique de l'Internet. On définit en particulier des matrices de politique de routage qui permettront d'inférer l'organisation hiérarchique des routages BGP et de valider nos hypothèses de modélisation des politiques de routage économique à l'échelle des AS.

Dans le chapitre 3 on montre comment inférer les accords d'interconnexion entre AS grâce à deux méthodes : une méthode heuristique permettant l'obtention de solutions sans cycle clients-fournisseur, et une méthode exacte. On définit une classe de problèmes d'optimisation appelée MaxVCAP[10] qui correspond à une généralisation du problème MaxTOR[11] NP-complet étudiée dans la littérature pour l'inférence des accords d'interconnexion entre AS. Dans cette thèse on propose une variante du problème MaxTOR appelée MaxTOR3 et de nouvelles contraintes dans le problème permettant d'obtenir des solutions plus fidèles à la réalité que les solutions pouvant être obtenues en résolvant le problème MaxTOR classique. On parvient à résoudre MaxTOR3 pour des instances réelles de très grande taille. En effet, la méthode de résolution exacte systématique proposée pour les problèmes MaxVCAP est raffinée pour le problème d'inférence en utilisant notamment une méthode de réduction et en introduisant des hyperplans efficaces coupant l'espace des solutions du modèle par programmation linéaire en nombres entiers.

Dans le chapitre 4, à l'aide d'hypothèses classiques sur les configurations des politiques de routage adoptées par les fournisseurs d'accès Internet, on propose une modélisation du routage interdomaine grâce à un graphe et un espace algébrique de métriques BGP simplifiées. Le modèle permet de calculer, en un temps polynomial, des chemins BGP qui respectent les contraintes économiques des relations d'interconnexion inter-AS. L'intérêt de cette approche "par modélisation analytique" est l'efficacité des calculs au détriment de l'hypothétique précision qui pourrait être atteinte par un simulateur du protocole BGP. La formulation algébrique proposée pour le problème du plus court chemin BGP inter-AS

[10] Maximum Valid Consecutive Arc Pairs
[11] Maximum Type Of Relationship.

est novatrice par rapport aux travaux existant sur le sujet et on prouve que l'algorithme de Dijkstra généralisé peut être utilisé pour calculer les chemins de "n sources" vers 1 destination sans que la structure ne respecte la propriété d'être doïde.
Au travers de cette thèse on conçois un ensemble de modèles et algorithmes et on développe un jeu d'outil qui permettent d'obtenir une meilleure connaissance de l'organisation de l'Internet et de son économie : politiques de routages, accords économiques d'interconnexion et chemin interdomaines de bout en bout.

Chapitre 1

La connaissance des politiques de routage

Le réseau Internet est composé de réseaux d'opérateurs et de clients. Ces réseaux sont interconnectés pour communiquer avec Internet Protocol (IP). L'interconnexion entre réseaux IP n'est pas soumise à une réglementation économique[1]. La stratégie d'interconnexion d'un opérateur se traduit en pratique au choix entre deux modes d'interconnexion économique antagonistes. Un opérateur va soit vendre ses capacités de réseaux (on parle alors de service de *transit*), soit acheter les capacités d'un autre réseau, soit partager gratuitement l'accès vers la totalité ou un sous-ensemble géographique de ses clients via une relation dite de *peering*. Le revenu d'un opérateur est conditionné par les types d'accords d'interconnexion négociés et les volumes de trafic routés par ses équipements pour les clients (profit) ou ceux de ses fournisseurs (consommation). Pour un opérateur, les multiples possibilités d'extension géographique de son réseau et la liberté dans le choix des AS voisins et des types de contrat, constituent un problème de décision économique très important et difficile. Le maintien ou la mise en place d'une liaison entre deux opérateurs fait l'objet de négociations commerciales entre les deux parties. Ces négociations sont menées pour (re-)définir un type de contrat d'interconnexion et ses modalités. Les différents points d'interconnexion, les capacités des liaisons, la nature du contrat et le prix, sont négociés en fonction des pré-requis de chaque opérateur (définis dans les politiques de d'interconnexion ou "Peering Policies"), des services proposés et de leur popularité,

[1]Les modalités économiques d'interconnexion utilisées aujourd'hui comme par exemple le partage gratuit de trafic ("Sender Keep All"), ont une origine historique. D'autres modèles modèles économiques éprouvés dans le domaine de la téléphonie ou des échanges d'énergie sont inapplicables.

des volumes échangés par les utilisateurs (les différentes destinations générant le plus de trafic), mais surtout en fonction de la taille estimée de l'opérateur.

Dans ce chapitre on étudie différentes facettes du processus d'interconnexion du réseau d'un opérateur. On commence par détailler le fonctionnement de Border Gateway Protocol (BGP) qui permet à chaque opérateur de s'interconnecter au réseau Internet et de contrôler les routages opérés par ses réseaux. Ce protocole permet aux administrateurs de délimiter un réseau en un ou plusieurs systèmes autonomes Autonomous System (AS) indépendants. Chaque AS établit une politique de routage qui respecte à la fois des critères économiques et des critères de performance[2]. D'un point de vue contrôle, l'optimisation de la configuration des routeurs BGP d'un AS est une problématique multi-critères complexe. Les contraintes économiques engendrées par les échanges entre opérateurs voisins, la sécurité, la rapidité de reconfiguration dynamique et les subtilités du routage interdomaine sont autant de paramètres auxquels sont confrontés les administrateurs pour définir une politique de routage robuste et peu onéreuse. Les meilleures pratiques issues de la littérature pour l'aide à la configuration sont basées sur la connaissance de la topologie du réseau Internet, la connaissance des routages mis en jeu, et sur les volumes de trafic échangés. Malheureusement, en règle générale, les nombreuses méthodes d'ingénierie de trafic utilisées ou utilisables souffrent d'un manque d'informations disponibles, alors que la volumétrie des données est de plus en plus conséquente avec le temps[3].

Dans la suite du chapitre, on montre pourquoi la connaissance du réseau Internet et des politiques de routage des opérateurs est une problématique complexe. Ce sujet déjà abordé dans la littérature, fait l'objet de nombreuses recherches. Ces travaux sondent le réseau Internet pour obtenir sa topologie. Cette topologie peut être obtenue pour différents niveaux de détail. On montre d'abord comment mesurer la topologie de réseaux à la couche IP. Puis on montre comment établir la topologie du réseau inter-AS de plusieurs manières. Enfin on montre que la connaissance des politiques de routage dans l'Internet est, malgré son importance, un sujet de recherche encore peu avancé, montrant alors l'intérêt des contributions de la thèse. La mesure et la compréhension des routages Internet est un point essentiel dans les études technico-économiques d'interconnexion interdomaine, dans les problématiques de configuration BGP, dans l'étude de l'organisation des opérateurs.

[2]Une politique de routage consiste à fixer des règles de sélection et de redistribution des informations de routage BGP.

[3]Les volumes de données échangés, les besoins en bande passante pour le trafic IP et le nombre d'acteurs présents sur le marché des communications IP suivent une perpétuelle expansion.

1.1 Économie de l'Internet et enjeux

Les enjeux économiques d'un fournisseur de transit Internet ou Internet Service Provider (ISP) sont multiples. Un ISP doit à la fois rentabiliser les investissements effectués dans son réseau, préparer les investissements futurs relatifs à son développement, et configurer son réseau afin d'optimiser ses coûts. Dans un premier temps, on va voir quelques raisons qui expliquent pourquoi les différents acteurs cherchent à agir sur le système en jouant au niveau de la configuration des mécanismes de routage BGP. On présente ensuite les différents types de contrats négociés entre opérateurs (peering et transit) et on explique leur influence sur la nature des routes empruntées dans le réseau Internet. En effet, les routages entre ISP ne sont pas des plus courts chemins car ils doivent respecter les contrats économiques établis deux à deux entre ISP. La nature et le nombre de réseaux partenaires et clients d'un opérateur peuvent avoir des conséquences dramatiques sur les coûts et les performances. Enfin, on discute de l'introduction possible d'une régulation de l'Internet, qui pourrait faire disparaître les accords de peering aujourd'hui très discutés car non taxés.

1.1.1 L'économie du trafic Internet

Depuis le début des années quatre-vingt-dix, le réseau Internet est un lieu d'échange public dont les épines dorsales appartiennent à quelques opérateurs privés et institutions. Le réseau est perçu comme un média transparent et fonctionnel où différents types de services peuvent être déployés. De l'envoi d'e-mail jusqu'au téléchargement de pages web et de contenus multimédia (images, musique, vidéo), du e-commerce jusqu'aux banques en ligne, du téléchargement de fichiers jusqu'aux échanges d'informations via les réseaux peer-to-peer, des radios en ligne jusqu'au streaming vidéo temps réel, de la télé-chirurgie jusqu'à la réalité virtuelle, le réseau Internet est et sera utilisé pour de multiples besoins. La diversité des usages et des services du réseau Internet implique l'existence de différents types de trafics et de passerelles de services déployés sur le réseau. Le routage IP seul ne fait intervenir aucune notion de circuit, de flot ou d'état dans le réseau. Même si grâce à des technologies de type MPLS[4] on peut mettre en place un routage avec des circuits dans un même domaine, le protocole BGP reste utilisé pour l'Internet. La « qualité de service Internet » obtenue de bout en bout (ou en d'autres termes les chemins utilisés dans le réseau) résulte du routage BGP entre les AS. Ce routage de type "best effort"[5] est de plus

[4]Multiple Path Label Switching

[5]Le réseau Internet publique fonctionne en mode non connecté : chaque entité communicante découpe

opéré sans garantie d'acheminement par les multiples opérateurs intervenant sur les routes empruntées. Lorsque deux machines du réseau Internet communiquent, les ressources de différents opérateurs sont mises en jeu et nécessitent une contre-partie financière. Malgré l'importance de la définition d'un système de recouvrement des coûts [32] et la richesse des travaux concernant les modes de tarification [151], un schéma de tarification complètement équitable comme en téléphonie n'existe pas pour les raisons suivantes :

Asymétrie des routes : une communication correspond a priori à deux routes distinctes dans le réseau (allé et retour). Voir figure 1.1.

Mesure : la taille du réseau et la dynamique des changements de route rend difficile la connaissance des routages exacts.

Acheminement : le routage IP au "best effort" ne garantit pas l'acheminement des paquets. Il est difficile de vérifier que des paquets ont bien été acheminés jusqu'à une destination sans ajouter une notion d'état dans le réseau.

Unité de transaction : sachant qu'un réseau IP est multi-services, il est difficile de savoir sur quelles transactions effectuer une tarification.

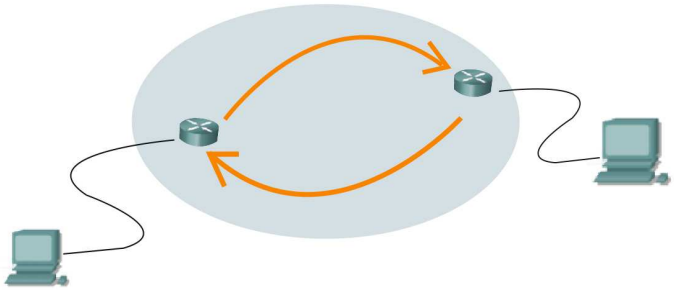


FIG. 1.1 – Asymétrie des chemins dans le réseau Internet

Pour de plus amples explications, on pourra se reporter au document [103]. Le mécanisme distribué BGP permet d'effectuer des routages entre opérateurs et permet à chacun d'être indépendant (voir en 1.2.1). En contrepartie, le système de communication obtenu n'offre donc pas la possibilité de contrôler totalement les routages de bout en bout. Les opérateurs de transit, libres dans leurs politique d'interconnexion, ont opté pour une organisation économique simple où seulement deux grands types de contrats sont négociés.

ses données à expédier en paquets, puis transmet ces derniers aux réseaux voisins afin qu'ils puissent à leur tour permettre l'acheminement jusqu'à leur destination. Les échanges de bout en bout sont possibles grâce à une coopération successive des réseaux mis en jeu dans le cheminement des informations, sans garantie d'acheminement particulière.

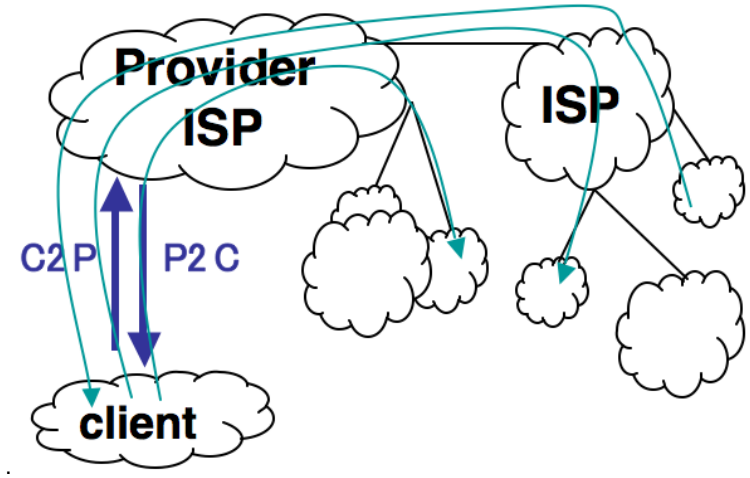
1.1.2 Interconnexions des opérateurs

L'Internet se constitue d'entités administratives qui communiquent entre elles. Ces entités administratives (pouvant être des clients finaux, des universités, des opérateurs de réseaux ou ISP, des administrations, des organismes de défense,...) disposent d'un réseau que l'on peut qualifier de domaine autonome de routage (ARD)[6]. Chaque ARD est autonome pour son routage interne et dispose d'une main mise complète concernant l'administration des routeurs. Par contre, un ARD n'a a priori aucune connaissance des autres ARD, et ne dispose d'aucun contrôle sur leur routage intra-ARD. L'indépendance de chaque ARD se traduit aussi par l'indépendance en termes de stratégie d'interconnexion. Chaque ARD est complètement indépendant économiquement et la négociation d'un accord d'interconnexion entre deux ARD tourne souvent à l'avantage du réseau le plus "connecté" qui propose la vente de ses capacités. Pour la vente de connectivité, on parle de relation de *transit* (figure 1.2(a)) : pour chaque liaison, l'ARD client paye l'opérateur fournisseur en fonction des volumes de paquets échangés (les deux sens sont payants), des points géographiques d'interconnexion, de la bande passante maximale et des clauses de qualité et de disponibilité établies dans des contrat appelés Service Level Agreement (SLA). Lorsque deux ARD sont de taille équivalente et si chacun dispose d'assez de clients pour être rémunéré correctement, ou lorsqu'un des deux réseaux est un fournisseur de contenu important[7], ou tout simplement pour des raisons de performance, deux réseaux peuvent mettre en place un partage des communications IP entre leurs clients respectifs. Ce partage s'effectue sans contrepartie financière et sans garantie particulière. On dit que les deux ARD sont en relation de *peering* ou "Sender Keep All". Un tel accord permet d'une part de réduire les délais de bout en bout entre clients des deux ARD. D'autre part, il offre une opportunité financière aux deux opérateurs, qui ne mettent pas en place de système de comptage sur les liaisons de peering, et qui ne reversent pas d'argent. Deux ARD vont espérer plus de trafic (donc plus de revenus) de leurs clients, par la nouvelle liaison de peering. De véritables jeux de forces se sont mis en place [100, 103, 117, 151, 185] aujourd'hui du fait de l'existence des accords de peering. Voici un exemple :

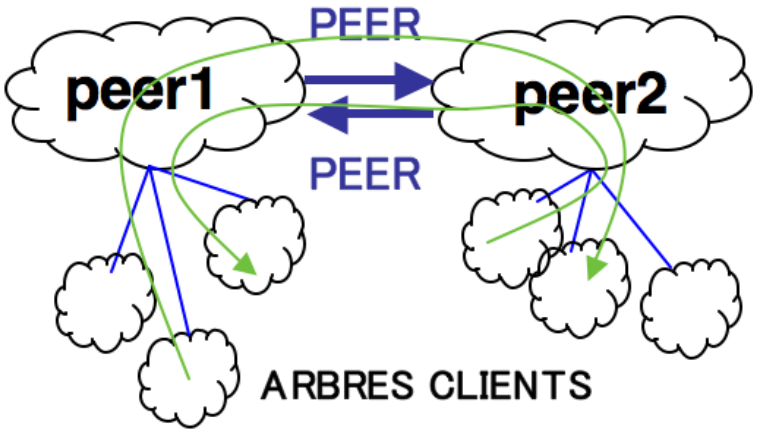
[6]Un ARD peut être un AS ou plusieurs AS.

[7]Certains réseaux peuvent être de petite taille, mais fournir des services très populaires (exemple : Google, Yahoo, eBay, Microsoft,...). Ils contribuent au contenu des données transmises sur le réseau Internet. Ces réseaux réussissent à négocier des contrats de libre échange (Peering) avec d'autres réseaux de grande taille qui ont investi beaucoup d'argent dans leur infrastructure et fournissent un service de transport des données. La dualité contenu/contenant entre les services basés sur la technologie IP est problématique au regard de la neutralité des réseaux IP ("Network Neutrality").

– Supposons que l'opérateur France Télécom et l'opérateur AT&T (opérateur historique américain) ont négocié un accord de peering. Chacun des deux réseaux peut accéder à tous les clients de l'autre via les liaisons de peering. Si l'accord de peering disparaît, les deux opérateurs doivent trouver un autre moyen d'accéder aux clients destinations respectifs. Le moyen le moins coûteux pour chacun est d'établir un accord de peering avec un fournisseur de l'ancien partenaire ou d'être le fournisseur d'un des fournisseurs de l'autre.



(a) Relation de Transit : un opérateur client paye un fournisseur pour pouvoir communiquer avec ses partenaires peering, clients et fournisseurs dans les deux sens. La relation d'un client vers un fournisseur est notée C2P, et la relation d'un fournisseur vers un client est notée P2C.



(b) Relation de Peering : deux opérateurs partagent gratuitement les communications entre leurs clients (et les clients des clients...). Les relations de peering sont notés PEER.

FIG. 1.2 – Les principaux contrats économiques d'interconnexion négociés entre opérateurs

Il existe d'autres modes d'interconnexion moins courants. Par exemple, les réseaux d'une même administration effectuent souvent un transit mutuel l'un pour l'autre, sans contrainte de routage particulière. Un tel contrat économique est appelé *sibling* ; Il est aussi possible d'établir des VPN[8] interdomaine ou du Peering virtuel[9], mais dans ce cas les contrats sont négociés spécifiquement. Il est aussi possible d'établir une liaison de secours vers un fournisseur. La relation client-fournisseur est appelé alors relation de *backup*. Dans la mesure où les relations de backup ne sont empruntées qu'en cas de pannes, les routages sont soumis à des contraintes spéciales, et la tarification peut aussi être modifiée.

Il existe des cas particuliers où un opérateur client va établir une relation de transit sur un sous-ensemble des destinations joignables par l'opérateur fournisseur. On parle de transit régional pour les relations de transit concernant les communications vers un unique continent, et de transit "domestic"[10] lorsque la connectivité est réduite à un unique pays. Les relations de peering peuvent elles aussi concerner un sous-ensemble géographique des clients, on parle alors de peering régional ou national. On notera aussi que certains opérateurs établissent des relations économiques hybrides transit/peering suivant la géographie des destinations, ou suivant la nature du trafic, comme dans le cas de la voix sur IP. En général, deux opérateurs optent pour la solution peering ou transit quels que soient les préfixes.

La nature hiérarchique des relations de transit (client vers fournisseur et fournisseur vers client), la nature symétrique des relations de peering et les jeux de pouvoir entre opérateurs ont mené à la structuration du réseau Internet en couches. Ces niveaux hiérarchiques, aussi appelés "tiers", suivent des lois d'échelle : il existe très peu d'opérateurs de plus haut niveau qui sont fortement interconnectés, alors qu'il existe beaucoup de réseaux clients en périphérie du réseau (aussi appelés réseaux stub). Les opérateurs au niveau le plus haut de la hiérarchie, sont appelés "Tiers-1". Ils peuvent fournir toutes les routes de l'Internet car l'oligopole des opérateurs Tiers-1 forme un ensemble de réseaux fournisseurs ou peer vers la totalité des autres réseaux. L'union des réseaux clients des opérateurs "Tiers-1" forme la totalité des réseaux. Les opérateurs Tiers-1 sont donc tous interconnectés deux à deux en relation de peering afin que chacun puisse avoir connaissance de la totalité des destinations

[8] Réseaux Privés virtuels

[9] Le Peering virtuel est un service offert par quelques opérateurs. Cela consiste à raccorder un réseau à un point d'interconnexion de façon transparente (en procédant à une encapsulation des paquets de données IP).

[10] Attention, le mot domestic est un anglicisme qui désigne national.

de l'Internet. Ce maillage complet[11] est appelé default-free zone, indiquant que chaque opérateur Tiers-1 connaît un chemin de routage vers chaque destination de l'Internet (via un de ses clients, ou via un peer). On se reportera aux travaux dans [137–139, 204] pour de plus amples détails.

Les contrats établis entre opérateurs impliquent des règles économiques concernant les routages :

Accords de peering : par définition d'un accord de peering, deux opérateurs n'utilisent leur lien de peering que pour accéder à leurs clients respectifs (figure 1.2(b)). Dans le cas contraire, un des deux opérateurs consomme à tort les ressources de l'autre opérateur qui peut éventuellement payer le trafic si ce dernier est routé avec ses fournisseurs. Le cas où un opérateur transmet les paquets d'un fournisseur vers un peer fait aussi perdre de l'argent à l'opérateur au profit de son fournisseur qui gagne de l'argent à chaque fois qu'il communique par le peer de son client.

Clients multi-fournisseurs : considérons un opérateur qui dispose de plusieurs fournisseurs. Si l'opérateur permet les communications entre deux de ses fournisseurs, le premier fournisseur fait payer l'opérateur client pour ses communications vers l'autre fournisseur et inversement.

Ces règles de viabilité économique utilisées par chaque opérateur (sauf accords d'interconnexions particuliers ou erreurs) sont reportées dans le tableau 1.1. D'après ces contraintes économiques, on peut montrer que les chemins vérifient certaines propriétés en terme d'accords d'interconnexion. On donne figures 1.3(a), 1.3(b) et 1.3(c) des exemples de cheminements inter-opérateurs possibles (économiquement valides), et figures 1.3(d), 1.3(e)

[11] Le coeur du réseau Internet est donc en théorie un maillage complet des opérateurs Tiers-1, à la différence près que certains opérateurs participant au coeur ne sont pas Tiers-1 car ils possèdent encore un fournisseur ou n'ont pas établi de relation de peering avec l'ensemble des opérateurs Tiers-1.

Opérateur de Sortie → ↓Opérateur d'entrée	Fournisseur (C2P)	Client (P2C)	Peering (PEER)
Fournisseur (P2C)	0	1	0
Client (C2P)	1	1	1
Peering (PEER)	0	1	0

TAB. 1.1 – Matrice de compatibilité exprimant les contraintes économiques de transit entre les différents partenaires économiques d'un opérateur.

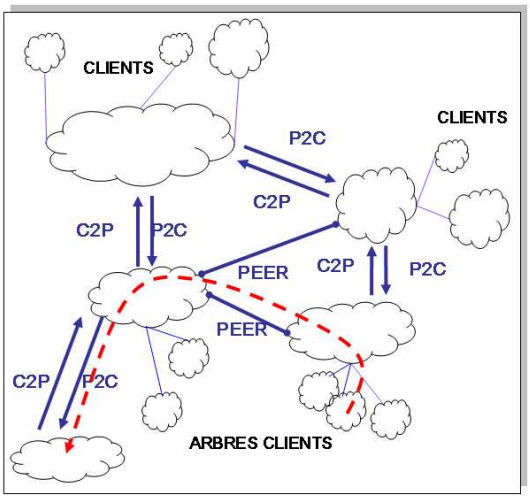
et 1.3(f) des exemples de cheminements économiquement invalides qui sont censés ne pas exister dans la réalité.

1.1.3 Perspectives de régulation de l'Internet

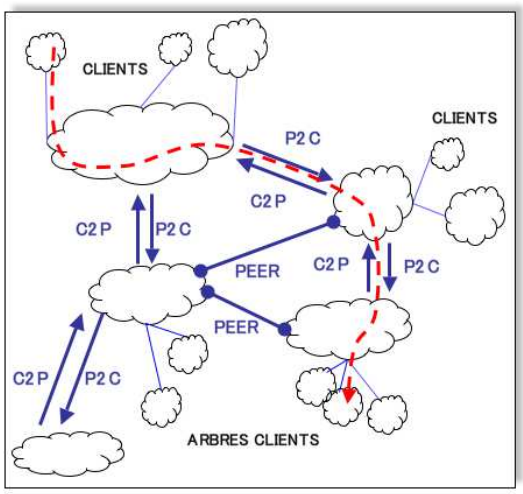
Imaginons une évolution du système de routage interdomaine qui permettrait une meilleure surveillance et la mise en place d'une autorité de régulation des interconnexions interdomaines et du prix de vente du transit. Le cas échéant, les relations de peering pourraient sûrement disparaître, ou devenir payantes puisqu'elles sont pour l'instant exonérée de taxes. Ce scénario pourrait se révéler économiquement critique, menaçant chaque opérateur d'une modification drastique de leurs revenus et coûts de trafic. Dans cette perspective économique de taxation *quasi* alarmante, la connaissance du réseau et des routages inter-opérateurs, ainsi que l'inférence des positions économiques et des stratégies des opérateurs Internet est une nécessité. L'incertitude actuelle globale concernant la structure économique exacte du réseau Internet montre qu'il est important de pouvoir inférer le graphe d'interconnexion des opérateurs avec les différents types de contrats. Cette information permet d'observer les stratégies des opérateurs.
Si on se place d'un point de vue plus global, la viabilité économique du réseau Internet est assurée par les clients finaux dont la demande concerne le contenu (par exemple pouvoir surfer sur le web) ou des capacités d'écoulement (par exemple la voie sur IP ou le téléchargement Peer-To-Peer). Les contenus sont fournis par les opérateurs de services et transportés par les opérateurs de transit. Dans une possible régulation des interconnexions Internet, il faudrait prendre en compte les externalités positives des offres de contenu. Les débats actuels autour de la "neutralité des réseaux" tentent de répondre à cette problématique.

1.2 Interconnexions et politiques de routage inter-opérateurs

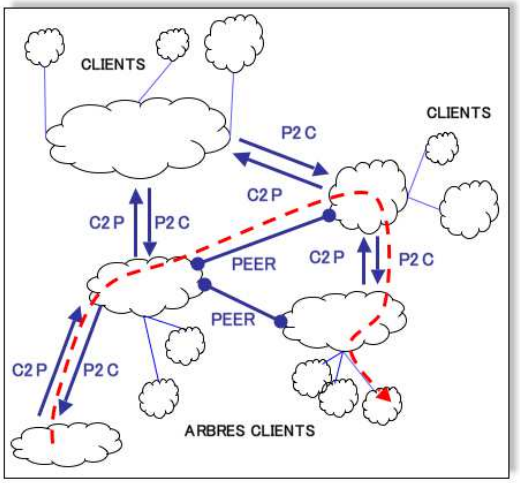
Un ISP doit pouvoir accéder aux plages d'adresses IP publiques pour s'interconnecter au reste du réseau Internet et être joignable depuis n'importe quel autre réseau connecté à l'Internet. Le routage interdomaine BGP permet à chaque routeur de savoir où router des paquets vers une adresse IP publique donnée. Les informations de contrôle du routage entre ISP, échangées en tant que messages BGP entre routeurs, permettent à chaque ISP d'annoncer ses propres plages d'adresses et de connaître les chemins de routage (savoir à quel réseau voisin envoyer les paquets) pour chaque plage d'adresses (du spectre) IP pu-



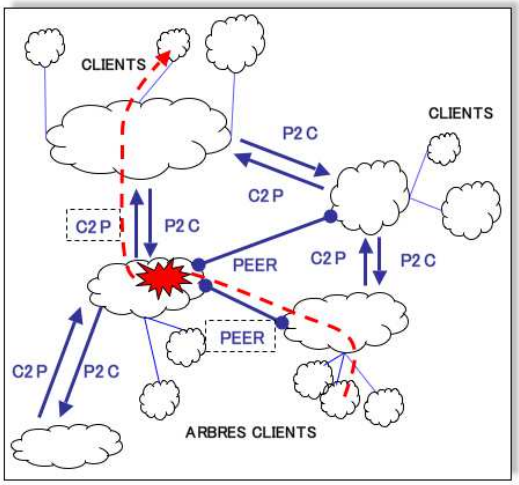
(a) Cheminement valide (1)



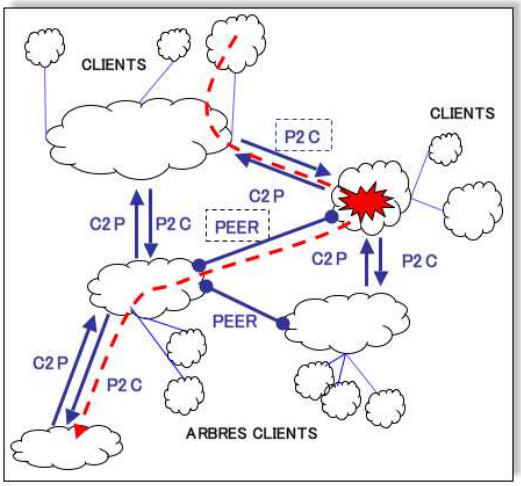
(b) Cheminement valide (2)



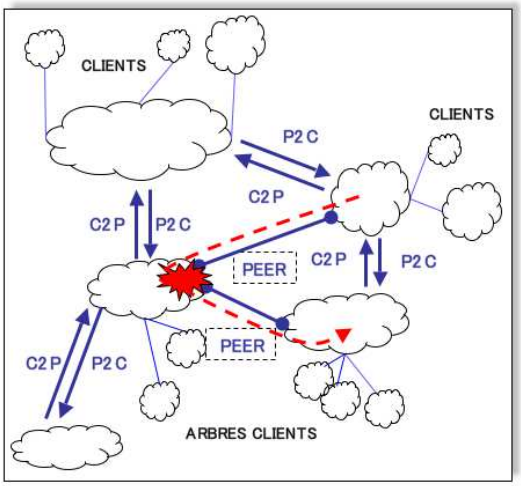
(c) Cheminement valide (3)



(d) Cheminement invalide (1)



(e) Cheminement invalide (2)



(f) Cheminement invalide (3)

FIG. 1.3 – Exemples de cheminements inter-opérateurs

blique. Chaque AS annonce ses propres plages aux AS voisins, qui à leur tour annoncent aux AS distants la disponibilité des plages. Les routeurs d'un AS ont le choix de propager ou conserver les annonces des disponibilités de plages d'adresses qu'il reçoivent. L'acheminement final de paquets IP se fait en traversant plusieurs AS successifs dans le sens inverse des annonces de messages BGP. Chaque AS est indépendant dans ses choix de routage et chaque ISP mets en place ses propres règles de routage BGP appelées *politiques de routage*. Les politiques de routage sont propres à chaque AS. Le routage de bout en bout est décidé de manière distribuée en respectant chaque politique. Après quelques précisions concernant le fonctionnement du routage interdomaine BGP, on abordera la manière dont un opérateur configure ses routeurs.

1.2.1 Le routage Internet interdomaine par le protocole BGP

Le protocole BGP permet aux routeurs de constituer des tables de routage dont les destinations sont les plages d'adresses IP publiques. BGP est un EGP[12] et il est mis en place depuis 1989 [153]. Ce protocole est aujourd'hui une technologie éprouvée [81] mais qui suscite de nombreuses interrogations [3, 20] , et souffre de nombreux problèmes : passage à l'échelle [22] notamment avec IP version 6 [38], vitesse de convergence et instabilité [13, 52, 74, 113, 197], conflits de politiques de routage [72, 75], sécurité [80, 136]... De plus, BGP tel qu'il a été conçu ne semble pas capable de répondre seul aux nouveaux besoins en termes de qualité de service et de nouvelles propositions sont régulièrement proposées sans être adoptées. Utiliser et optimiser le fonctionnement du protocole BGP entraîne des problèmes opérationnels complexes[13] qui font de ce protocole une technologie très particulière [50]. Pourtant, ce protocole reste une réalité technique incontournable pour un opérateur de télécommunications IP. Nous introduisons ici les principes algorithmiques qui gouvernent le fonctionnement de ce protocole. Ces principes nous permettront par la suite la mesure et l'étude de l'Internet à la granularité interdomaine.

Le protocole BGP découpe un réseau en systèmes autonomes (AS). Chaque système autonome est identifié par un numéro codé sur 16 bits[14] (de 1 à 65536) et regroupe un ensemble de routeurs et de réseaux locaux (LAN[15]). Les routeurs d'un même AS utilisent géné-

[12] External Gateway Protocol

[13] Certaines configurations de routage instables et les gadgets de configuration BGP entre plusieurs AS sont difficiles à déterminées.

[14] Le nombre restreint d'AS possibles pose un problème. Les numéros d'AS seront peut-être bientôt codés sur 32 bits.

[15] Local Area Network

ralement un même protocole de routage interne Interior Gateway Protocol (IGP)comme IS-IS[16] ou OSPF[17] afin d'assurer de bonnes performances pour les routages intra-AS et d'être indépendant vis à vis du reste de l'Internet. Dans un même AS, certains routeurs[18] vont implémenter le protocole BGP afin d'assurer la redistribution des informations de routage BGP (routage en direction de la sortie de l'AS). Notons que chaque routeur BGP est forcément rattaché à un unique AS public ou privé [188]. Les routeurs BGP établissent des sessions TCP[19] bilatérales et permanentes pour s'échanger des messages protocolaires. On dit que deux routeurs BGP qui partagent une session sont ***voisins***. La figure 1.4 illustre un petit réseau BGP. Notons que lorsque le protocole est utilisé pour la redistribution des routes Internet entre routeurs d'un même AS, on parle de Internal BGP (iBGP), et lorsqu'il est utilisé pour l'échange de routes entre AS, on parle de External BGP (eBGP). Les messages échangés BGP entre routeurs concernent soit l'établissement de la session soit la mise à jour de chemins de routage. Les destinations d'un réseau BGP sont des plages d'adresses IP, aussi appelées Network Layer reachability

[16]Intermediate-System to Intermediate-System
[17]Open Shortest Path First
[18]en général, la majorité des routeurs en bordure d'un AS implémentent le protocole BGP.
[19]Transport Control Protocol

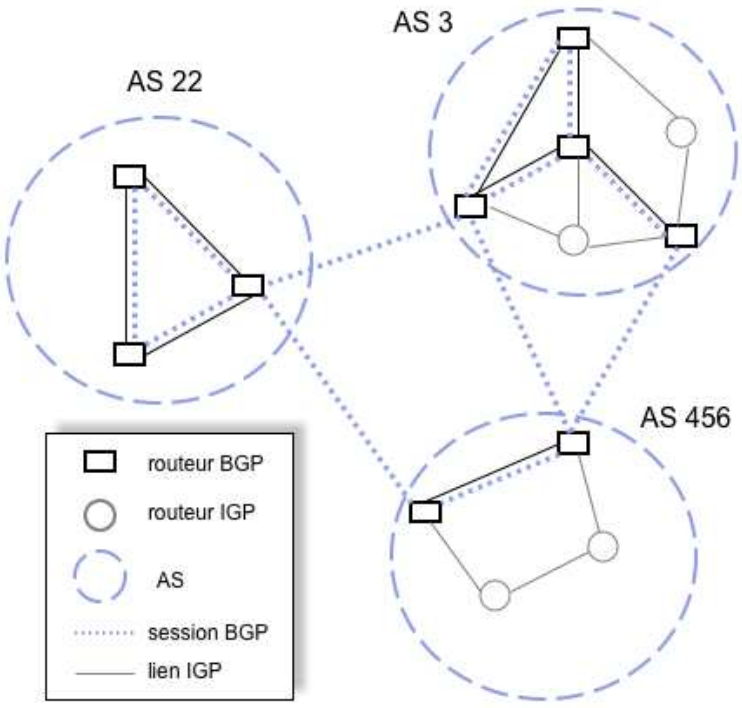


FIG. 1.4 – Un exemple de réseau BGP formé de 3 AS.

Information (NLRI)[20]. Les messages BGP échangés entre routeurs sont des annonces de routes représentant des promesses d'acheminement vers un NLRI ou des révocations de routes précédemment annoncées. Un chemin de routage BGP (ou route BGP) est constitué d'un ensemble d'attributs de natures très diverses qui permettent de décrire la route à destination du NLRI. Les attributs suivants sont obligatoires :

NLRI : le réseau IP destination,

Origin : la provenance protocolaire de l'annonce du NLRI,

AS_PATH : suite de segments qui correspond au chemin d'AS inverse parcouru par les messages de mise à jour BGP lorsqu'aucun segment n'est un AS Set,

Next Hop : l'adresse d'une liaison externe à l'AS[21] si le NLRI est extérieur à l'AS ou l'adresse IP de loopback[22] d'un routeur BGP de l'AS sinon.

Pour chaque NLRI annoncé par un routeur[23], des routes BGP vont se former de proche en proche. Chaque routeur va compléter et modifier chaque route reçue avant de la propager à ses voisins.

De nombreux attributs caractérisent une route BGP. La plupart des modifications apportées au protocole résident dans l'ajout de nouveaux attributs et des fonctionnalités correspondantes dans le coeur de traitement des messages. Suivant les cas iBGP ou eBGP, et suivant la version des routeurs, les attributs peuvent être, propagés, rejetés[24], ignorés[25] et modifiés. Les attributs BGP d'une route vont permettre en général deux fonctions essentielles du protocole :

Sélection : la discrimination de routes. Les attributs sont vus comme des critères de sélection. Lorsqu'un routeur a connaissance de plusieurs routes pour un même NLRI, les attributs de chacune des routes vont être comparés pour ne retenir qu'une unique[26] meilleure route (voir le processus de décision BGP page 1.2.1),

[20]Network Layer Reachability Information

[21]L'adresse de Next Hop BGP peut toutefois correspondre à l'adresse d'un routeur interne à l'AS lorsque l'option next-hop self est activée.

[22]Une adresse IP de loopback d'un routeur correspond à un des alias IP du routeur permettant de l'identifier au niveau du protocole BGP.

[23]Un routeur qui annonce un réseau est dit origine du NLRI, ainsi que son AS

[24]Certains attributs iBGP comme AS_CONFED_SET ou AS_CONFED_SEQUENCE qui représentent des chemins d'AS privés n'ont par exemple aucun sens au niveau eBGP, donc ils sont supprimés de la route avant propagation sur une session eBGP. L'attribut Community d'une route peut aussi par exemple être mis à zéro avant redistribution de la route.

[25]Un routeur peut propager un route avec des attributs qu'il ne comprend pas.

[26]En général unique. Mais bientôt certains routeurs BGP vont proposer d'exporter plusieurs routes pour un même NLRI destination.

Filtrage : le filtrage de routes. Les attributs sont vus comme des critères d'acceptation et de propagation. Lorsqu'une route est reçue par un routeur, celui-ci commence par vérifier s'il doit traiter la route. Lorsqu'une route est susceptible d'être annoncée vers un routeur voisin, le routeur vérifie si cette propagation est autorisée d'après les règles de la politique de routage (voir schéma 1.5(b)).

Le protocole BGP permet d'implémenter des règles au sein de chaque routeur afin de décider dans quels cas accepter, modifier, sélectionner et propager une route donnée. La figure 1.5(b) décrit le traitement d'un message de mise à jour BGP par un routeur. Le protocole BGP fonctionne en mode incrémental : un routeur reçoit continuellement des messages de mise à jour de routes BGP des autres routeurs connectés via une session. Chaque route reçue (il y a plusieurs routes par message) est traitée par le routeur qui applique les règles définies dans sa configuration en fonction des attributs BGP reconnus et de la session d'entrée. Il décide alors de conserver ou de rejeter la route. Si la route est acceptée, elle est ajoutée dans la Routing Information Base (Loc-RIB) du routeur. Puis le routeur applique le processus de décision BGP pour le NLRI de la route reçue et décide alors de la meilleure route parmi toutes celles qu'il connaît. Si la route a changé pour le NLRI, cette dernière est injectée dans la table de forwarding ou Forwarding Information Base (FIB), et la nouvelle meilleure route est candidate pour exportation vers les routeurs voisins. Pour chaque routeur voisin, le routeur modifie les attributs de la route (les règles dépendent de la route et de la session voisine), et si aucune règle n'interdit la propagation, il transmet cette nouvelle route BGP modifiée au routeur voisin. Un point important concerne l'échange des messages d'update BGP : un routeur n'exporte que son unique meilleure route BGP pour un NLRI donné à destination d'un autre routeur voisin[27].
Le processus de décision BGP, consiste à sélectionner une meilleure route à destination d'un NLRI parmi les routes disponibles. Ce processus décide de la meilleure route en plusieurs étapes successives. A chaque étape, si plusieurs routes sont ex aequo, le routeur passe à l'étape suivante. Ce processus de décision est différent suivant les constructeurs de routeurs. Par exemple pour Cisco, il existe un attribut *Weight* en plus du standard, mais ce dernier n'est pas propagé à l'extérieur de l'AS. Certains constructeurs laissent même la liberté de modifier certaines étapes du processus de décision standard, de placer des valeurs par défaut aux attributs, et de vérifier l'existence des routes pour d'autres

[27]Des modifications ont été proposées dans [182], mais le cas de l'export de plusieurs routes résout la plupart du temps des problèmes de convergence iBGP.

protocoles intra-domaines. Généralement, on retrouve les étapes suivantes :

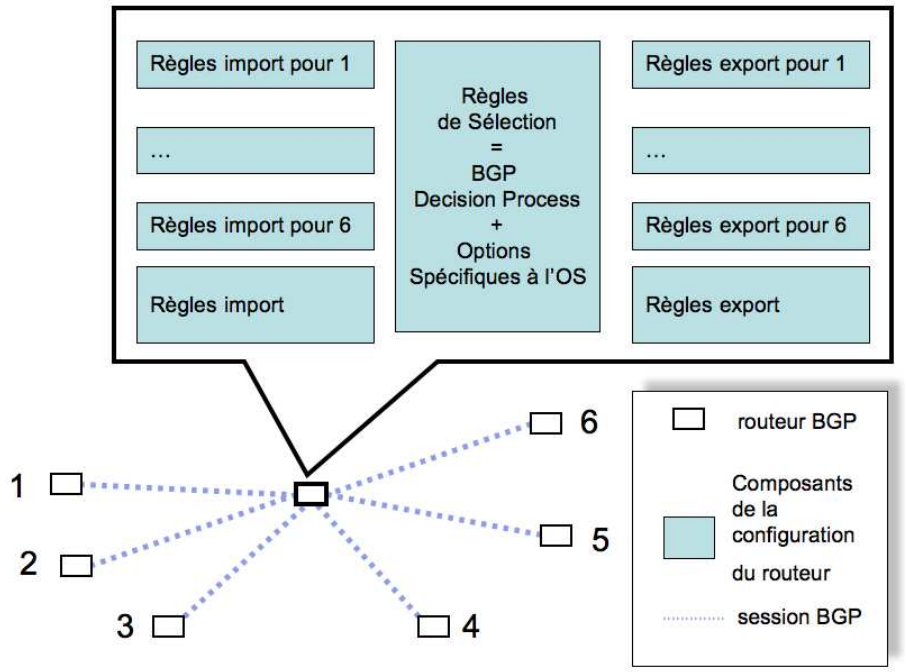
1. Sélectionner les routes avec une valeur *Local Pref* maximale.
2. Sélectionner les routes de longueur d'*AS_PATH* minimale.
3. Préférer les routes provenant d'un protocole intra-domaine (IGP), des routes provenant d'un protocole interdomaine EGP, et des routes de provenance inconnue.
4. Préférer les routes dont l'attribut *MED*[28] est le plus petit si les routes commencent par le même AS.
5. Préférer les routes connues par eBGP plutôt que par iBGP.
6. Préférer les routes dont le poids intra-domaine (IGP) vers le *next hop BGP* est minimal.
7. Préférer les routes dont l'ID[29] du routeur voisin est minimal (étape de tie break totalement discriminante).

Au processus de décision BGP s'ajoutent des règles vérifiant : la non existence d'un cycle d'AS, l'existence d'une route IGP vers le next hop BGP, la possibilité d'agréger[30] des NLRI contigus, la conservation de la route la plus ancienne... Les règles relatives aux agrégations de NLRI ou aux clusters iBGP[31] sont elles aussi spécifiques aux constructeurs.
L'ensemble des règles fixées au niveau d'un routeur représente la configuration de ce dernier (voir figure 1.5(a)). Chaque constructeur définit un langage d'expression de ces règles, mais il n'existe pas encore aujourd'hui de langage standardisé (cf. [73] pour plus de détails). On définit ainsi la politique de routage d'un AS comme l'ensemble des règles fixées au niveau des routeurs de l'AS. La politique de routage d'un AS est généralement définie de façon globale à l'AS. Au niveau de chaque routeur, des règles spécifiques sont fixées en fonction des liens avec les routeurs d'AS voisins.

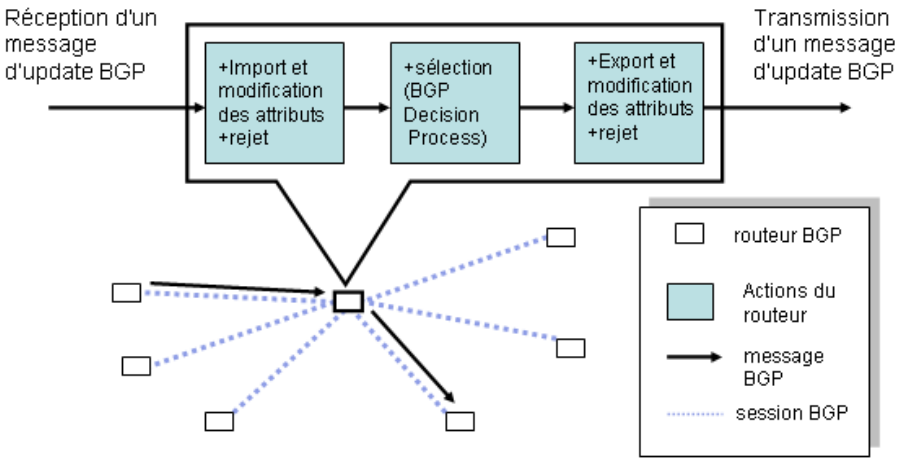
1.2.2 Le plan de contrôle d'un opérateur et configuration des routeurs

La configuration des routeurs d'un AS dépend de la manière dont on veut exploiter le réseau. Dans le cas de réseaux clients, si un AS est utilisé pour le routage Internet[32],

[28] Multi Exit Discriminator
[29] L'ID d'un routeur correspond en général à son adresse IP de loopback.
[30] On parle d'agrégation lorsqu'on regroupe différents NLRI destinations dans une même route.
[31] Un cluster iBGP identifie un groupe de routeurs BGP du même AS.
[32] Un réseau peut être invisible au niveau BGP lorsqu'il est rattaché de façon statique ou par un protocole intra-domaine à un opérateur. L'opérateur utilise un AS pour ré-annoncer les plages publiques



(a) Règles issues de la politique de routage d'un routeur BGP. Une route est traitée d'après la session BGP d'entrée, ses attributs BGP et la session BGP de sortie.



(b) Le traitement d'un message d'update BGP par un routeur. La sélection et l'acceptation d'une route BGP reçue par une session est fonction de la politique de routage. La sélection de la meilleure route BGP est effectuée par le BGP Decision Process. L'export de la meilleure route modifiée pour un préfixe dépend de la session BGP de sortie et des attributs de la route.

FIG. 1.5 – Fonctionnement d'un routeur BGP : réception des mises à jour incrémentales et application des règles de la politique de routage sur chaque route BGP reçue.

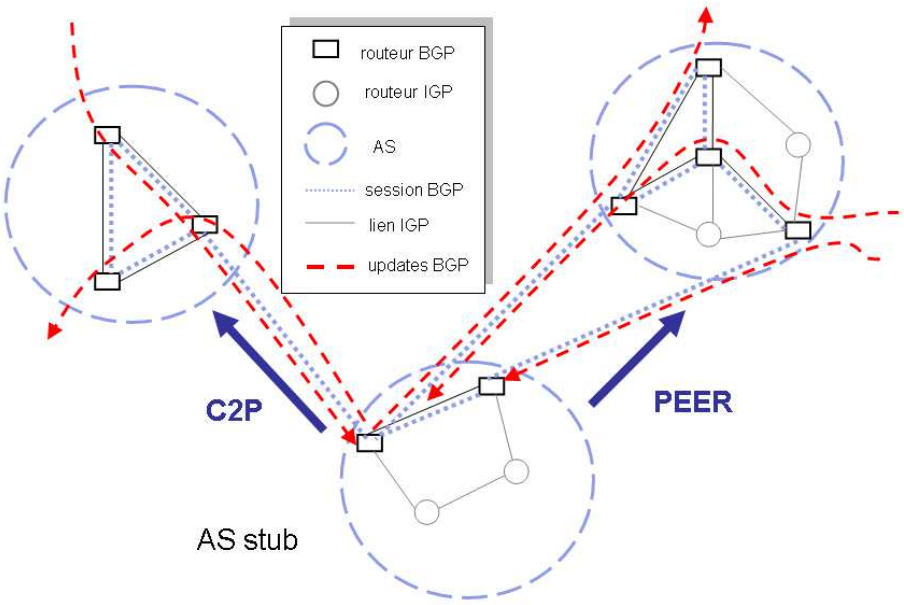
alors le protocole BGP va permettre de choisir le meilleur fournisseur (ou le meilleur AS à traverser) pour chaque préfixe destination et de choisir les fournisseurs permettant aux plages d'adresses du client d'être propagées vers l'ensemble du réseau. Un AS client final est souvent qualifié d'AS stub ou réseau stub (cf. figure 1.6(a)).
Dans le cas de l'AS d'un opérateur de réseau, on parle d'AS de coeur ou d'AS de transit. Par définition, un AS de transit transmet des messages BGP entre deux AS voisins (voir figure 1.6(b)). Lorsqu'un routeur d'un AS x annonce une route pour un préfixe p à un routeur d'un autre AS y , qui l'annonce ensuite à d'autres routeurs de l'AS y, pour finalement l'annoncer à un routeur d'un troisième AS z : l'AS y promet à l'AS z qu'il pourra acheminer les paquets IP en destination d'adresses de la plage p via l'AS x. Un AS de transit doit donc avoir une politique de routage avec des règles d'échange qui vont dépendre de nombreux paramètres, et en particulier de l'AS de provenance et de l'AS destination d'une route BGP afin de respecter les contrats économiques. En plus d'assurer un échange cohérent des messages au sein de la topologie des sessions iBGP, la traduction des politiques de routage économiques doit permettre aux routeurs voisins d'obtenir toutes les routes promises. Une session eBGP qui correspondant à un accord de peering ne doit généralement pas faire transiter des routes vers des préfixes obtenus par un AS en peering ou par un AS fournisseur. Créer et adapter une configuration BGP robuste requiert la connaissances des politiques de routage des AS voisins pour annoncer et sélectionner les routes voulues en fonction des attributs BGP, et la connaissance des différentes routes BGP susceptibles d'être propagées dans l'AS. Des outils comme [53] offrent de nouvelles possibilités évoluées pour une vérification simple des configurations d'un AS. D'un point de vue opérationnel, pour faire du tuning de configuration, il est très courant d'utiliser la méthode **itérative** suivante :

1. mise à jour de la configuration d'un routeur en fonction de la convergence théorique,
2. attente de la convergence du protocole,
3. observation des tables de routage et des routes disponibles,
4. correction des règles BGP,
5. retour à l'étape 1

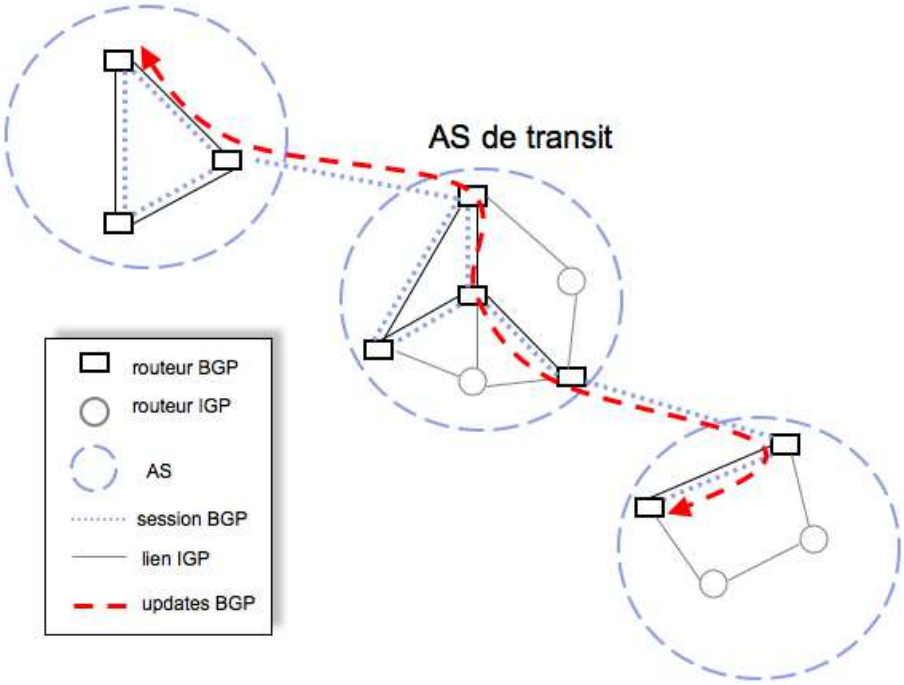
D'un point de vue global (contrôle, utilisation et management), on attend d'un réseau qu'il fonctionne mais aussi qu'il minimise les coûts d'achats du transit, qu'il permette d'obtenir

---

du réseau client.



(a) Exemple d'un AS stub. Les AS stub sont des clients finaux et ne transmettent pas les messages BGP entre leurs AS voisins (fournisseurs ou peer).



(b) Exemple d'un AS d'un opérateur de transit. Un AS de transit transmet des messages d'update BGP entre différents AS.

FIG. 1.6 – Les AS clients finaux (AS stub) et les AS de transit.

des délais les plus courts possibles, qu'il résiste bien aux pannes extérieures et intérieures[33], qu'il soit robuste aux dysfonctionnements[34]. De plus, les accords économiques conclus avec les AS voisins doivent être respectés. Les principales difficultés rencontrées lors de la configuration de routeurs BGP sont les suivantes :

– complexité du protocole interdomaine BGP. Pour un réseau BGP, il existe une multitude de façons de configurer les routeurs sans pour autant garantir la convergence des routages escomptés,
– connaissance non triviale des routages complets de ses propres équipements,
– connaissance très réduite des politiques de routage des autres AS. L'existence de politiques de routage indépendantes pour chaque opérateur rend de plus la simulation du processus de convergence très compliquée.

Finalement, les raisonnements sur les flux de trafic de bout en bout peuvent être vite très complexes à mettre en oeuvre dans une politique de routage qui passe à l'échelle, qui tient compte des différents types de routeurs d'un AS, et qui se déploie et se maintient facilement. Le protocole BGP n'a été conçu que pour assurer l'existence d'au moins une route par préfixe pour assurer la connectivité totale.

1.2.3 Ingénierie de trafic interdomaine

Les méthodes d'ingénierie de trafic interdomaine ou Inter-Domain Trafic Engineering (IDTE)[35] consistent à adapter les routages BGP pour rendre le réseau plus robuste, pour faire baisser les coûts et augmenter les performances. Un AS fait de l'IDTE en modifiant sa politique de routage. Le point essentiel dans ce type de méthode est donc la connaissance des réseaux et de leurs configurations (voir par exemple les travaux dans [51, 68, 123, 150, 194]). Il existe deux types majeurs d'ingénieries de trafic :

Ingénierie sortante : l'IDTE sortant consiste à correctement propager dans l'AS et sélectionner les routes BGP parvenant dans l'AS via des sessions eBGP. Il faut savoir quels attributs BGP fixer aux différentes routes BGP pour indiquer des préférences. La connaissance approfondie des routes propagées par les différents AS voisins est utile pour déterminer les meilleures routes qui seront

[33] reconvergences rapides pour éviter les chemins avec des boucles en cas de pannes de routeurs, de changements de routages IGP ou de pannes et maintenances de sessions BGP

[34] Dans le sens où les filtres concernant les routes BGP annoncées par les AS voisins sont correctement réglés.

[35] Inter-Domain Traffic Engineering

sélectionner par le processus de décision BGP et installées dans la FIB des routeurs de l'AS.

Ingénierie entrante : l'IDTE entrant consiste à correctement configurer les annonces de NLRI afin que les AS voisins et distants sélectionnent préférentiellement ou non l'AS et quel routeur d'entrée pour chaque préfixe.

La connaissance des politiques de routage des AS voisins permet de détecter des erreurs de configuration (en particulier les Gadgets BGP), améliorer les délais de bout en bout (en sélectionnant de meilleures routes), éviter les oscillations et les reconvergences BGP trop longues.

L'optimisation des politiques de routage d'un AS et l'IDTE sont en pratique souvent résolus en fixant divers paramètres BGP (quelquefois redondants les uns par rapport aux autres). En particulier, les administrateurs utilisent des fonctions hétérogènes des routeurs[36] et modifient le maillage des sessions iBGP dans l'AS. Les configurations peuvent être instables, souffrir de *deflection*[37], avoir des filtres inadaptés aux situations de secours. Le maillage des sessions BGP peut rendre la convergence du protocole très lente ou induire une montée en charge inattendue des routeurs. La connaissance a priori de la topologie interdomaine et la connaissance partielle des politiques de routage des AS peut aider à améliorer la robustesse de la configuration d'un AS.

1.3 Mesure et connaissance du routage Internet : un état de l'art

La mesure du réseau Internet se fait généralement par le biais de sondages ou de "traces" dans le réseau. Ces méthodes d'exploration et d'exploitation des chemins de routage dans le réseau portent parfois le nom de tomographie de l'Internet. La coopération des équipements de routage rend difficile la mesure de l'Internet. De plus, le nombre de points d'observation est souvent négligeable par rapport à la taille du réseau. Finalement, la dynamique à plusieurs niveaux de granularité du réseau s'ajoute comme un bruit dans les observations. Suivant le niveau de détail considéré, les problèmes et les outils de mesure

[36]Par exemple l'option *next-hop self*, les comparaisons de MED pour différents AS, l'attribut Metric de Cisco,...

[37]Si un routeur BGP choisit un routeur de sortie (next hop BGP) en fonction du poids intra-domaine, et qu'un routeur BGP placé sur la route de sortie vers le next hop BGP choisit un autre routeur de sortie, alors le choix du premier routeur est problématique.

sont différents et le passage de l'échelle IP à l'échelle des AS est approximatif.
On a exploré plusieurs méthodes pour mesurer le réseau interdomaine pendant la thèse. Dans cette section, on montre pourquoi on choisit de mesurer le réseau Internet à la précision AS et préfixes au détriment de la précision routeur obtenue par tomographie IP. On propose d'évaluer les principales difficultés rencontrées dans la mesure du routage Internet en 1.3.1, puis on étudie un état de l'art concernant la mesure de l'Internet. On se limite cependant au réseau Internet d'un point vue du routage unicast[38] des adresses IPv4 publiques.

1.3.1 Difficultés de la tomographie de l'Internet

a) Dynamique du réseau

Le réseau Internet est en constante évolution. Que ce soit au niveau du maillage des routeurs, des intervenants, des équipements, des adressages, des configurations des routeurs. Les protocoles de routage (notamment BGP) permettent de prendre en compte cette dynamique. Lorsqu'on tente de mesurer le réseau Internet, deux modes opératoires se distinguent pour prélever l'information :

Sondage actif : le sondage actif d'un réseau IP consiste à interroger en temps réel les équipements et en particulier les routeurs afin d'obtenir différentes informations qui varient au cours du temps. Le sondage actif regroupe en particulier les mesures SNMP[39] (pour les mesure des trafics ou l'accès aux configurations), l'accès en temps réel aux routeurs (exécution de commandes sur le routeur), l'envoi (local ou sur des routeurs distants) de paquets UDP[40], TCP[41] et ICMP[42] (traceroute[43], ping[44]...), et l'interrogation répétée de bases de données publiques comme les registres de routage Internet ou les serveurs DNS[45].

Sondage passif : le sondage passif du réseau Internet consiste à analyser des informations qui varient au cours du temps mais qui sont collectées sur des équipements

[38] Le routage unicast correspond à l'acheminement de données d'un hôte source vers un unique hôte destination.
[39] Simple Network Management Protocol
[40] User Datagram Protocol
[41] Transmission Control Protocol
[42] Internet Control Message Protocol
[43] La commande traceroute permet de récupérer une suite d'adresses IP correspondant au chemin de routage entre un hôte source et un hôte destination.
[44] La commande ping permet de savoir si une adresse IP correspond à une machine du réseau.
[45] Domain Name System

à intervalles fixes. On inclut aussi la consultation de données publiques prétraitées (comme certaines provenant des registres de routage Internet). On pourra noter que le sondage passif est au sondage actif ce que la photo est à la vidéo.

b) Limitations générales des méthodes de mesure

En plus de la dynamique d'évolution du réseau, s'ajoutent plusieurs problèmes liés à la mesure de l'Internet.

- La taille du réseau est un point essentiel pour les temps de mesure, le stockage et les temps de traitements (il existe des millions d'adresses IP, des centaines de milliers de routeurs et plus de 20 000 systèmes autonomes).
- La coopération des équipements n'est pas toujours adéquate : un administrateur n'a un accès privilégié que pour son propre réseau. De plus, la mise en place d'instruments et d'interfaces de mesure n'est pas systématique, coûte cher, et alourdit le contrôle et l'administration du réseau.
- La pertinence des informations peut induire en erreur. Les informations déclaratives comme celles proposées par les registres de routage Internet sont souvent incomplètes et non systématiques. Les noms DNS des routeurs peuvent aussi se révéler erronés ou même complètement obsolètes.
- Les modalités de mesure elles-mêmes conduisent aussi à des incohérences et des vue incomplètes. Il arrive souvent de n'avoir qu'un sous-ensemble des informations dont on a besoin (comme un nombre limité de sondes BGP ou traceroute) et les mesures peuvent concerner une période de temps non négligeable face à la fréquence de leur variation.

c) Distinction entre mesure et inférence

L'extraction d'informations concernant les opérateurs Internet n'est pas directement une opération de mesure. Il faut bien différencier les informations mesurées/observées des informations déduites. Les méthodes d'inférence permettent de déduire des informations sans garantir leur validité. Par exemple les accords d'interconnexion entre opérateurs, la hiérarchie des opérateurs Internet, les routages recalculés entre AS, les pratiques de filtrage ou de sélection de routes BGP, la localisation d'une adresse IP, la correspondance entre une adresse IP et un AS,...

1.3.2 Mesure et inférence de topologies de réseaux à la couche IP

Les outils de mesure usuels des topologies de réseaux IP sont les commandes ping et traceroute. On peut se reporter à l'annexe A page 220 pour des explications détaillées concernant le fonctionnement de ces deux outils. La mesure de topologies de réseaux au niveau IP regroupe de multiples opérations permettant l'obtention de différentes informations. Nous proposons d'expliquer quelques problèmes de mesure et d'inférence importants :

– Mesure de la topologie d'un réseau dont on fait partie (sondage actif).
– Mesure de la topologie d'un réseau de l'extérieur (sondage actif).
– Mesure de la topologie de l'Internet à la granularité adresse IP (sondage actif).
– Résolution d'interface pour trouver les adresses IP d'un même routeur (sondage actif).
– Inférence de l'AS associé à un routeur (sondage passif).

Le lecteur pourra se référer aux travaux [31, 69, 76, 105, 159, 167] pour d'autres états de l'art concernant la mesure de l'Internet.

a) Mesure d'un réseau sous environnement coopératif

La mesure d'une topologie de réseau IP sous environnement coopératif est assez simple en général. Tout dépend des accès possibles aux machines. Par exemple, grâce au protocole SNMP on peut souvent récupérer les différentes adresses IP d'un routeur et les liens avec ses voisins (voir [8, 9, 21] pour des exemples). On peut aussi utiliser des requêtes ICMP pour interroger les machines (traceroute et ping). L'utilisation de ces outils de mesure sous environnement coopératif fonctionne souvent bien (les traceroutes TCP sur des ports usuels sont très efficaces) mais dépend du niveau de sécurité des équipements du réseau et de la hiérarchie des routages lorsqu'on l'on dispose de peu de sources.La découverte de réseaux avec adresses anonymes a été abordée dans [193]. Il existe beaucoup de façons d'explorer un réseau en inférant une connaissance progressive des adresses IP des interfaces. On pourra consulter les travaux [43, 161].

La coopération des équipements est un point clé de la mesure d'un réseau IP. Dans le paragraphe suivant nous montrons la complexité de la mesure de réseaux de l'extérieur.

b) Mesure du réseau d'un ISP

Il existe peu de travaux qui se focalisent sur la découverte précise du réseau d'un ISP sans la découverte du réseau tout entier. On considérera ne pas être sous environnement coopératif, et que les seules mesures actives sont l'envoi de paquets ICMP, UDP et TCP.

Les travaux de Neil Spring [168] montrent très bien la complexité de l'obtention de cartes d'adresses IP. De telles mesures se heurtent aux problèmes suivants :

- Le routage BGP entre AS, qui prend en compte les politiques de routage BGP, ne rend pas accessible tous les cheminements topologiques IP possibles.
- Il faut disposer d'une plate-forme de mesure comme [99, 165, 170].
- Il faut disposer d'un bon placement des sondes, et bien choisir les destinations (dans le réseau de l'ISP ou non).
- Certaines adresses de routeurs ne répondent pas aux requêtes envoyées et d'autres adresses ne sont jamais connues.
- Les adresses IP peuvent appartenir à des réseaux différents [105].
- Le problème de résolution des alias IP d'un même routeur est très difficile [169].
- L'inférence de la géographie des routeurs peut se faire de différentes manières approchées (l'outil undns de Neil Spring ou l'outil IPtoGeO), mais aucune méthode n'est réellement déterministe.
- L'inférence des poids intra-domaine entre les routeurs dépend du nombre de traces, du nombre de noeuds et de liens découverts [168] alors que la résolution n'est pas un frein.

Les problématiques de mesure du réseau d'un ISP à la précision adresses IP, se retrouvent dans la tomographie de l'Internet tout entier.

c) Inférence de la topologie adresse IP de l'Internet

La mesure de l'Internet par traceroute, aussi appelée tomographie de l'Internet, est un sujet vaste qui a donné naissance à beaucoup de projets de grande ampleur. On peut distinguer plusieurs types de travaux dans la mesure de l'Internet :

- les travaux orientés théorie de l'exploration de grands réseaux, par exemple [35, 43, 118, 146, 146, 157, 179],
- analyse des graphes mesurés [24, 115, 115, 171, 180],
- les travaux permettant une meilleure compréhension des incohérences et des problèmes de mesure, par exemple [23, 104, 112, 161, 181],
- les infrastructures de mesure [19, 71, 99, 148, 160, 165, 166]

Les observations majeures de ces dernières années sont les suivantes.

- Les adressages des routeurs sont souvent multipliés [105].
- La mesure de la topologie d'adresses IP de l'Internet est fortement biaisée (du fait de la méthode de sondage) [33, 77, 114, 157, 193].
- Le nombre de sources nécessaires à la mesure peut être très inférieur au nombre de

destinations pour avoir un échantillonage correct [11, 83].

– La topologie de l'Internet semble suivre une loi de puissance à la granularité IP [49, 128, 174].

A noter que la prise en compte des contraintes de routage dans les modèles d'exploration de l'Internet n'est pas encore très bien traitée du fait de sa méconnaissance.

d) Inférences des différents interfaces IP d'un routeur

A partir d'un graphe d'adresses IP (où chaque adresse est celle de l'interface d'un routeur), on peut inférer de façon incertaine un graphe de routeurs. En effet, les adressages des routeurs BGP peuvent être très complexes comme indiqué figure 1.7. Connaître les différentes adresses pour un même routeur est fondamental si on veut caractériser la topologie IP réelle de l'Internet. En effet, il existe beaucoup de travaux théoriques qui traitent de cartes d'adresses IP sans pour autant maîtriser la topologie réelle correspondante (qui est importante). Les principales méthodes pour reconnaître les différentes adresses d'un même routeur utilisent par exemple des requêtes ping, des combinaisons de requêtes pour analyser les champs IP, les noms DNS des routeurs, la topologie de la carte d'adresse...On pourra se reporter aux travaux [23, 71, 165, 167–169, 177, 193] pour de plus amples détails.

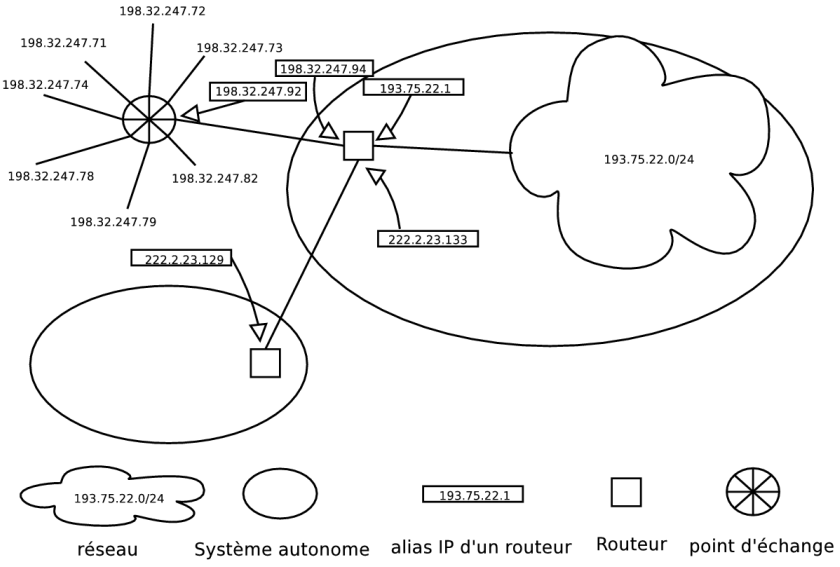


FIG. 1.7 – Exemple où un routeur a plusieurs interfaces IP dans des plans d'adressage différents

e) Inférence de l'AS d'une adresse IP dans un traceroute

Les problèmes d'adressages discutés au paragraphe précédent rendent aussi très difficile l'inférence de l'AS associé à un routeur. En effet, conformément à la réalité, les routeurs de bordure d'un AS et les routeurs au niveau des points d'échanges utilisent les adresses en provenance de différentes plages. Ce fait peut être exploité par exemple pour illustrer les différents AS connectés sur un point d'échange [192]. Mais ce problème d'adressage multiple, cumulé aux problèmes de routeurs anonymes dans les traceroutes, rend très difficile la conversion d'un traceroute en chemin d'AS. On se reportera aux travaux suivants pour de plus amples détails [6, 7, 26, 98, 104, 105, 108, 126, 154, 167, 186].
La topologie du réseau interdomaine peut donc être inférée d'après des cartes de traceroute agrégées au niveau AS. Mais le niveau d'incertitude assez fort dans les méthodes d'inférence, la faiblesse des outils de mesure même en dépit d'une coordination complète des équipements, les limites théoriques de la technique de sondage, et la lourdeur des mesures à effectuer sont des freins pour l'obtention mais aussi pour l'exploitation des données. On utilise par la suite une méthode beaucoup plus adéquate pour obtenir la topologie de l'Internet interdomaine et pour la traiter.

1.3.3 Mesure du routage Internet inter-opérateurs avec le protocole BGP

On reporte ici un état de l'art concernant la mesure et l'analyse globale des politiques de routage inter-AS. Pour mieux comprendre le procédé de mesure, on se reportera au chapitre 2 de la thèse.

a) Mesure de la topologie du réseau interdomaine avec le protocole BGP

Une façon très naturelle de mesurer la topologie de l'Internet à la granularité AS est d'utiliser le protocole BGP. En effet, seul le protocole BGP exploite la notion d'AS pour effectuer les routages. De plus ce protocole à vecteurs de chemins[46] utilise les séquences d'AS traversés dans les messages protocolaires[47]. Le procédé de sondage consiste à utiliser les informations BGP en provenance du multiples routeurs pour en déduire un ensemble

[46] un protocole à vecteur de chemins conserve le chemin de signalisation utilisé par un message échangé entre les routeurs.

[47] Les chemins d'AS sont utilisés dans les messages pour éviter les boucles, pour compter le nombre d'AS, et pour utiliser des filtres dans les politiques de routage.

de liaisons entre AS. L'attribut *AS_PATH* correspondant au chemin d'AS permet en effet l'inférence de liens logiques entre les AS en séquence dans le chemin. Les autres attributs BGP indiquent quelques informations supplémentaires sur les routes. Mais en général, soit les attributs concernent la politique de routage locale à l'AS et leur connaissance est limitée à quelques AS, ou alors on ne peut pas attribuer l'AS responsable de l'ajout ou de la modification des attributs de la route (l'attribut community[48] est un bon exemple). Différentes sources de données BGP publiques sont reportées en [84, 97, 135, 140, 187] et en [93, 178] pour les looking glass publics. Pour des exemples de topologie on pourra télécharger les informations parmi les historiques [86, 88, 90]. La mesure de la topologie interdomaine par le protocole BGP a succité beaucoup d'attention [28, 29, 39, 42, 90, 121, 143, 174, 196, 200]. La question de savoir si la vision de quelques tables BGP est vraiment exhaustive est encore en suspend car le procédé d'exploration donne un extrait biaisé [146, 157] de la topologie réelle.
Pour une meilleure connaissance des administrations des AS, on pourra se reporter en [189].

b) Analyse de la topologie interdomaine de l'Internet

La topologie interdomaine est très étudiée dans la littérature. Les travaux pour tenter de mieux la comprendre sont nombreux [10, 18, 47, 82, 109, 116, 119, 131, 132, 144, 145, 180, 195, 199]. Il existe une multitude de générateurs de graphe [64, 120, 202, 203] qui permettent de capturer différentes propriétés empiriquement identifiées. Par exemple : le clustering[49], la distribution en nombre de bonds[50], la centralité[51], la connectivité rich club[52] [201], la pertinence des D-K graphes aléatoires[53] [118], les K-core [106]... La distribution des degrés des AS en loi de puissance a attiré l'attention de la littérature (voir [30, 49, 88, 107, 128, 163, 195] et [176]), sachant de plus que de nombreux modèles permettent de générer des graphes aléatoires avec une loi de puissance sur les degrés du même type.

[48] L' attribut BGP community correspond à un nombre de taille 32 bits. Chaque routeur BGP peut ajouter et supprimer autant d'attributs community qu'il le souhaite dans une route BGP.

[49] Les méthodes de clustering consistent à scinder un ensemble d'informations en plusieurs groupes.

[50] La distribution en nombre de bonds consiste à connaître le nombre de destinations accessibles à un nombre de sauts quelconque en partant d'une source donnée.

[51] La centralité, proposée par Freeman en 1979 [54], est une mesure qui comptabilise les voisins de chaque noeud dans un graphe, avec les autres noeuds du système étudié.

[52] La connectivité rich-club indique le nombre de connectivités entre un noeud et tous ses noeuds voisins.

[53] L'analyse en séries de D-K graphes constitue une nouvelle approche systématique pour analyser des graphes et en particulier des topologies de réseaux. La méthode consiste à étudier les corrélations de degré pour les sous-graphes de taille au maximum K.

c) Faiblesses dans la connaissance des politiques de routage BGP

Les politiques de routage des opérateurs sont issues de contraintes économiques et techniques. La littérature est abondante concernant le routage interdomaine avec le protocole BGP [20, 22, 32, 81, 102, 152] et sa stabilité théorique [50, 62, 73, 74]. Quelques problèmes précis seront abordés dans le chapitre 2 et dans l'annexe Bcomme par exemple l'analyse des annonces de NLRI [134, 198, 198].

La mesure et l'analyse des politiques de routage au sens large est un problème qui a attiré beaucoup d'attention ces dernières années [27, 149, 162, 183, 184]. Deux phénomènes très important ont été en particulier mis en évidence :

Stabilité des tables de routage BGP : l'analyse des tables BGP et de la stabilité des routages comme par exemple dans [70, 155] a donné naissance à des travaux concernant la simulation du routage Internet et son inférence [79,125,133]. De manière générale, les modèles adoptés souffrent d'un manque d'information concernant les politiques de routage des opérateurs et notamment les accords d'interconnexion.

Inflation des routages : les chemins BGP sont souvent plus longs que le plus court chemin disponible car les politiques de routage influencent les "détours" dans les routages inter-AS [61, 164, 175]. Les accords d'interconnexion sont en particulier responsables de l'inflation des routages.

Inférence des accords d'interconnexion entre AS : les accords d'interconnexion jouent un rôle important dans les politiques de routage car les règles qu'ils impliquent dans la redistribution des routes BGP sont les principales règles de rejet dans les configurations. Le problème d'inférence des accords d'interconnexion est un problème d'optimisation industriel qui a succité beaucoup d'attention de la part de la littérature. Pour un état de l'art complet, on se reportera dans l'ordre chronologique aux travaux introductifs [60, 78, 173], puis aux travaux concernant la complexité du problème [14,40,46], puis aux récents travaux proposant des méthodes heuristiques de résolution [15, 41, 127, 129, 130, 162, 190, 191]. Le problème d'inférence des accords d'interconnexion entre AS sera approfondi durant cette thèse dans le chapitre 3. Ce problème joue un rôle central dans l'étude des politiques de routage interdomaines car les différents accords influencent les procédures de mesure et d'analyse de la topologie de l'Internet en dépit des règles de routage mises en place. La sécurité des routage, l'aide à la

configuration de routeurs et l'ingénierie de trafic, les problématiques d'aide à la décision économique entre AS, la modélisation de la topologie de l'Internet et sa mesure efficace, la compréhension des routages interdomaines avec BGP et son évolution [122, 182], la simulation du réseau Internet sont autant de problèmes pour lesquels la connaissance des accords d'interconnexion joue un rôle déterminant.

Hiérarchie des AS : la hiérarchie des AS a pu être déduite dans [66, 89, 109, 173] grâce aux relations client/fournisseur et peering alors inférées.

Dans le chapitre 2 on présente une nouvelle méthode de mesure de la topologie de l'Internet qui permet d'obtenir une vision statique et idéale des routages interdomaines BGP. Les éléments topologiques du graphe inter-AS déduits, sont ensuite analysés. En particulier, les matrices de politique de routage (voir définition 10) observées seront utilisées dans le chapitre 3 pour inférer les accords d'interconnexion économiques entre AS, et dans le chapitre 4 pour modéliser les routages inter-AS.

Chapitre 2

Observation des politiques de routage économiques avec une tomographie BGP

A partir des configurations des routeurs BGP d'un AS, on ne peut obtenir que la connaissance d'une seule politique de routage (celle de l'AS concerné). Pour être en mesure d'observer l'ensemble des politiques de routage, il faudrait connaître les configurations de tous les routeurs BGP du réseau Internet. Cette connaissance est impossible. Au mieux un opérateur pourra accéder aux données de routage de son(ses) AS(s). Il n'aura pas d'accès direct aux données de routage des routeurs des autres ASs. Les politiques de routage doivent donc être inférées. La construction d'une **topologie interdomaine** ainsi que **la modélisation des politiques de routage** sont deux pré-requis à l'inférence. Ces tâches doivent être effectuées avec soin car elles influent directement sur la connaissance souhaitée : les résultats de l'inférence. Le niveau de détail et la fiabilité des informations récoltées doivent être nécessaires et suffisants pour pouvoir appliquer des méthodes d'optimisation efficaces. Les sources de données utilisées pour construire la topologie du graphe des AS doivent donc être sélectionnées et filtrées avec précaution. C'est pourquoi dans ce chapitre on commence par analyser l'intérêt des diverses sources de données à notre disposition. L'échelle de détail des modèles et les hypothèses utilisées doivent aussi être cohérents avec la nature des informations récoltées. On introduit de nouveaux objets dans cette thèse : **les matrices de politique de routage**. Ces matrices sont les éléments de base permettant d'inférer les accords d'interconnexion entre AS dans le chapitre 3 et les chemins interdomaines dans le chapitre 4.

Nos contributions dans ce chapitre concernent les points de modélisation suivants :

La modélisation des politiques de routage à la granularité AS : On introduit de nouveaux éléments topologiques : les atomes de politique de routage de transit-préfixe, de transit et d'annonce. Ces éléments topologiques fournissent une précision intermédiaire entre l'observation de préfixes le long de chemins d'AS et la topologie inter-AS avec les préfixes appartenant à chaque AS.

La mesure d'une topologie AS-AS stable : On propose une nouvelle méthodologie, pour construire une topologie de l'Internet à la granularité interdomaine. Cette méthode repose sur la collecte et le filtrage de tables de routage BGP. On collecte une table par sonde et par jour pendant un mois. On appelle tomographie BGP l'ensemble des routes ainsi collectées.

Une caractérisation simple du biais de mesure de la tomographie BGP : On étudie l'évolution de la taille de la topologie AS-AS construite et du nombre d'atomes de politique de routage observés en fonction du nombre de tables de routage collectées. On montre que l'exploitation des tomographies ne permet d'avoir qu'une vision biaisée des préfixes et un vision partielle des atomes de politique de routage.

Une analyse des routages similaires à la granularité AS : On montre que les politiques de routage à la granularité AS sont simplifiables dans le sens où de nombreux routages sont similaires.

On propose aussi en fin de chapitre l'inférence des deux propriétés économiques suivantes :

L'inférence de la hiérarchie des AS : Les atomes de politique de routage de transit-préfixe sont utilisés pour déterminer un indicateur de la taille de chaque AS. Cet indicateur est calculé en tenant compte du biais de mesure introduit par le placement des AS-sondes. Une structure hiérarchique des AS en trois niveaux est alors inférée en déterminant les AS qui appartiennent à la périphérie du routage BGP.

La géolocalisation des NLRI et des AS : A l'aide d'une structure de donnée formée d'un double arbre et des informations enregistrées dans les registres de routage Internet, on peut inférer les pays potentiels des NRLI annoncés par chaque AS.

2.1 Obtention des politiques de routage stables à la granularité AS

2.1.1 Choix de la nature des données

On porte une attention particulière sur la pertinence et la cohérence des données pour pouvoir utiliser ensuite nos algorithmes d'optimisation qui infèrent les accords d'interconnexion. On a répertorié différents procédés et sources de données suivant la nature du sondage dans le réseau (actif ou passif) et suivant la provenance des mesures (plan de gestion, plan de contrôle, ou plan transfert). L'expérience montre qu'il est préférable d'utiliser les tables de routages pour obtenir des topologie d'AS avec un maximum de fiabilité. Enfin, on observe une stabilité sous-jacente dans les topologies mesurées.

a) Plan de gestion du réseau ("Management Plane")

Sondage actif des registres de routage : extraction des informations d'allocation des NLRI et des AS

Les registres Internet (ou IRR) sont les bureaux d'enregistrement régionaux des ressources de l'Internet. Ils offrent un accès publique aux informations administratives concernant la plupart des AS (en théorie tous) et des NLRI. L'accès à leurs bases de données se fait via un service spécifique dénommé whois (la liste des serveurs reportée dans [96]). Ces serveurs doivent être interrogés avec modération afin de ne pas mettre en péril leur bon fonctionnement[1]. Certains registres donnent parfois des indications sur la politique de routage des AS[2]. On peut en déduire dans ce cas les règles de routage de nombreux AS fournisseurs. Ces règles permettent notamment la détermination de liens logiques entre AS[3]. Pour plus de détails, on peut voir les travaux dans [29, 88, 121] et les outils présentés dans [16, 162]. Signalons que les informations contenues dans ces registres de routage ne sont que déclaratives. Elles peuvent se révéler incomplètes ou obsolètes. La saisie manuelle de ces informations peut induire des erreurs[4]. En général, les travaux de la littérature se limitent à l'exploitation des informations sur les liens logiques entre AS

[1]Les requêtes sont effectuées AS par AS, ce qui rend la procédure d'autant plus longue. Mais, on peut parfois récupérer les sauvegardes de certain registres sur leurs sites FTP par exemple dans [55].

[2]Le format RPSL (Routing Policy Specification Langage) est utilisé par les registres pour décrire les politiques de routage de chaque AS. Seulement certains registres utilisent ce format [4, 5]

[3]Ces liens peuvent être absents des données BGP publiques.

[4]Le format des données n'est d'ailleurs pas toujours correctement respecté.

dont l'extraction reste simple. Certains graphes d'AS obtenus d'après ces registres sont proposés dans [16, 88, 90].

b) Plan de contrôle du réseau ("Control Plane")

Le plan de contrôle d'un réseau correspond aux informations de transport et de signalisation utilisées pour le routage.

Sondage passif des données de routage BGP : historique de messages d'update et de tables de routage BGP

Pour obtenir des données de routage BGP, il suffit d'établir des sessions eBGP entre un routeur sonde et les routeurs des différents AS pour lesquels l'information de routage est recherchée. Le routeur sonde est alors utilisé comme un "trace collector". Il permet de connaître à chaque instant les meilleures routes BGP de chacun des routeurs voisins[5]. Un opérateur peut aussi placer un routeur dans le full-mesh[6] du maillage iBGP de son AS pour récupérer les données provenant de tous les routeurs de bordure (en supposant la configuration de l'AS correcte). Le sondage passif du réseau BGP consiste à récupérer périodiquement les messages protocolaires BGP et les tables de routage BGP ainsi constituées.
Des opérateurs de réseau [84, 97] et des projets publics comme Route-Views [140], RIPE [135] et Intel-Dante [187] mettent à disposition ces informations (messages de mise à jour et photos périodiques des tables de routage).

Sondage actif BGP : interrogation de looking glass BGP

Le principe du sondage actif BGP consiste à interroger des routeurs BGP en temps réel. Ces routeurs peuvent être interroger via le protocole telnet, ou par le biais de pages web (voir les sites suivants [178]). On parle de routeurs ou serveurs "looking-glass". Les commandes suivantes permettent de récupérer des informations sur le routage BGP d'un réseau via un looking-glass :

"show ip bgp summary" permet de connaître la liste des routeurs voisins avec lesquels le routeur est configuré pour établir des sessions. En particulier, l'AS est indiqué pour chaque session,

[5]Les messages de mise à jour reçus permettent de reconstruire partiellement ou totalement la vision du routage des routeurs voisins.

[6]Le full mesh iBGP d'un AS correspond au maillage complet entre les routeurs BGP dans un même AS.

"show ip bgp a.b.c.d/e " permet de connaître les routes BGP disponibles à destination du NLRI *a.b.c.d/e* spécifié,

"show ip bgp community" permet d'obtenir une partie de la table de routage d'un routeur BGP[7].

"show ip bgp" permet d'obtenir la table de routage complète d'un routeur BGP , en fait toutes les entrées dans sa Loc-RIB, voir figure 2.1). Cette commande est souvent interdite car elle peut surcharger un routeur pendant des dizaines de minutes.

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 0.0.0.0          193.10.68.101                125      0 2603 i
*> 15.212.0.0/17    193.156.90.16                         0 3292 15412 9304 151 i
*> 62.24.229.0/24   193.156.90.16                         0 3292 3344 i
* i                 128.39.0.89                0    100      0 3292 3344 i
*> 62.24.230.0/23   193.156.90.16                         0 3292 3344 25396 29129 i
* i                 128.39.0.89                0    100      0 3292 3344 25396 29129 i
*> 62.26.0.0/15     193.156.90.51            163           0 3257 12312 i
*> 62.43.252.0/23   193.156.90.51            411           0 3257 28842 29119 i
* i62.44.64.0/19    128.39.0.89                0    100      0 2119 8210 8406 i
*>                  193.156.90.2                          0 2119 8210 8406 i
*  62.44.192.0/18   193.156.90.30               9          0 6667 6667 6667 6667 15501 i
*                   193.156.90.2                          0 2119 8434 16086 15501 i
*>                  193.156.90.39           1000          0 8434 16086 15501 i
* i                 128.39.0.89             1000    100      0 8434 16086 15501 i
*> 62.56.238.0/23   193.156.90.11               0          0 5377 i
*> 62.56.252.0/22   193.156.90.51             331          0 3257 13264 8584 20565 i
* i62.61.160.0/19   128.39.0.89                0    100      0 3292 15412 8966 8966 8966 8966 8961 8529 28885 i
*>                  193.156.90.16                         0 3292 15412 8966 8966 8966 8966 8961 8529 28885 i
```

FIG. 2.1 – Exemple d'une trace d'exécution de la commande « show ip bgp » sur un looking-glass.

c) Sondage actif du plan de transfert ("Data Plane")

Le plan de transfert d'un réseau IP est responsable de l'acheminement des paquets dans ce dernier. Le sondage actif du réseau d'un opérateur, ou de l'Internet tout entier, au niveau plan de transfert, consiste à exécuter des commandes traceroute depuis un grand nombre de routeurs à destination de milliers d'adresses IP. Une trace obtenue par la commande traceroute contient la séquence des adresses IP des interfaces des routeurs qui ont été traversés pour atteindre la destination. Ce chemin d'adresses IP doit ensuite être agrégé en chemin d'AS. On obtient alors des listes de chemins d'AS (voir [16] pour plus de précision).

Toutefois, pour effectuer le sondage actif d'un réseau dans son plan de transfert, il faut disposer de beaucoup de sources. Celles-ci doivent être bien positionnées afin de pouvoir

[7]En fait toutes les entrées dans la Loc-RIB qui comportent une communauté BGP

emprunter un maximum de routes disjointes. De plus, la sélection de l'adresse destination d'un traceroute influence le chemin emprunté en fonction du réseau BGP le plus spécifique contenant cette adresse (cf. [168] pour plus de détails). Cependant, on dispose de peu de sondes ou de looking-glass publiques à partir desquels effectuer des commandes traceroute. Finalement, la correspondance entre adresse IP d'un routeur et numéro de l'AS auquel il appartient nécessite l'utilisation de données supplémentaires (IRR notamment). Par conséquent, il existe une forte incertitude quant à la validité des chemins d'AS obtenus.

d) Comparaison des topologies d'AS obtenues à partir des différentes sources de données

On compare les topologies de l'Internet à la granularité AS obtenues par exploitation des sources de données publiques disponibles. Les différents graphes d'AS étudiés sont les suivants :

Whois-ImpExp : Graphes d'AS composés de liens inférés d'après les données extraites des IRR [87]. Les liens inter-AS sont inférés d'après les politiques de routage publiées dans ces IRR. On peut remarquer que certains liens ne sont indiqués que par un seul des deux AS, ou que seul les règles d'import ou d'export pour les routes BGP sont indiquées. On ne conserve un lien que si celui-ci est indiqué par les deux AS et à la fois dans leur politique d'import et d'export. Les autres cas sont considérés peu fiables et sont donc ignorés.

Caida : Graphes d'AS obtenus seulement à partir de tables de routage BGP [91] (les relations économiques sont également indiquées [41]).

Spynet : Graphes d'AS construits à partir de données obtenues par sondage passif et actif du réseau BGP dans le plan de contrôle (bases publiques et looking-glass).

Skitter : Graphes d'AS obtenus à partir de mesures dans le plan de transfert dans le cadre du projet Skitter [92]. Les liens inter-AS proposés sont inférés d'après des liens entre routeurs observés par la commande traceroute. Certains liens sont notés "MOAS" ou "AS Set" car ils correspondent à une information peu fiable. Ces liens sont ignorés.

BGP-Whois : Graphes d'AS obtenus à partir de mesures de tables de routage BGP (projets Route-Views, RIS et routeurs looking glass) et de liens extraits d'IRR [196].

La comparaison de ces topologies s'effectue en étudiant l'évolution de grandeurs caractéristiques au cours du temps (la période d'observation s'étale sur presque trois années). On s'intéresse notamment à l'évolution des grandeurs suivantes :

Nombre d'AS observés dans les différents graphes interdomaine (figure 2.2(a)) .

Nombre de liens AS-AS observés dans les différents graphes interdomaine (figure 2.2(b)) .

On constate que les graphes d'AS construits à partir des informations provenant du plan de contrôle et du plan de gestion ($BGP-WHOIS$) sont les plus grands (en termes d'AS et de liens AS-AS). Toutefois, leur intérêt est à relativiser. En effet, le surplus d'éléments topologiques observés peut être expliqué par le grand nombre de routeurs collecteurs BGP [196] mais aussi par le caractère peu fiable des données des IRR (données périmées et erronées). Les graphes construits à partir d'informations collectées uniquement dans le plan de transfert ($Skitter$) ou de gestion ($WHOIS-ImpExp$) sont ceux qui contiennent quant à eux le moins d'éléments topologiques. Cela résulte entre autres de la suppression des éléments considérés peu fiables et souligne les limitations de ces méthodes. On peut constater que les graphes *Spynet* et $Caida$, obtenus à partir de sondages passifs et actifs, sont les plus complets. Pour notre étude, on utilisera donc des topologies obtenues par exploitation des données provenant du plan de contrôle. Ces topologies offrent l'intérêt d'être à la fois relativement complètes et de présenter les plus faibles incertitudes de mesure.

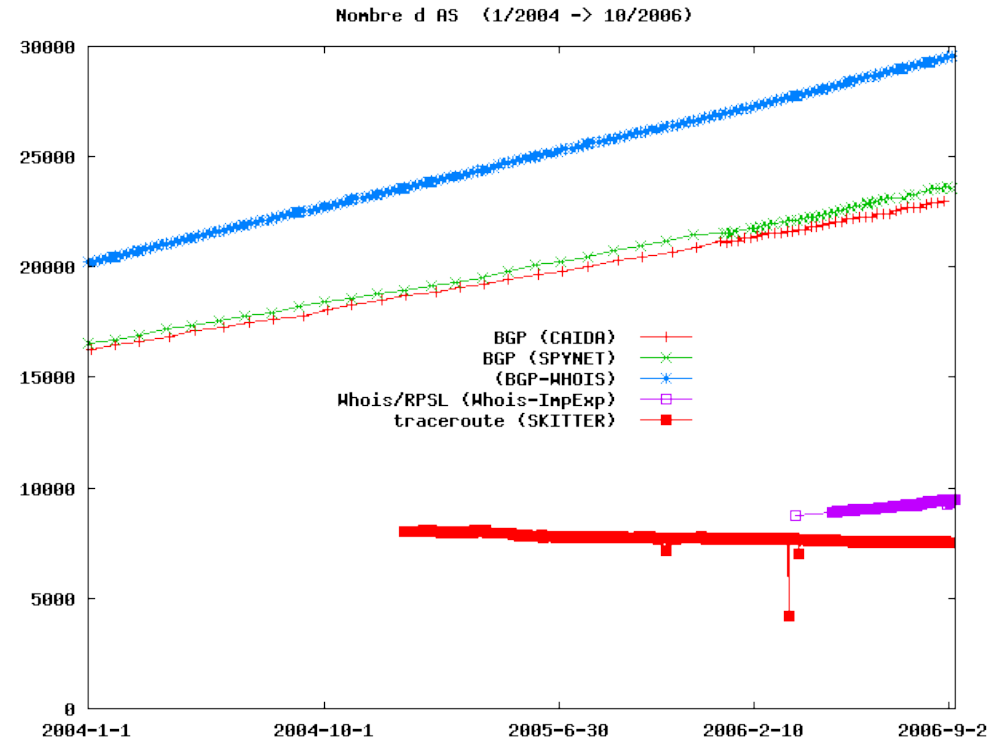
e) Évolution des topologies interdomaines issues des tables de routage BGP

On souhaite maintenant observer la stabilité temporelle des données contenues dans les tables de routage BGP collectées. Pour ce faire, on étudie à nouveau l'évolution du nombre d'AS et de liens AS-AS dans les topologies associées, ainsi que l'évolution dans le temps de deux nouvelles grandeurs :

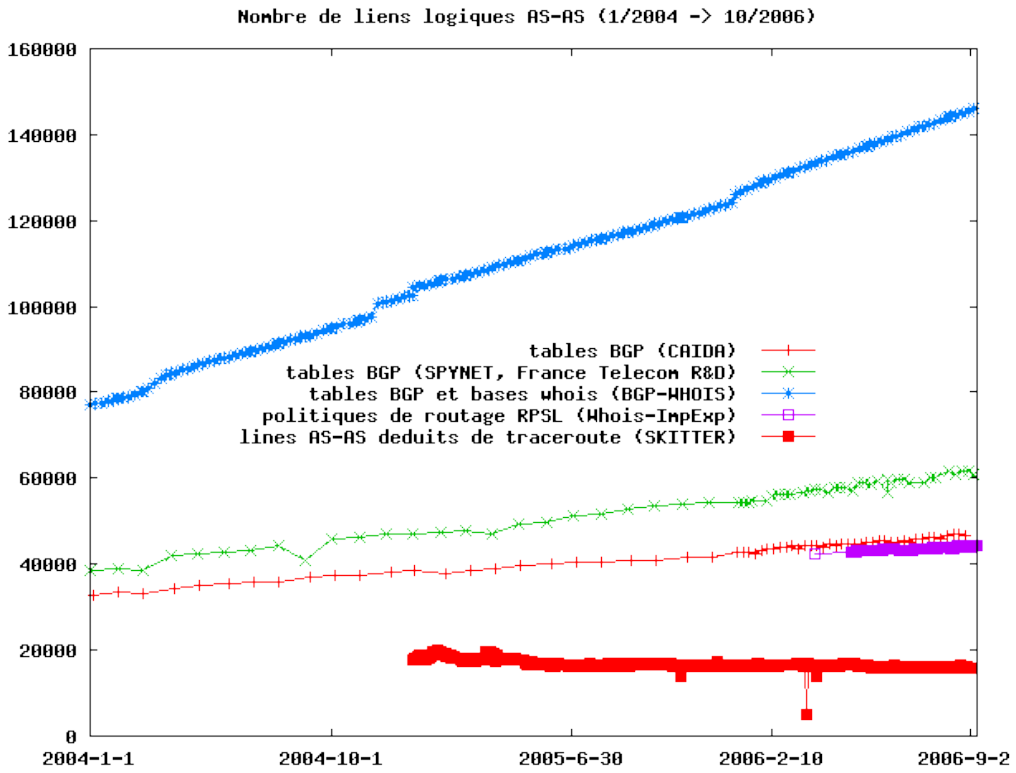
Nombre de triplets d'AS : un triplet d'AS, aussi appelé atome de politique de routage, est formé de trois AS présents consécutivement dans un chemin d'AS. Cet indicateur mesure le nombre d'éléments de politique de routage de transit. Son intérêt sera explicité plus en détails page 49.

Nombre de routes BGP : celui-ci donne le nombre de routes BGP distinctes utilisées pour construire une topologie.

La figure 2.3 montre l'évolution des ces grandeurs sur plus de quatre ans. Ces grandeurs sont mesurées au sein de graphes *Spynet*, construits comme expliqués précédemment.



(a) Evolution du nombre d'AS en fonction du temps

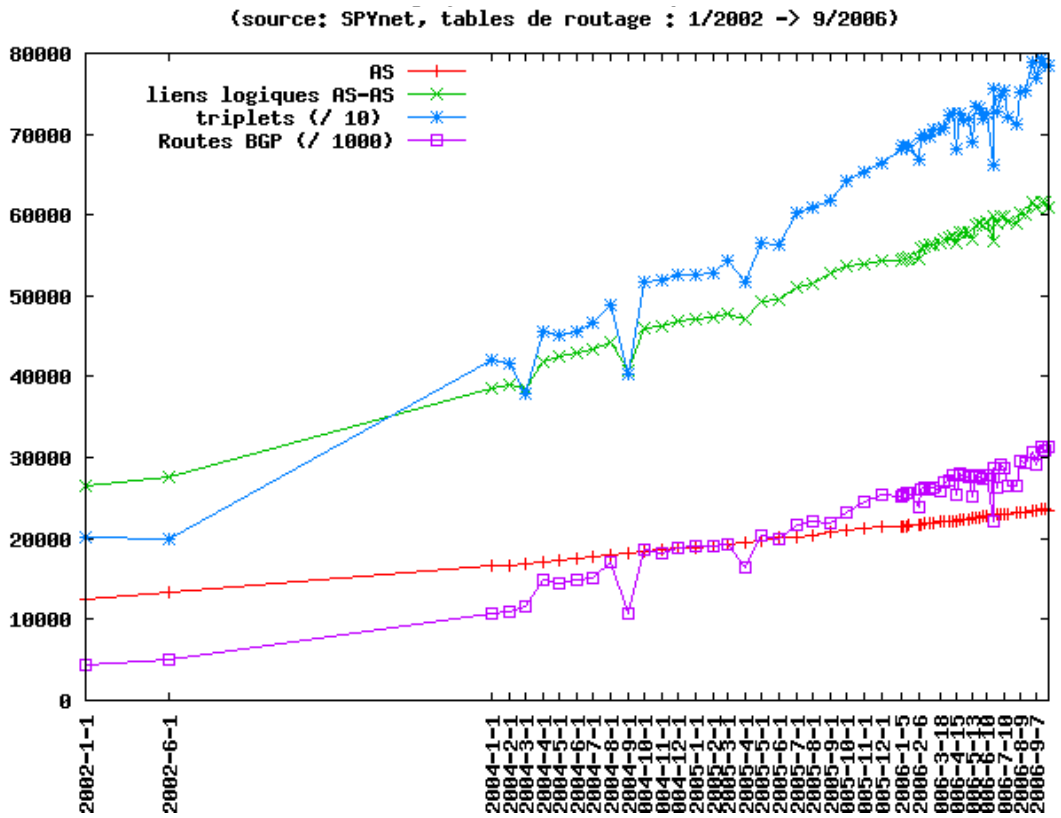


(b) Evolution du nombre de liens inter-AS en fonction du temps

FIG. 2.2 – Évolution des grandeurs caractéristiques des topologies interdomaines construites à partir des diverses sources de données répertoriées

On observe une forte évolution des différents éléments topologiques, bien que nos sondes soient connectées à plus d'AS sources à la fin de la période de mesure (Août 2006) qu'au début (Janvier 2002). Une très grande partie des AS est visible parce qu'ils participent aux routages de plages d'adresses ou parce qu'ils annoncent leurs préfixes. Lorsqu'on utilise des tables de routage en provenance d'autres d'AS-sonde, on observe plus d'arcs mais pas significativement plus d'AS. Les tables de routage BGP complètes des routeurs sondes[8] contiennent au moins une route BGP (la meilleure) pour plus de 200 000 préfixes. A un instant donné, une table de routage contient un nombre non négligeable de routes temporaires vers les préfixes dont la convergence du protocole BGP est en cours. Sur la figure 2.4 on a reporté les différentes distributions CDDF des degrés de toutes les topologies interdomaines de la figure 2.3. La distribution des degrés est très stable et conserve nettement sa forme en une loi de puissance. En résumé, même si sur de longues périodes la taille des topologie évolue beaucoup (nombre d'AS et nombre de liens), et

[8]Ou des routeurs voisins des routeurs sondes.



(a) Evolution des nombres d'AS, de liens AS-AS, de triplets d'AS et de routes BGP. La légende indique le cas échéant le facteur d'échelle qui a été utilisé pour chaque indicateur.

FIG. 2.3 – Évolution des graphes interdomaine mesurés par sondage BGP passif

même si l'activité des échanges de messages BGP est soutenue, la distribution des degrés des AS du graphe conserve sa forme générale au cours du temps. On peut donc conclure que les instabilités de routes présentes dans les tables de routage BGP n'influencent pas de manière significative la distribution des degrés de la topologie interdomaine mesurée. On s'intéresse toutefois aux topologies interdomaine qui correspondent à des routages stables. En toute rigueur, une topologie issue de différentes tables de routage ne répond pas à une telle recherche, car une partie de chaque table de routage relate des routes temporaires. Pour identifier les informations de routages stables ce celles qu'ils ne le sont pas, il faut étudier l'évolution de la dynamique de la topologie et des routes BGP. Le problème majeur est la taille des données. Lorsqu'on répertorie des données dans le temps (des topologies d'AS ou des chemins d'AS), la taille des structures de données devient vite très grande. Il est nécessaire d'utiliser une méthode performante en temps de calcul qui n'utilise qu'un espace mémoire restreint mais qui permet d'étudier la dynamique du réseau BGP dans son intégralité. Pour cette thèse, nous avons créé nos propres outils téléchargeant périodiquement les tables de routages BGP publiques.

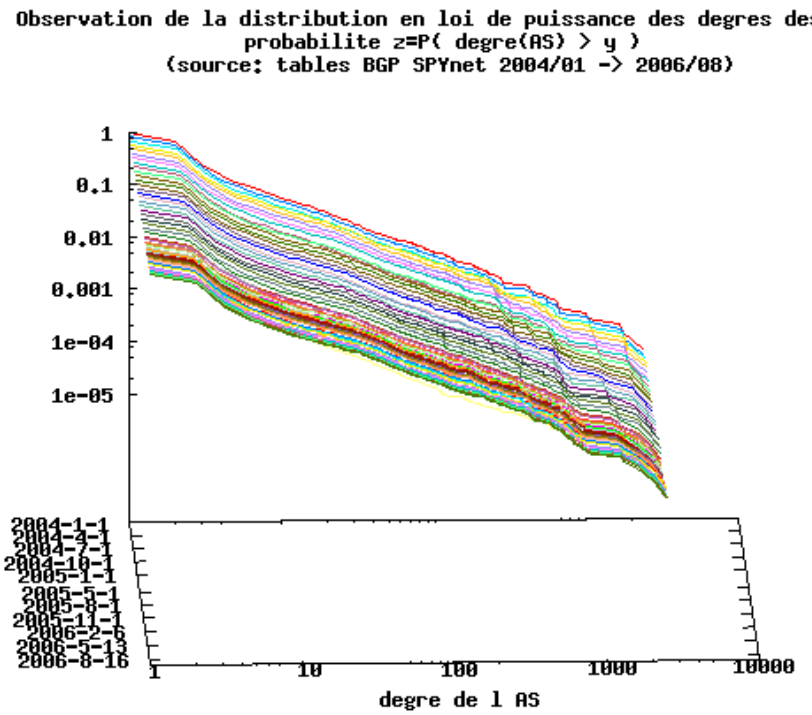


FIG. 2.4 – Loi de distribution des degrés des AS pour les topologies interdomaines dans l'application SpyNET

2.1.2 Mesure des politiques de routage avec une tomographie BGP

La tomographie peut se définir comme un procédé d'exploitation des entrées et des caractéristiques d'un système à partir d'observations partielles faites sur les sorties et les observables de ce système. A partir de la connaissance de ces sorties et du mode de fonctionnement de ce système, il devient possible de reconstituer les différentes entrées. On utilise ici le terme tomographie non pas dans son sens usuel mais pour désigner un ensemble de chemins BGP dans l'Internet.

a) Données d'entrée

Une table de routage BGP peut être archivées sous la forme d'un fichier avec les deux formats suivants :

Format MRT : les attributs BGP sont disponibles pour chaque route contenue dans le fichier (next hop, *AS_PATH*, Local Preference,, *community*, *MED*, *origin*, poids IGP, attributs iBGP...). L'accès aux tables de routage sous format MRT [2] n'est possible que si l'on peut accéder directement aux routeurs.

Trace de la commande "show ip bgp" : la Loc-RIB du routeur est sous la forme d'un fichier texte constitué d'une route BGP par ligne. Les routes sont formées des attributs NLRI (adresse du préfixe et longueur du masque), next hop, *AS_PATH*, *origin* et quelquefois les attributs *Local Pref* et *Weight* (de Cisco) comme le montre la figure 2.1.

Hormis l'*AS_PATH*, les valeurs des différents attributs BGP (*Local Pref* par exemple) découlent d'une vision locale de l'AS pour le routeur qui fournit sa table de routage. Le nombre limité de sondes disponibles ne fournit donc qu'une vue partielle et biaisée des configurations locales de chaque AS. L'information la plus pertinente pour notre étude est l'attribut *AS_PATH*. Celui-ci représente le chemin d'AS qu'il faut traverser pour atteindre le préfixe. De ces attributs *AS_PATH*, on extrait les séquences d'AS uniquement[9]. L'ensemble des chemins d'AS obtenu par ce procédé fournit des routes BGP pour chaque routeur collecteur.

[9]Un attribut *AS_PATH* est une séquence de segments. Les segments sont des listes ordonnées d'AS ou d'AS-Set, un AS-Set étant un groupe d'AS potentiellement traversés sans garantie et dans un ordre quelconque. Tous les cas d'AS-Set sont donc rejetés car on ne connaît ni l'ordre de parcours des AS ni le nombre d'AS traversés. Par exemple, l'AS PATH [2] [3] [22] est accepté avec le chemin 2-3-22, mais l'AS Path [1] [4] [55,3006,788] est rejeté en raison de l'AS-Set [55,3006,788].

Définition 1. *Route canonique et motif de répétition d'un chemin d'AS*

La route canonique *d'un chemin d'AS est la séquence d'AS distincts sans répétition déduite (voir exemple ci-dessous).*

Le motif de répétition *d'un chemin d'AS est un tableau d'entiers indiquant le nombre de fois qu'un AS est répété dans le chemin. Autrement dit, si le i-ème AS de la route canonique d'un chemin est répété n fois dans le chemin d'origine, alors le i-ème élément du motif de répétition porte la valeur n.*

Exemple : Le chemin d'AS 2 1 1 1 500 500 300 a pour route canonique 2 1 500 300 et pour motif de répétition 1 3 2 1.

Définition 2. *Route BGP simplifiée*

*Une **route BGP** simplifiée est identifiée par les données suivantes :*

- *une route canonique,*
- *un motif de répétition,*
- *un préfixe destination,*
- *un nexthop,*
- *une date de mesure.*

Exemple : l'*AS_PATH* 5511 2200 2200 observé le 10/01/2006 pour le préfixe 193.49.0.0/23 obtenu par la sonde rrc04 du projet RIS, avec le next hop BGP 193.251.245.128 est la route : (5511 2200, 1 2, 193.49.0.0/23, 193.251.245.128, $10-01-2006$)
On appellera par la suite "route", une route BGP simplifiée.

Remarque : chaque routeur sonde est connecté à de multiples routeurs (internes à l'AS et externes à l'AS). Les routes BGP extraites d'une sonde d'un projet de routage comme Route-Views [140] ou RIS [135] sont des *AS_PATH* avec l'AS du routeur connecté à la sonde en début de chemin canonique. Par contre dans le cas de routeurs *looking-glass*, on connaît l'AS du routeur absent des routes d'AS fournies. Par exemple, si on extrait des routes en provenance d'un routeur de l'AS 5511, alors aucune route n'a d'attribut *AS_PATH* qui commence par le numéro 5511. Si ce routeur avait été connecté à un routeur de Route-Views, l'AS 5511 serait en début des *AS_PATH* dans les informations fournies par Route-Views. Nous allons donc transformer les routes en provenance de routeurs looking glass et les routes en provenance de routeurs sondes publiques pour avoir des données homogènes.

b) Une tomographie BGP

Un ensemble de routes collectées sur plusieurs routeurs sondes et sur plusieurs périodes de temps est appelé tomographie BGP. Plus formellement :

Définition 3. *Tomographie BGP*
Une tomographie BGP P est un ensemble de routes BGP R en provenance de plusieurs sondes BGP.
On dénote note $SONDES$ l'ensemble des sondes (points de collecte). On note $Prefix$ l'ensemble des NLRI. On note next hop l'ensemble des adresses de Next Hop BGP. On note $ROUTES$ l'ensemble des routes canoniques et $REPETS$ l'ensemble des motifs de répétition. On note $DATES$ l'ensemble des dates.

Tomographie *(routes pour chaque next hop de chaque sonde) :*

$$P = \bigcup_{s \in SONDES} \bigcup_{nh \in next\,hop(s)} P_{s,nh}$$

Routes d'un collecteur vers les NLRI :

$$P_{s,nh} = \bigcup_{p \in Prefix(P(s,nh))} P_{s,nh,p}$$

Routes d'AS d'un collecteur vers un NLRI :

$$P_{s,nh,p} = \bigcup_{path \in PATHS(s,nh,p)} \bigcup_{repet \in REPETS(s,nh,p,path)} P_{s,nh,p,path,repet}$$

Apparitions temporelles d' une même route d'un collecteur vers un NLRI :

$$P_{s,nh,p,path,repet} = \bigcup_{date \in DATES(s,nh,p,path,repet)} R$$

On dira qu'une tomographie est élémentaire, lorsqu'elle ne contient des données que pour une seule date. On appelle début d'une tomographie (respectivement fin d'une tomographie) la date d'apparition de la route la plus ancienne (resp. la plus récente) de la tomographie P. On appelle durée d'une tomographie la durée totale d'observation des routes.

Il est possible de diminuer le niveau de détail des dates de collecte des routes. En effet, si on ne conserve qu'une table de routage par jour pour chaque sonde, on ne mémorisera ni l'heure, ni les minutes, ni les secondes. La granularité temporelle d'une tomographie peut être adaptée de façon à conserver plus ou moins de détails.

Définition 4. *Tomographie BGP passive et active*

Une tomographie passive est formée de routes extraites de tables de routage. Typiquement, on utilisera les tables de routage successives en provenance des mêmes sondes, à intervalles réguliers.

Une tomographie active est formée de routes extraites de messages d'update BGP. Typiquement, on utilisera tous les messages BGP collectés sur toutes les sondes, pendant une période donnée.

Une tomographie mixte est formée de routes extraites de messages d'update BGP et de tables de routage BGP.

Une tomographie passive se compose de tables de routage BGP extraites à des moments précis, notamment pendant la convergence du protocole. Une tomographie active se compose des messages de mises à jour en provenance de plusieurs routeurs, montrant la dynamique du protocole (qui dépend du point d'observation), et inclut le redémarrage des sessions BGP et les messages BGP qui suppriment des routes (BGP Withdrawal).

Définition 5. *AS source d'une route*
L'AS source d'une route est le premier AS du chemin canonique de la route. On supposera que le next hop d'une route appartient à l'AS source de la route.

Exemple : La route canonique 2200 5511 3215 pour le préfixe 93.0.0.0/8 indique que l'AS 2200 est AS source pour cette route.
Une tomographie se place dans un cas idéal, où un routeur est connecté avec des sessions eBGP à de multiples routeurs dont l'adresse de next hop BGP identifie le premier AS de chaque route fournie par le routeur. Pour des raisons pratiques[10], on transforme les routes pour qu'un next hop d'une tomographie soit la représentation des 3 éléments : AS source, sonde, adresse de next hop.

Définition 6. *AS origine d'un préfixe*
On appelle AS origine d'un préfixe p pour une route canonique R le dernier AS de la route R. On note cet AS $ORIG(R)$.

Exemple : La route canonique 2200 5511 3215 pour le préfixe 93.0.0.0/8 indique que l'AS 3215 est AS origine du préfixe.

[10]On peut observer le même next hop BGP pour différents AS sources.

L'AS origine d'un préfixe est le premier AS qui annonce ce préfixe. Cette première annonce est ensuite propagée dans le réseau inter-AS. En règle générale, chaque préfixe est annoncé par un unique AS.

c) Les observables des politiques de routage

Dans une tomographie BGP, on mesure des routes d'AS en direction de préfixes. Autrement dit, on dispose d'une ou plusieurs routes d'AS pour un ou plusieurs routeurs d'un AS source vers chacun des préfixes observé par ces routeurs.

Définition 7. *Routes d'AS*
Les routes d'AS sans répétition d'une tomographie sont les routes canoniques PATHS.
Les différents chemins sans répétition du next hop nh vers le préfixe p sont notés PATHS(nh,p).

$$PATHS(nh,p) = \left\{ \begin{array}{c} path \in PATHS/ \\ \exists\, d, motif \quad avec\ R = (nh, p, path, motif, d) \in P \end{array} \right\}$$

L'ensemble des routes d'AS d'une tomographie (avec répétition) est noté $ASPATHS$.

$$ASPATHS(nh,p) = \left\{ \begin{array}{l} (path, motif) \in PATHS \times REPETS/ \\ \exists d \ \ avec\ R = (nh, p, path, motif, d) \in P \end{array} \right\}$$

Une route avec répétition est un couple composé d'une route canonique et d'un motif de répétition.

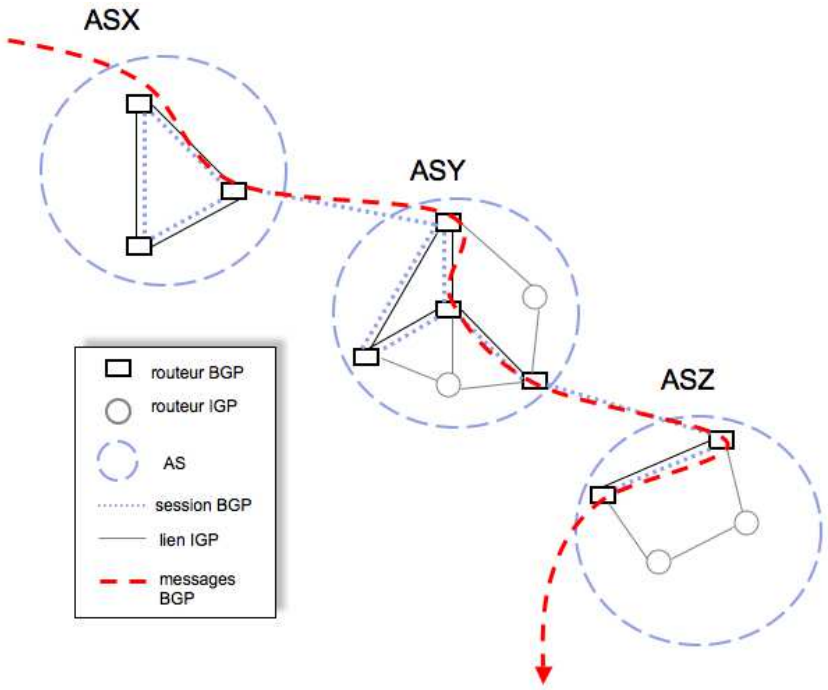
Les routes observées avec différents préfixes permettent de déterminer les éléments topologiques suivants :
– les différents préfixes dont un AS est origine,
– la topologie du graphe logique des AS.
Afin de caractériser plus précisément les routages, on introduit des éléments topologiques supplémentaires que nous appelons atomes de politique de routage.

Définition 8. *Atomes de politiques de routage*
*Un **atome de politique de routage « transit-préfixe »** noté (p, X, Y, Z, r) correspond à la transmission d'un message BGP pour un préfixe p. Le message BGP est propagé par l'AS Y, de l'AS X vers l'AS Z. Le poids r correspond au nombre de répétition de l'AS Y dans l'AS_PATH du message propagé à Z en provenance de X.*
Un atome de politique de routage d'annonce d'un AS X, correspond à la transmission d'un message BGP concernant un NLRI p dont AS X est origine. Le message BGP est

propagé vers AS Y, un voisin de AS X. A chaque atome de politique de routage d'annonce, on associe un poids entier qui correspond au nombre de fois que l'AS X se répète dans les AS PATH des messages BGP propagés à Y pour le NLRI p. On notera un ***atome d'annonce*** *par* (p, X, Y, r)*, où* r *correspond au poids de répétition.*

Exemple : considérons la route canonique 7911 3356 5511 3215 observée pour le préfixe $p = 83.202.71.0/24$ avec le motif de répétition 1 2 3 2. Alors, on peut en déduire les deux atomes de politique de transit-préfixe $(p, 3215, 5511, 3356, 3)$ et $(p, 5511, 3356, 7911, 2)$, et l'atome d'annonce $(p, 3215, 5511, 2)$.

Définition 9. *Triplets d'AS et atomes de politique de transit*
Un ***atome de politique de routage de transit*** *noté* (X, Y, Z, r) *correspond à la transmission d'un message BGP pour un préfixe quelconque. Le message BGP est propagé par l'AS Y, de l'AS X vers l'AS Z. Le poids r correspond au nombre de répétition de l'AS Y dans l'AS_PATH du message propagé à Z en provenance de X.*
Un atome de transit est aussi appelé triplet d'AS (voir figure 2.5).



(a) Un atome de politique de transit correspond à un chemin d'AS de taille 3 déduit d'un attribut AS Path d'un message BGP.

FIG. 2.5 – Exemple d'un atome de transit (X, Y, Z) avec un poids quelconque.

Les atomes de politique de routage de transit-préfixe représentent les décisions de routage par préfixe d'un AS transit entre deux AS voisins. Les triplets d'AS permettent de sim-

plifier les décisions de transit de chaque AS sans tenir compte des préfixes. Les décisions de transit de messages BGP entre deux AS dépendent, dans la pratique, de nombreux attributs BGP qui sont inconnus de notre point de collecte. Les atomes de politique de transit reflètent les règles agrégées de la politique de routage de l'AS en milieu du triplet (l'AS Y). Malheureusement, même si la volumétrie des informations est grande (on peut récupérer plus de 25 000 000 de routes publiques par jour), le nombre de préfixes observés sur chaque lien inter-AS est inégalement réparti. Sur la figure 2.11 on peut voir que seulement 1% des liens sont observés avec plus de 5% de l'espace d'adresse. On constate que le nombre de préfixes différents observés pour chaque triplet d'AS est très limité. Sur la figure 2.11, seulement $0,1\%$ des triplets sont observés avec plus de 5% de l'espace d'adresse. Plus de 99% des triplets d'AS sont observés avec moins de $0,1\%$ de l'espace d'adresse. Ceci est principalement dû au faible nombre de routes disponibles par rapport à la vision possible de tous les couples origine destination et la nature hiérarchie du routage BGP[11]. Par la suite, les triplets d'AS sont analysés sans tenir compte des préfixes, mais la vision de préfixes sur les liens et les atomes de politique de routage sera quand même utilisée pour la classification hiérarchique des AS dans 2.3.3. On définit les matrices de politique de routage comme la réunion des atomes de politique de routage observés pour le même AS au milieu des atomes.

Définition 10. *Matrices de politiques routage*
Une matrice de politique de transit $M_{transit}(Y)$ d'un AS Y, correspond à la donnée de tous les atomes de politique de transit où Y est l'AS au milieu du triplet d'AS. Les différents voisins de l'AS Y sont indicés de 1 à n. La matrice $M_{transit}(Y)$ est telle que :

$$M_{transit}(Y) = \begin{array}{c} \text{'Z'} \\ \left(\begin{array}{ccc} \dots & \dots & \dots \\ \dots & r & \dots \\ \dots & \dots & \dots \end{array}\right) \end{array} \leftarrow \text{'X'}$$

$$\text{Pour tout } X,Y,Z \in AS, \begin{cases} \text{si } \nexists (X,Y,Z,r) \text{ alors } (M_{transit}(Y))_{X,Z} = 0 \\ \text{sinon } (M_{transit}(Y))_{X,Z} = \underbrace{Min\{r \,/\, \exists (X,Y,Z,r)\}}_{\geq 1} \end{cases}$$

Une matrice de politique d'annonce $M_{annonce}(X)$ d'un AS X, correspond à la donnée de

[11] Un préfixe qui n'appartient pas aux clients ou aux clients des peer d'un AS ne pourra être joint que via un AS fournisseur de plus haut niveau qui lui même utilisera peut-être un AS tiers-1. Les AS de plus haut niveau sont donc statistiquement plus visibles dans les chemins de routage que les AS en périphérie du routage. On observe donc moins de préfixes sur les liens et sur les atomes de politique de routage pour des AS de petite taille que pour les AS dans les plus hauts niveaux de la hiérarchie du routage.

tous les atomes de politique d'annonce où X est l'AS origine du préfixe. Les différents voisins de l'AS X et les préfixes de l'AS X sont indicés de 1 *à* n *et de* 1 *à* n_p*. La matrice* $M_{annonce}(X)$ *est telle que :*

$$M_{annonce}(X) = \begin{matrix} \\ \\ \\ \end{matrix} \overset{\text{'}Y\text{'}}{\begin{pmatrix} \dots & \dots & \dots \\ \dots & r & \dots \\ \dots & \dots & \dots \end{pmatrix}} \begin{matrix} \\ \\ \leftarrow \text{'}p\text{'} \end{matrix}$$

$$\text{Pour tout } X, Y \in AS,\ p \in NLRI,\ \begin{cases} \text{si } \nexists (p, X, Y, r) \text{ alors } \left(M_{annonce}(X)\right)_{p,Y} = 0 \\ \text{sinon } \left(M_{annonce}(X)\right)_{p,Y} = \underbrace{Min\left\{r / \exists (X, Y, r)\right\}}_{\geq 1} \end{cases}$$

Remarque : la matrice de politique de transit d'un AS stub est nulle. S'il n'existe pas de préfixe dont un AS est origine, la matrice de politique d'annonce de l'AS est nulle.

La matrice de politique de transit d'un AS Y que l'on peut déduire d'une tomographie indique les transits de l'AS Y entre ses AS voisins. La matrice de politique d'annonce d'un AS X, extraite d'une tomographie, indique les NLRI annoncés par l'AS X pour chaque AS voisin. Les atomes de politique d'annonce vont être utiles pour connaître les éléments de routages stables dans le temps pour une tomographie. Les atomes de politique de transit définissent les contraintes dans le problème d'inférence des accords d'interconnexion entre AS.

2.1.3 Construction des tomographies BGP stables

Pour obtenir des tomographies BGP stables on utilise un ensemble de filtres qui analysent les préfixes, les annonces de préfixes, les liens et les routes. Notre méthode de création d'une tomographie stable s'ajoute aux réflexions menées dans [28,42,121,144,196] concernant la constitution d'une topologie AS de l'Internet. Notre nouvelle approche combine les informations temporelles, structurelles et le filtrage de routage invalides ou temporaires. La méthode proposée exploite des tomographies passives. Chaque tomographie passive est une table de routage BGP $P_{j,s}$ capturées un jour j sur une sonde s. On collecte les tables des mêmes sondes jour par jour pendant 30 jours. Dans l'annexe B, on utilise une première tomographie $T_{2005-01}$ pour présenter une étude complète de l'effet des filtres. Pour la seconde tomographie $T_{2006-08}$, la volumétrie des données est telle qu'il n'a pas

été possible de stocker toutes les données en même temps[12]. Par contre il est possible en deux phases de lecture de pouvoir extraire une tomographie stable. Une première phase de lecture des données BGP ne conserve que les informations nécessaires pour initialiser les filtres. La deuxième phase de lecture des donnée construit la tomographie finale en ne retenant que les routes non filtrées. La méthode permet le passage à l'échelle car les informations d'initialisation des filtres sont des matrices de politique de routage, et les filtres peuvent être appliqués collecteur par collecteur dans la seconde phase. Pour des raisons de place, les détails des premiers filtres sont reportés dans l'annexe B. Voici les filtres détaillés dans l'annexe :

Correction des préfixes et des routes : suppression des préfixes invalides (filtre pp), des AS privés (filtre ap), et des préfixes trop rarement observés durant la période de mesure (filtre pt).

Correction des annonces de préfixe : réattribution d'un unique AS origine pour chaque préfixe (filtres pe,rm,pc,poo) excepté les préfixes observés avec deux AS origines pendant une grande part de la période d'observation (filtre pm).

Correction des AS sources : les AS sources qui ne fournissent pas une part suffisante de leurs routes sont supprimés (filtre ea).

Après application des filtres (ap), (pp),(pt), (pe), (pm), (rm), (pc), (poo), (ea) sur une tomographie P, on obtient un ensemble de routes cohérentes du point de vue des sources et des destinations BGP.

a) Stabilité temporelle des éléments topologiques du graphe logique des AS

La constitution des données pendant une période de 30 jours permet d'observer plus de routes différentes et plus de couples origine-destination que dans le cas d'une seule table de routage par sonde. Certaines de ces routes sont temporaires. Celles-ci sont par exemple utilisées en cas de pannes ou alors apparaissent durant l'exploration des chemins BGP pour un préfixe. Ces routes vérifient des règles de routage spécifiques.
Le terme *densité absolue* d'un objet désigne la fraction du nombre de jours pendant lesquels cet objet est observé dans les données (voir annexe B pour plus de détails). Un

[12]Les méthodes de compression et d'indexation des données dans nos implémentations permettent d'obtenir de bonnes performances, mais l'espace mémoire disponible sur un ordinateur (plus de $3Go$) ne permet pas le chargement de toutes les routes avec les collecteurs et les dates d'observation correspondants.

objet observé dans les données est dit temporaire lorsque sa densité absolue est faible. Plusieurs éléments topologiques peuvent être observés comme temporaires :

Les liens : les liens de "backup" sont censés apparaître rarement dans une période de temps assez longue, puisque la probabilité de conservation d'un panne pendant 30 jours est faible. Les routes canoniques d'une tomographie qui contiennent des liens de backup sont donc a priori des routes de backup que l'on doit filtrer.

Les atomes d'annonce : les atomes temporaires d'annonce d'une tomographie sont supposés former des routes canoniques temporaires dans la mesure où les voisins d'un AS origine sont supposés router de façon permanente les NLRI annoncés par l'AS origine pendant la période d'observation.

Les atomes de transit : les atomes de transit temporaires d'une tomographie indiquent que l'on a peu observé un AS transmettre des messages entre deux de ses AS voisins. En filtrant tous les atomes de transit de faible densité absolue (atomes peu observés dans le temps), on perdrait l'intérêt d'utiliser beaucoup de tables de routage. En effet grâce à la dynamique des décisions BGP on peut voir plus d'informations sur les politiques de routage de chaque AS en utilisant les mêmes tables de routage en entrée pour des dates successives.

Les chemins d'AS : les chemins d'AS d'une tomographie (considérés avec ou sans répétition) correspondent à une tomographie agrégée sur les AS sources où on ne distingue plus les sondes et les next-hops pour un même AS. Si on filtrait les chemins d'AS temporaires, alors on pourrait perdre comme précédemment l'intérêt de la mesure dynamique des politiques de routage.

On définit pour la suite, les deux filtres (lt) et (aat) qui permettent la suppression des routes BGP contenant des éléments topologiques instables d'une tomographie.

Définition 11. *Filtre lien-temps (lt)*
On appelle filtre lien-temps ou (lt), le filtre qui supprime les routes d'une tomographie qui contiennent au moins un lien logique AS-AS dont la densité absolue est faible.

Définition 12. *Filtre atome-annonce-temps (aat)*
On appelle filtre atome-annonce-temps ou (aat), le filtre qui supprime les routes d'une tomographie qui contiennent un atome d'annonce dont la densité absolue est faible.

Tomographie	$T_{2005-01}$
Filtres $pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}), pc_{0.4}, poo, ea_{0.85}$	
Seuil filtre lt	$\frac{2}{31} < 0.07 < \frac{3}{31}$
Routes filtrées	44 515
Seuil filtre aat	$\frac{2}{31} < 0.07 < \frac{3}{31}$
Préfixes concernés	7 271
Routes filtrées	57 588
Routes canoniques conservées après $lt_{0.07}, aat_{0.07}(\%)$	$\frac{2\,664\,347}{2\,679\,326} \simeq 99.4\%$

TAB. 2.1 – Filtrage lien-temps et atome-annonce-temps

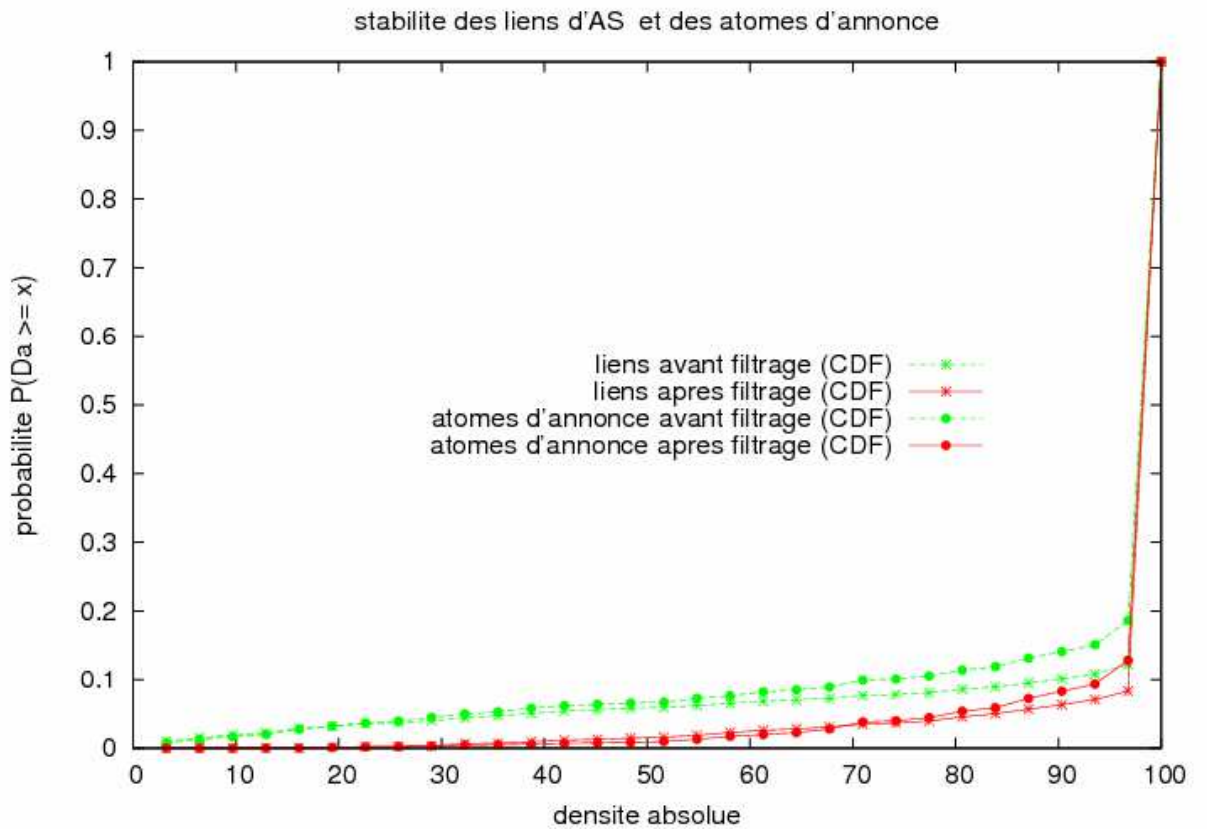


FIG. 2.6 – Filtrage des atomes d'annonce et des liens en fonction de la densité absolue

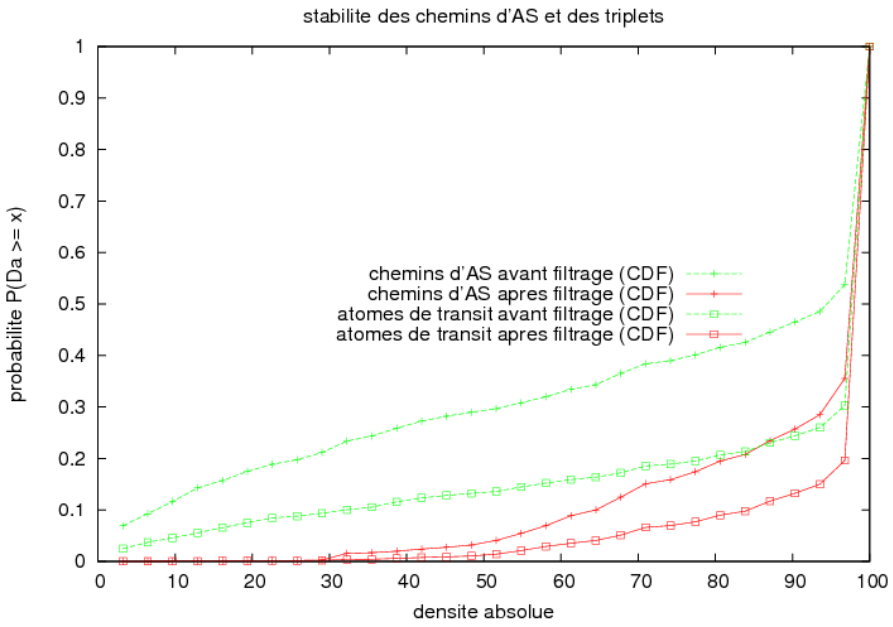


FIG. 2.7 – Impact des filtres (lt) et (aat) sur la densité absolue des chemins et des préfixes

Même avec une valeur de coupure faible (aux alentours de 0.07) les deux filtres (aat) et (lt) permettent d'éliminer des associations (collecteur,chemin,préfixe) et des routes canoniques a priori temporaires[13] (voir figures 2.6 et le tableau 2.1). Une tomographie après l'application des filtres (lt) et (aat) contient beaucoup moins de routes temporaires comme l'indique la densité absolue résultante des atomes de transit et des chemins sur la figure 2.7.

Pour finir, la figure 2.8 indique la densité absolue des liens, des atomes d'annonce, des atomes de transit et des chemins avec ou sans répétition après l'application des filtres. Hormis pour les chemins d'AS, les densités absolue sont très similaires selon la prise en compte des poids de répétition. On peut donc conclure que la diversité des éléments topologiques dans le temps est peu influencée par les pratiques de répétition d'AS.

b) Stabilité des routes canoniques des AS source vers les préfixes

Pour une tomographie donnée P, on s'intéresse aux différents chemins canoniques (chemins d'AS sans répétition) à destination de chaque préfixe. Plus particulièrement, on conserve les routes pour chaque préfixe après avoir appliqué l'ensemble des filtres définis précédemment.

[13] la valeur 0.07 conserve les liens et les atomes d'annonce observés plus de trois jours sur 31.

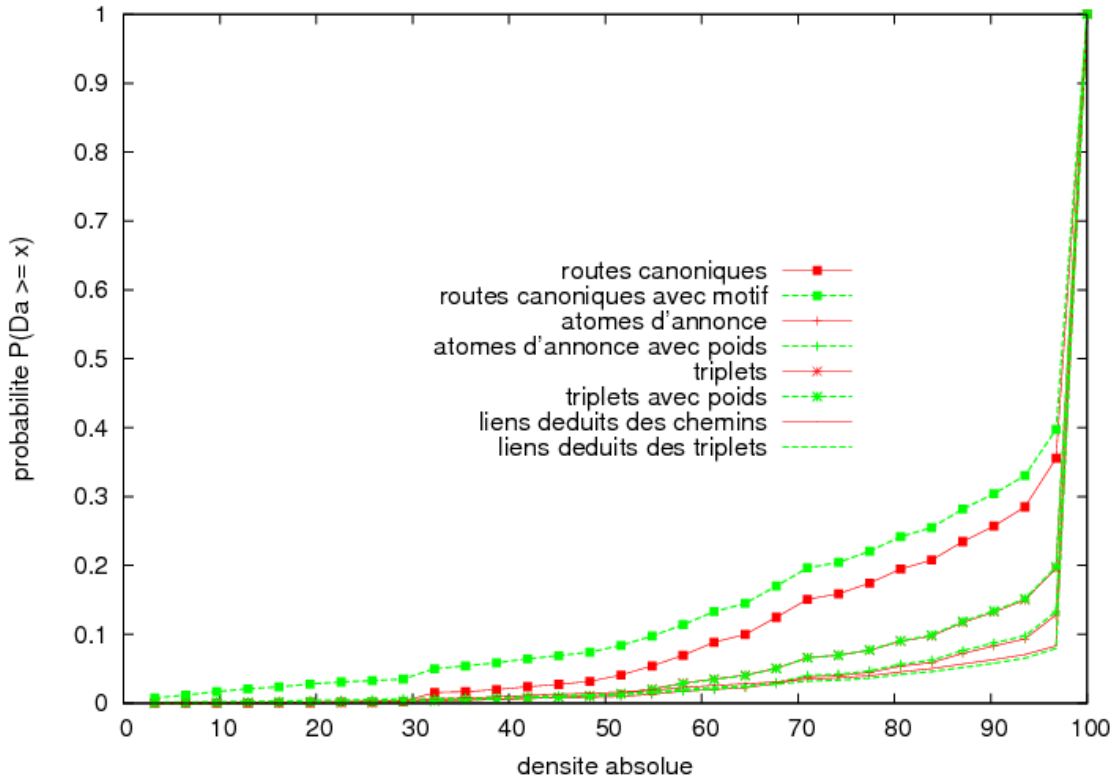


FIG. 2.8 – Répartition des éléments topologiques fonction de la densité absolue

Pour observer les chemins canoniques d'AS, on utilise les deux observables $PATHS$ (routes canoniques) et $ASPATHS$ (chemins avec répétition) introduits dans la définition 7. Soit un couple next hop/préfixe (nh, p) pour lequel il existe au moins une route dans P. Les différents chemins sans répétition (respectivement avec répétition) du next hop nh vers le préfixe p sont notés $PATHS(nh, p)$ (respectivement $ASPATHS(nh, p)$). On a :

Si $|ASPATHS(nh, p)| = 1$: la route vers p pour le routeur n'a pas changé d'AS_PATH.

Sinon si $|PATHS(nh, p)| = 1$: le routeur n'a pas changé de route d'AS vers le préfixe p mais un AS le long du chemin a répété son AS de plusieurs façons différentes pendant la période de la tomographie pour le préfixe p dans les annonces à destination de l'AS le précédent dans le chemin canonique $PATHS(nh, p)$).

Sinon $|PATHS(nb, p)| > 1$: on a $|ASPATHS(nb, p)| > 1$. Le routeur a changé de route d'AS en direction de p.

Une façon plus pratique et moins coûteuse en calcul est d'étudier les cheminements BGP stables de chaque AS source vers chaque préfixe sans distinguer les différents next-hops de chaque AS source. Une tomographie d'AS source à préfixes consiste à regrouper les routes

en provenance d'un même AS source, quel que soit le next hop et que que soit la sonde. On règle tout d'abord les problèmes liés aux changements d'adressage des routeurs next hop pendant la période de la tomographie. Puis on obtient une vision des routes disponibles au sein des routeurs d'un AS source. Si l'AS source est connecté avec plusieurs sessions eBGP à des routeurs collecteurs (de type route-views ou ris), alors les routes disponibles pour l'AS source dépendent des couples de routes distinctes (décidées sur le poids IGP par exemple) exportés pour les mêmes préfixes sur les différentes sessions. Si l'AS source correspond à l'AS d'un routeur looking glass, la table de routage contient des routes alternatives disponibles au sein de l'AS source. On note :

$$PATHS(AS_{source}, p) = \left\{ \begin{array}{c} path \in PATHS / \, path_1 = AS_{source} \\ \exists nh, d, motif \;\; \text{avec } R = (nh, p, path, motif, d) \in P \end{array} \right\}$$

Exemple : pour le préfixe 203.193.185/24, on a observé quatre routes canoniques différentes en partant de l'AS source 5511 :

Chemin	Dates d'apparition	
$PATH(5511, 203.193.185/24)_i$	d_i (total)	avec conflits
AS5511-AS3561-AS15412-AS18101-AS7633	21	21
AS5511-AS4637-AS7633	3	0
AS5511-AS2914-AS4755-AS7633	7	1
AS5511-AS2914-AS15412-AS18101-AS7633	22	22
Période d'observation	31	22

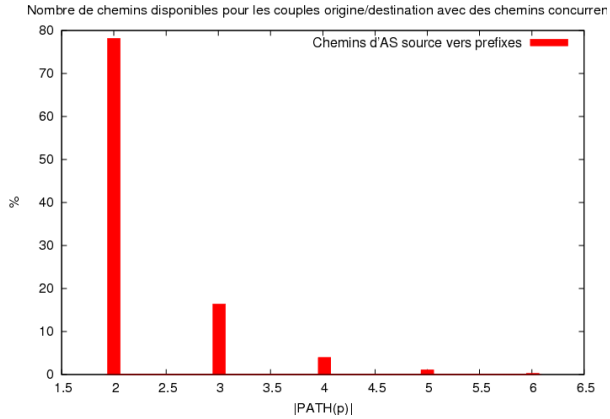
En tenant compte de tous les couples (AS_{source}, préfixe) pour lesquels il existe plusieurs chemins, nous avons reporté la répartition de différents observables en rapport avec la multiplicité des valeurs de $PATHS(AS_{source}, p)$. Dans la figure 2.9(a) on peut voir que pour plus de 80% des couples, il existe seulement deux routes canoniques. Il existe plus de 5 routes canoniques pour un faible pourcentage de couples.

La figure 2.10(a) indique les conflits entre routes concurrentes avec la **densité comparée** D_c (cf. annexe B). Pour l'exemple : $D_c(PATH(5511, 203.193.185/24)) = \frac{22}{31} = 0.709$.

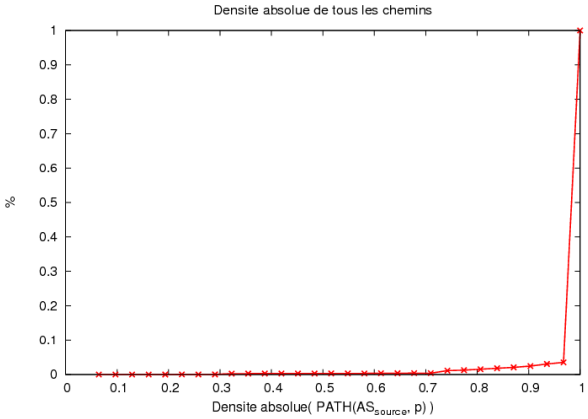
Il apparaît distinctement que 75% des couples correspondent à des changements de routes (pas ou peu de conflits et D_c proche de 0) et presque 20% des couples correspondent à des routes concurrentes (beaucoup de conflits et D_c proche de 1). La figure 2.10(b) indique les juxtapositions ou les chevauchements entre routes concurrentes avec le **coefficient de persistance** U (cf. annexe B). Pour l'exemple :

$$U(PATH(5511, 203.193.185/24)) = \frac{31}{\sum_j d_j} = \frac{31}{53} = 0.584$$

Parmi les différents chemins disponibles pour les mêmes couples (AS_{source},préfixe), on peut remarquer qu'environ 15% de ces couples correspondent à un coefficient de persistance de l'ordre de $\frac{1}{2}$ (lorsque deux chemins sont concurrents pendant toute la période d'observation) et qu'environ 60% correspondent à un coefficient de persistance proche de 1 (changement de chemin avec peu de conflits). Finalement, on remarque figure 2.9(b) que la densité absolue de l'observable $PATHS(AS_{source}, p)$ est proche de 1. On peut très souvent garantir l'observation d'un chemin pendant toute la période si un chemin existe

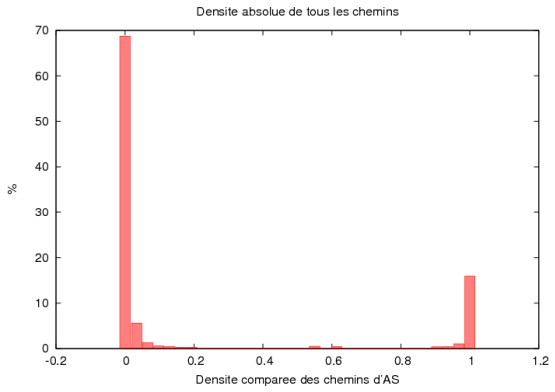


(a) Répartition des chemins multiples pour $T_{2006-08}$

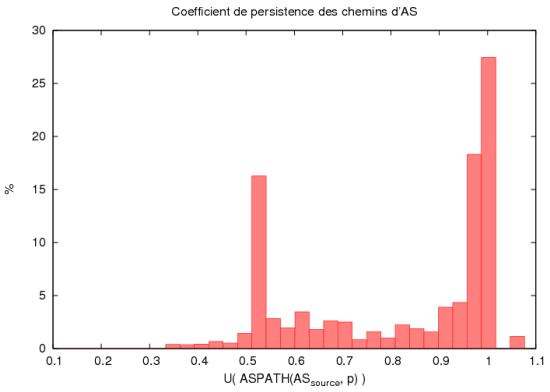


(b) Répartition des valeurs de densité comparée de l'observable $PATHS(AS_{source}, p)$

FIG. 2.9 – Loi de répartition des densité absolues des chemins multiples pour les AS sources vers les préfixes



(a) Répartition des valeurs de densité comparée de l'observable $PATHS(AS_{source}, p)$



(b) Répartition des valeurs de coefficient de persistance de l'observable $PATHS(AS_{source}, p)$

FIG. 2.10 – Répartition des valeurs de densité comparée et de coefficient de persistance pour l'observable $PATHS(AS_{source}, p)$.

Tomographie	$T_{2006-08}$
Filtres $pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}), pc_{0.4}, poo, ea_{0.85}, lt_{0.07}, aat_{0.07}$	
Seuil filtre rs	$\frac{1}{5} = 0.2$
Nombre de routes total	27 006 616
Couples (AS source, préfixe) avec 1 route	$\frac{13\,690\,702}{27\,006\,616} \simeq 50.7\%$
Couples (AS source, préfixe) avec plusieurs routes	$\frac{5\,511\,567}{27\,006\,616} \simeq 20.4\%$
Routes non stables (% total)	$\frac{7\,539\,442}{27\,006\,616} \simeq 27,9\%$
Routes stables (% total)	$\frac{13\,690\,702+5\,776\,472}{27\,006\,616} \simeq 72,1\%$
Routes stables (% multiples)	$\frac{5\,776\,472}{27\,006\,616-13\,690\,702} \simeq 43\%$
Routes canoniques conservées après $rs_{0.2}$(% total)	$\frac{3\,001\,284}{4\,258\,035} \simeq 70.5\%$

TAB. 2.2 – Filtrage route stable

pour une date. On en déduit que pour la majorité des couples (AS_{source},préfixe) il existe une ou plusieurs routes stables pendant la période de la tomographie. Soit un AS source pour lequel on connaît des routes vers un même préfixe. Les routes de densité absolue maximale pour le préfixe sont appelées *routes canoniques dominantes*. Toute route canonique non dominante d'un AS source vers un préfixe est susceptible d'être une route non stable.

Définition 13. *Filtre routes stables (rs)*
Le filtre routes stables ou (rs) supprime pour chaque AS source et chaque préfixe, les routes dont la densité absolue est faible par rapport à la densité absolue des routes canoniques dominantes.

Les routes non stables utilisés par les AS sources et filtrées par $rs_{0.2}$ représentent presque 30% des routes canoniques comme l'indique les résultats du tableau 2.2. Lorsque plusieurs routes existent pour un AS source vers un préfixe, plus d'une route sur deux (57%) n'est pas conservée par le filtre $rs_{0.2}$ car elle est observée au moins 5 fois moins souvent que les routes dominantes conservées.

Dans le début de ce chapitre, on a introduit de nouveaux objets topologiques rendant compte des politiques de routage des AS (les matrices de politique de routage). On a montré comment utiliser un ensemble de tables de routages BGP publiques pour en déduire un jeu idéal de donnée constitué de chemins d'AS vers des préfixes. Ces jeux de données sont utilisés dans la suite du chapitre.

2.2 L'observation des politiques de routage

Il faut bien garder à l'esprit que les chemins utilisés dans une tomographie ne représentent qu'un faible nombre de couples origine-destination (cf. tableau 2.3). Le nombe de points de collecte naturellement incomplet constitue un frein à la connaissance des politiques de routage[14].

Période	206/08/01 - 2006/08/31
Échelle	AS source vers préfixe
Filtres basiques	$basic_{pp,ap,pt_{0.75}}$
Filtres MOAS	$moas_{pe_{0.15},pm_{0.75},\beta_{0.5},U_{1.25},pc_{0.4}}$
Filtres visibilité des AS sources	$ea_{0.85}$
Filtres stabilité	$stable - temps_{lt_{0.07},aat_{0.07}}, rs_{0.2}$
Nombre de préfixes	215 655
Nombre de routes BGP	27 006 616
Nombre de routes canoniques	4 258 035
couples AS source/préfixe destination	$\frac{27\,006\,616}{215655*21000} = 0.6\%$
couples AS source/AS destination	$\frac{4\,258\,035}{21000*21000} \simeq 1\%$

TAB. 2.3 – Nombre de couples origine/destination dans une tomographie

On analyse les limitations de l'observation des politiques de routage avec des tomographies en deux points :

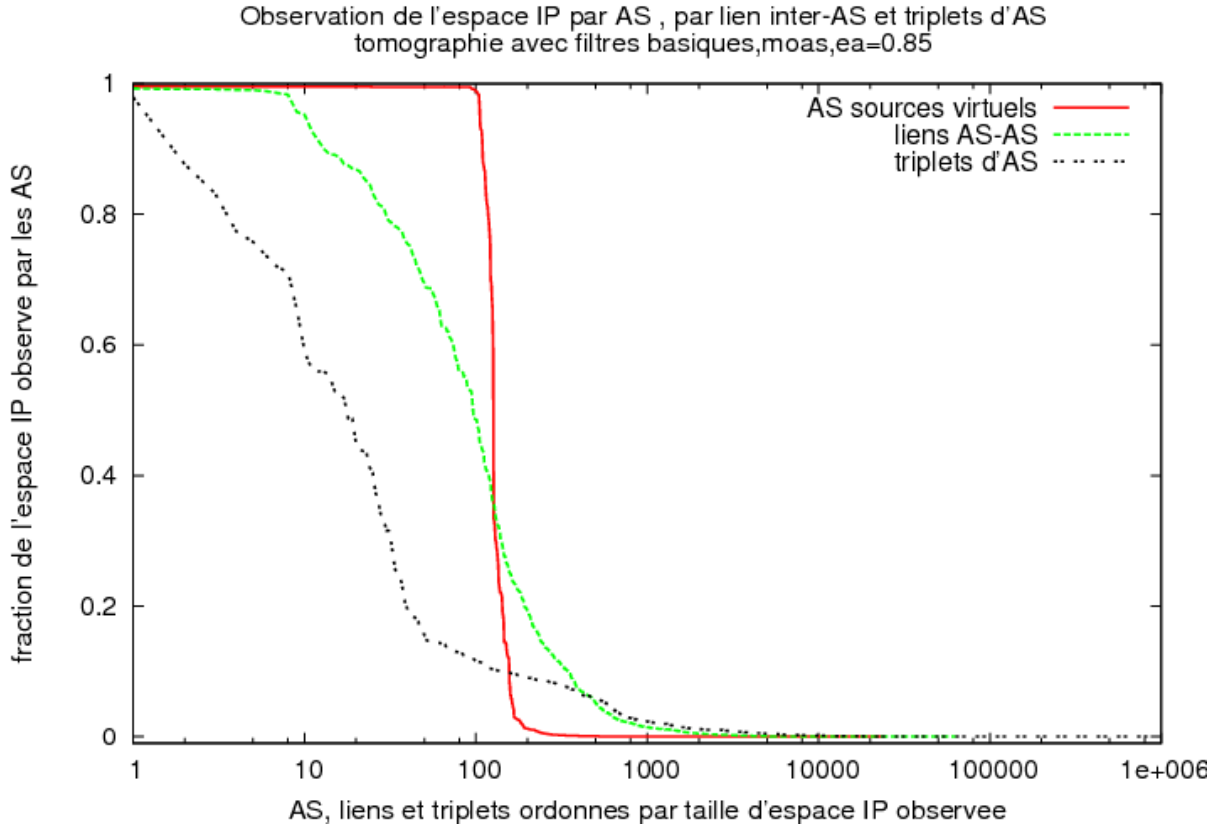
1. Une analyse des différents préfixes mesurés avec les éléments topologiques du graphe interdomaine (cf. 2.2.1).
2. Une analyse des éléments topologiques du graphe interdomaine en fonction du nombre d'AS sources (cf. 2.2.2).

2.2.1 Visibilité des préfixes et des espaces d'adresses

On représente les différentes tailles d'espace d'adresses observées sur les liens et les atomes de politique de routage dans la figure 2.11. Certains liens inter-AS et certains triplets d'AS sont observés avec un espace important d'adresses. On pourrait croire à tort que ces liens ou ces triplets sont proches de la hiérarchie haute des AS. Ces éléments topologiques, en général très proches de certains des AS sources, introduisent un biais de mesure si on comptabilise les données sans tenir compte de la vision simultanée des AS sources. Par exemple, lorsqu'un AS source est de petite taille, et qu'il utilise un AS fournisseur

[14]Le faible nombre d'AS source comparé au nombre total d'AS est à relativiser car il existe moins de 4000 AS non stub.

principal pour une majorité de préfixes, alors le lien est observé avec un espace d'adresse important. L'AS fournisseur quant à lui est observé au milieu d'atomes de politique de routage pour de nombreux préfixes. La prise en compte de la vision des AS sources pour limiter ce biais sera utilisée pour inférer la structure hiérarchique des AS dans 2.3.3.



(a) Quelque éléments topologiques sont observés avec tout l'espace d'adresse. Cela montre le biais que peut introduire le placement des sondes.

FIG. 2.11 – Espace d'adresse observé sur les éléments topologiques (tomographie $T_{2006-08}$)

Pour la suite, on s'intéresse aux liens et aux triplets d'AS indépendamment des préfixes observés.

2.2.2 Éléments topologiques en fonction du nombre d'AS sources

On souhaiterait savoir si la topologie inter-AS mesurée et notamment la distribution des degrés des AS dépend fortement de la méthode de mesure. Le faible nombre d'AS sources possibles (environ 100 sur 20000) peut être une source potentielle d'erreur dans la mesure de certains indicateurs comme le nombre de liens ou les matrices de politique de routage. Si on reprend les résultats concernant la topologie inter-AS mesurée via la lecture des

registres de routage (cf. 2.1.1), on peut voir qu'il est existe a priori significativement plus de liens inter-AS que ceux observés par sondage passif BGP. De même les résultats présentés dans [42] donnent un nombre de liens inter-AS, déduits après décodage des messages de mise à jour BGP, significativement supérieur au nombre de liens observés par un sondage passif. Notons toutefois que les données extraites des registres de routages sont déclaratives et peuvent donc être incomplètes ou obsolètes. Beaucoup de routes d'AS indiquées dans les messages de mise à jour BGP sont également temporaires et instables car elles apparaissent seulement durant la convergence du protocole.

a) Analyse par AS source

Une tomographie BGP exploite un faible nombre de tables de routage d'AS sources différents comparé la taille du réseau, mais :

- les couches hiérarchiques les plus hautes du réseau inter-AS sont souvent empruntées par les routes fournies par chaque AS. Les liens entre AS dans les plus hautes couches hiérarchiques apparaissent très souvent dans les routes fournies par les AS sources.
- si un AS utilise toujours le même lien inter-AS vers un de ses fournisseurs pour annoncer un NLRI, alors ce lien est observé dans les routes des AS sources vers le NLRI. Le principe de connectivité totale permet d'assurer qu'il existe au moins une route vers chaque préfixe. La probabilité d'observer chaque lien inter-AS utilisé par un AS origine pour annoncer chacun de ses NLRI est d'autant plus forte que le nombre de liens utilisés pour ces annonces est faible. Les liens inter-AS vers la périphérie de l'Internet et les liens vers les clients multi-homés sont donc a priori souvent observés.

Les limitations accompagnées du faible nombre d'AS source sont les suivantes :

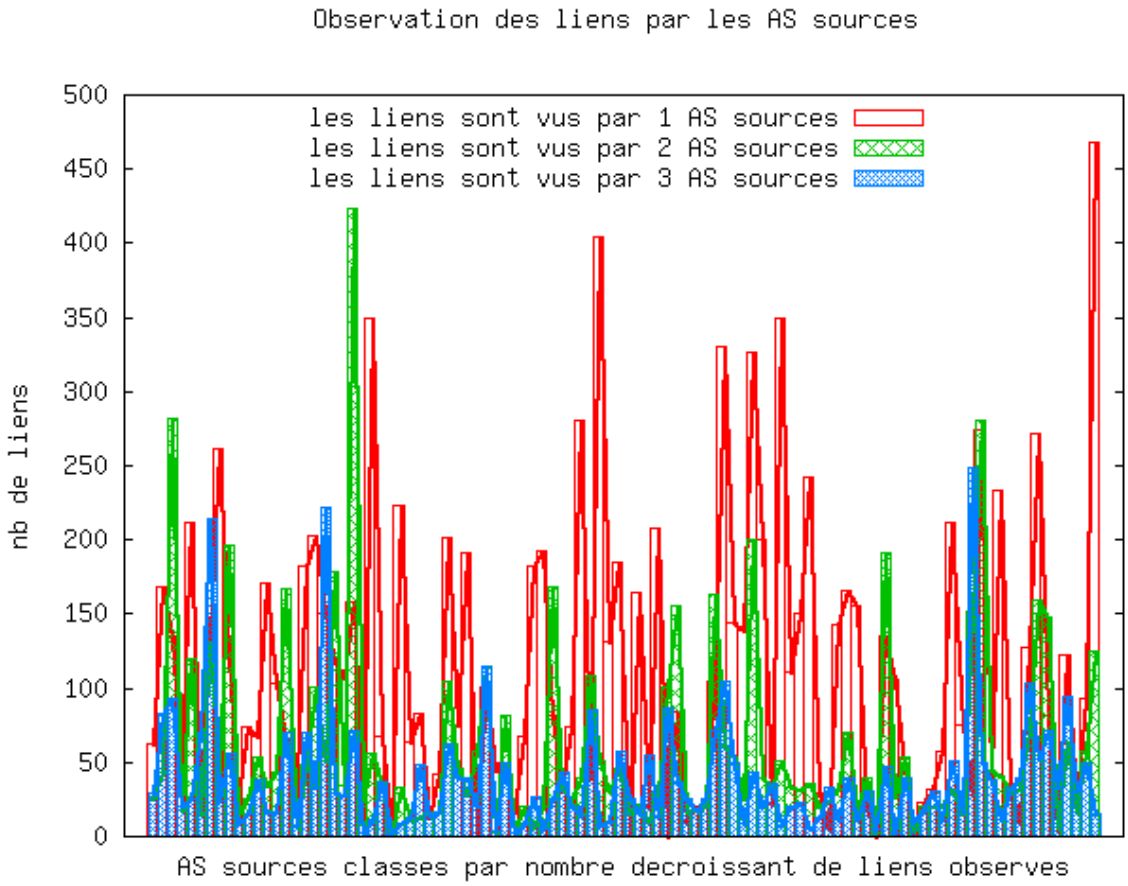
- On ne peut observer une route qui emprunte un lien de peering entre deux AS seulement si l'un de ces deux AS est AS source ou si un AS source est client d'un de ces deux AS.
- L'exportation par un routeur de sa seule meilleure route pour un NLRI limite la diversité des chemins observés[15].

La topologie interdomaine mesurée est donc nécessairement incomplète. La distribution du nombre de liens en fonction du nombre d'AS sources va montrer toutefois qu'une partie majeure de la topologie est "populaire".

[15] En tenant compte du faible diamètre du graphe des AS (d'après la distribution en nombre de bonds figure 2.25) l'effet négatif de l'export de la meilleure route BGP est plus limité qu'il n'apparaît.

b) Popularité des liens inter-AS

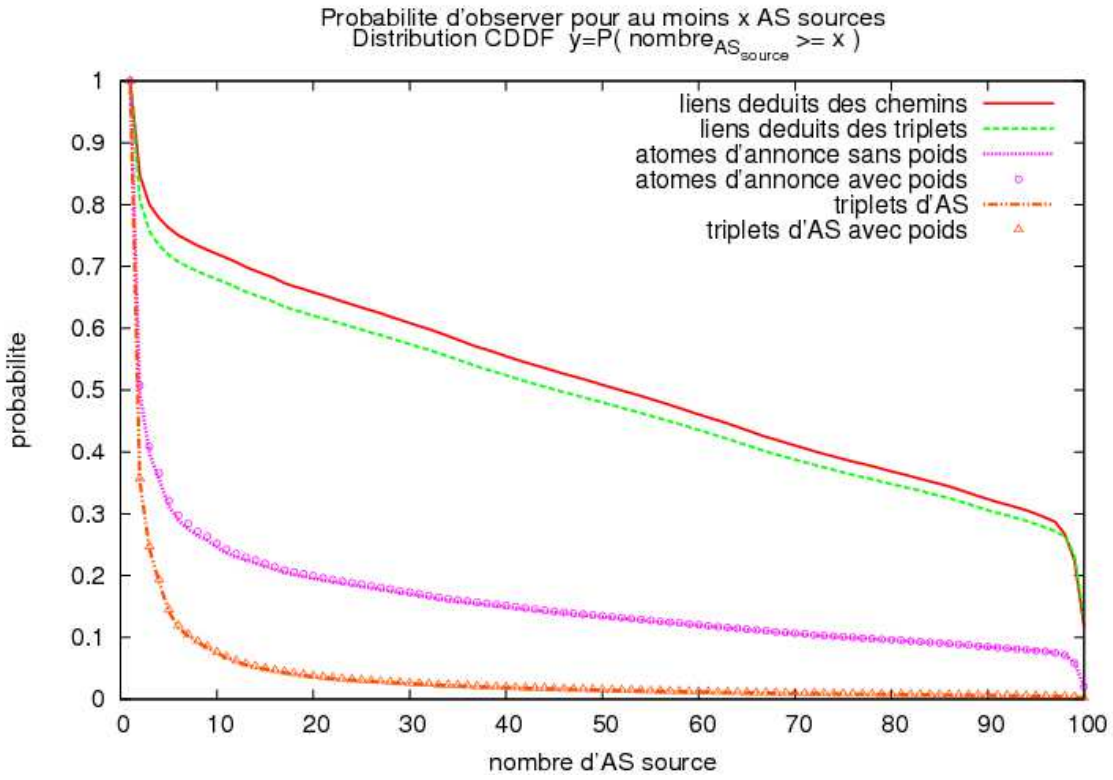
On appelle **popularité d'un lien** le nombre d'AS sources ayant observés ce lien. On appelle liens "rares", les liens observés par peu d'AS sources. On peut supposer que les liens rares sont des liens de peering invisibles des autres AS sources ou alors des liens qui n'apparaissent que dans des chemins BGP alternatifs[16] instables. La figure 2.12 montre que les AS sources qui observent des liens rares sont répartis au hasard. Certains AS voient très peu de liens "rares", alors que d'autres peuvent voir seul jusqu'à 700 liens : soit plus de 1% du total des liens.



(a) tomographie $T_{2006-08}$

FIG. 2.12 – Liens inter-AS observés par un, deux ou trois AS sources seulement

[16]On désigne par chemin BGP alternatif une route d'AS qui apparaît dans les attributs *AS_PATH* des messages BGP indiquant des routes différentes des meilleures routes sélectionnés de façon stable par les routeurs d'un AS.

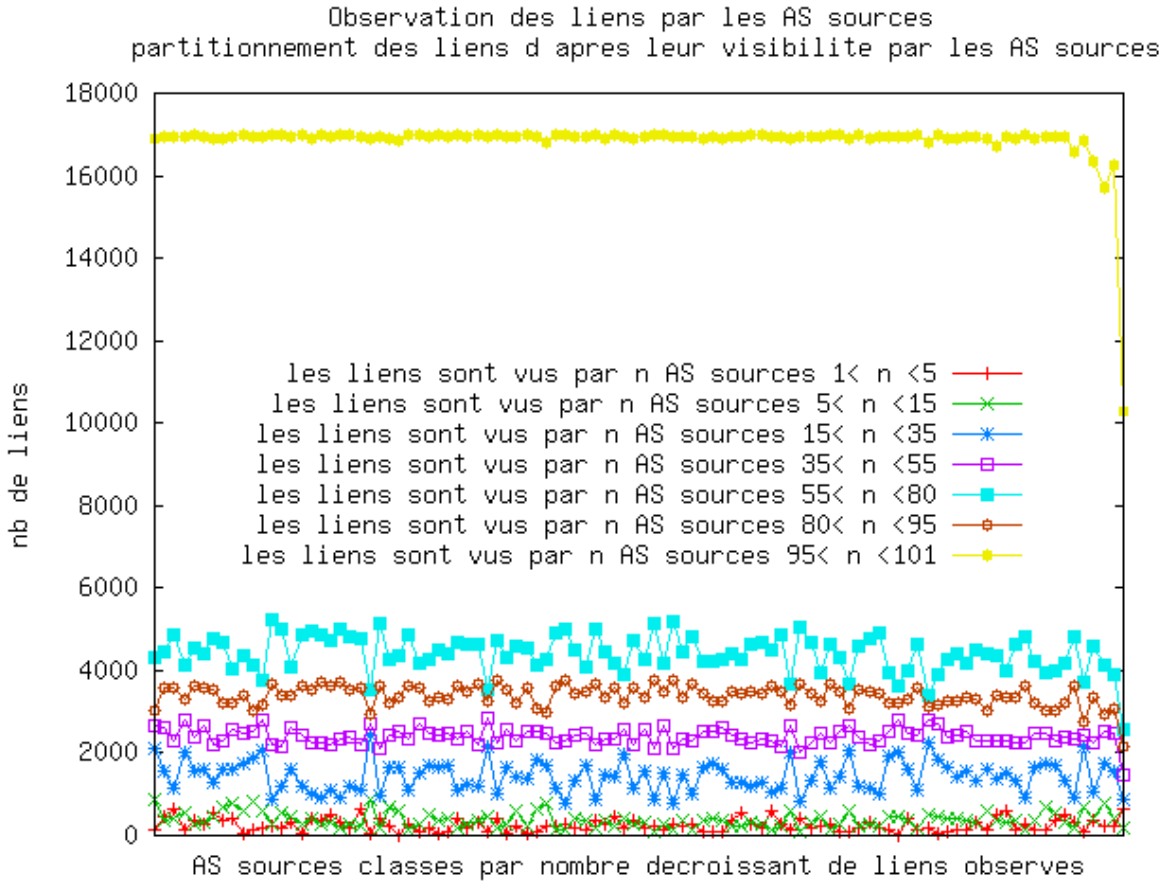


(a) tomographie $T_{2006-08}$

FIG. 2.13 – Probabilité d'observer un élément topologique par un nombre minimal d'AS sources

La figure 2.13 montre la probabilité d'observation d'un lien inter-AS par un nombre minimal d'AS sources. On peut distinguer sur cette courbe 3 types de popularité :

- *Les liens rares (1%≤ ≤3% des AS sources)* : il y a environ 25% de liens rares dans notre topologie dont 15% sont observés uniquement par un seul AS. On considère les liens observés par moins de 3 AS sources au lieu d'un car certains AS sources peuvent être proches les uns des autres et avoir beaucoup de chemins d'AS qui se recouvrent (un AS et un de ses fournisseurs principaux par exemple).
- *Les liens essentiels* (97%≤ ≤100% des AS sources) : il y a environ 25% de liens essentiels dans notre topologie dont un peu moins de 15% sont observés par tous les AS sources. On considère les liens observés par plus de 97% des AS sources et pas 100% car il peut manquer certaines routes en provenance de certains AS sources pour des problèmes de mesure ou bien ces routes ont été supprimées par nos filtres.
- *Les liens quelconques* (3%< <97% des AS sources) : de façon surprenante, dans cette partie des figures, la probabilité d'observer un lien inter-AS ni rare ni essentiel varie linéairement avec le nombre d'AS sources.



(a) tomographie $T_{2006-08}$

FIG. 2.14 – Liens inter-AS observés par un, deux ou trois AS sources seulement

On peut s'intéresser plus précisément aux types de liens observés par chaque AS source. Sur la figure 2.14, on groupe les liens en fonction de leur popularité. On suppose que pour chaque tranche de popularité, le nombre de liens visibles d'un AS source est approximativement constant. Si alors on dispose de deux fois plus d'AS sources et si le coefficient directeur de la partie linéaire des figures varie peu, alors la probabilité qu'un lien soit observé par seulement un seul AS source va tendre vers zéro. En ajoutant un nouvel AS source, on ajouterait en moyenne $\frac{1}{100} * \frac{15}{100} \simeq 0.15\%$ de liens en plus. Le nombre de liens est donc représentatif, car 25% d'AS sources supplémentaires n'ajouterait approximativement que 3% de liens.

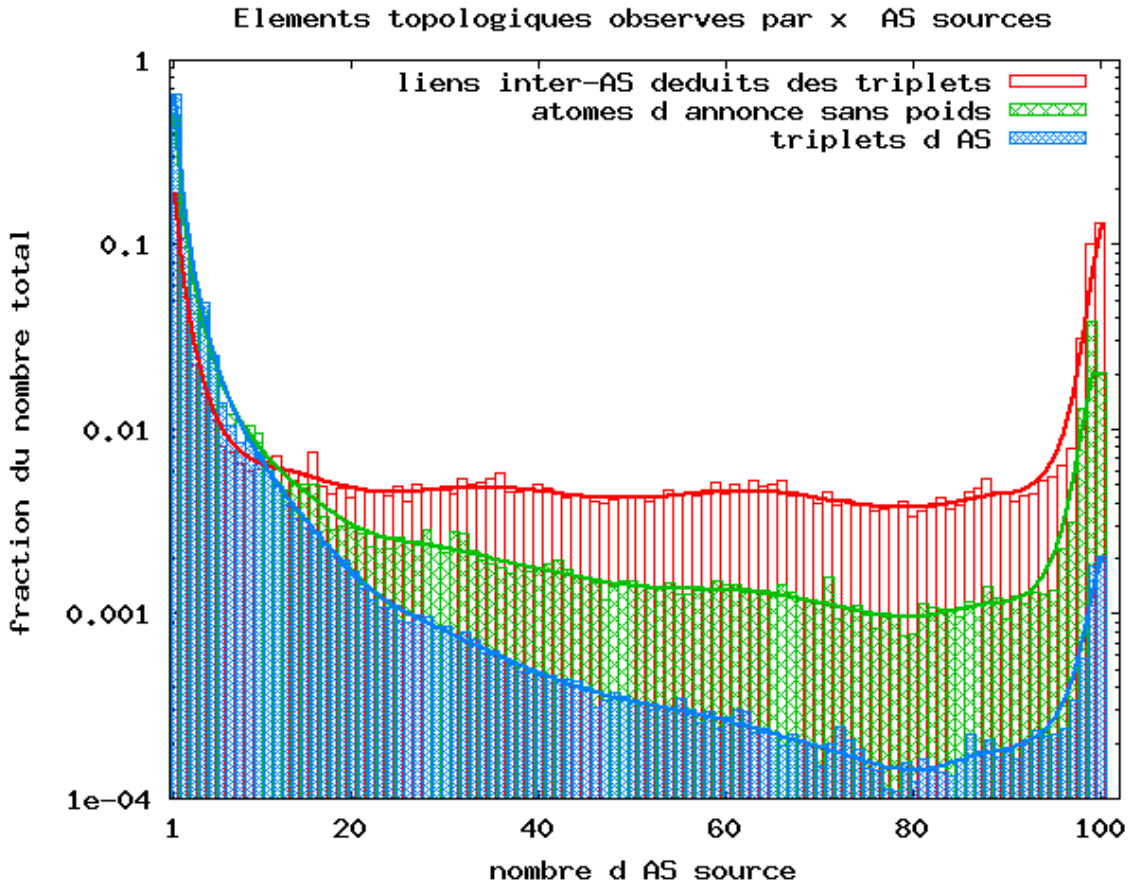
c) Popularité des atomes de politique de routage

Le nombre d'atomes de politique de routage observés en fonction de leur popularité est reporté sur la figure 2.15. Nous utilisons la même terminologie pour la popularités des triplets que pour la popularité des liens. On observe qu'environ 8% des atomes d'annonce sont essentiels (partie droite de la figure 2.15(c)). Ils correspondent aux liens primaires des préfixes concernés. On remarque également que le nombre d'atomes de politique de routage de transit essentiels est très faible (partie droite de la figure 2.15(c)). Cela conforte l'idée que le coeur du routage de l'Internet est de petite taille. En effet, en supposant que le coeur du routage Internet soit principalement constitué d'AS en relation de peering, les atomes de politique de transit essentiels (en d'autres termes les atomes intervenants dans les politiques de routage des AS de plus haut niveau) ne peuvent pas être internes à la couche de plus haut niveau car deux accords de peering ne peuvent pas être utilisés par un même chemin (pour plus détails voir la définition 17). Les atomes de transit essentiels sont de deux types :

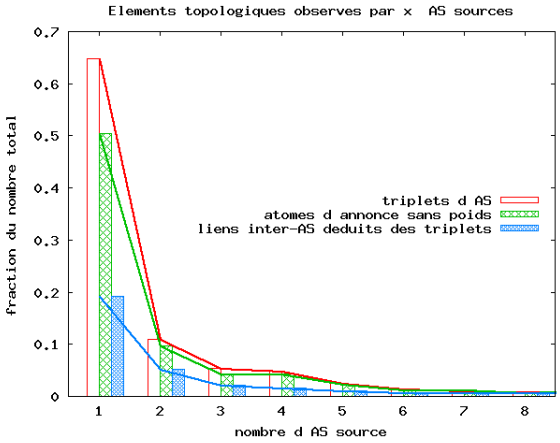
Les atomes de transit essentiels du coeur : Ils sont constitués d'un lien interne à la couche de plus haut niveau (lien inter AS tiers-1) et d'un lien entre la couche de plus haut niveau et une autre couche (AS de transit Tiers-2 ou AS stubs clients finaux).

Les atomes de transit essentiels de périphérie : Ils sont constitués de deux liens successifs dont l'un relie un AS stub à son fournisseur principal et l'autre relie le fournisseur à un de ses fournisseurs principaux. Ces triplets correspondent au cas ou un préfixe d'un AS est majoritairement accédé par les trois mêmes AS terminaux successifs. Ces triplets de périphérie sont d'autant plus présents que les AS stubs annoncent chaque préfixe à un unique AS fournisseur.

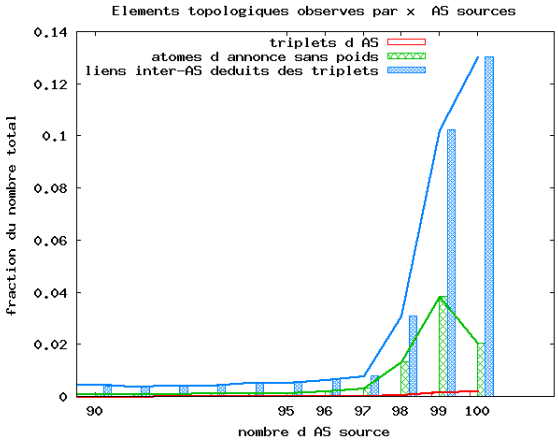
On peut donc classer les différents types de triplets en fonction de la popularité des deux liens (cf. tableau 2.4). En ce qui concerne les atomes rares de politique de routage (partie gauche de la figure 2.15(b)), on remarque que 80% des atomes de transit et 65% des atomes d'annonce sont observés par moins de 4 AS sources. Ceci nous indique que chaque AS source permet la déduction d'un grand nombre de nouveaux atomes de politique de routage alors que le nombre de nouveaux liens est faible. Les matrices de politique de routage fournies sont donc influencées par le nombre d'AS sources.



(a) tomographie $T_{2006-08}$

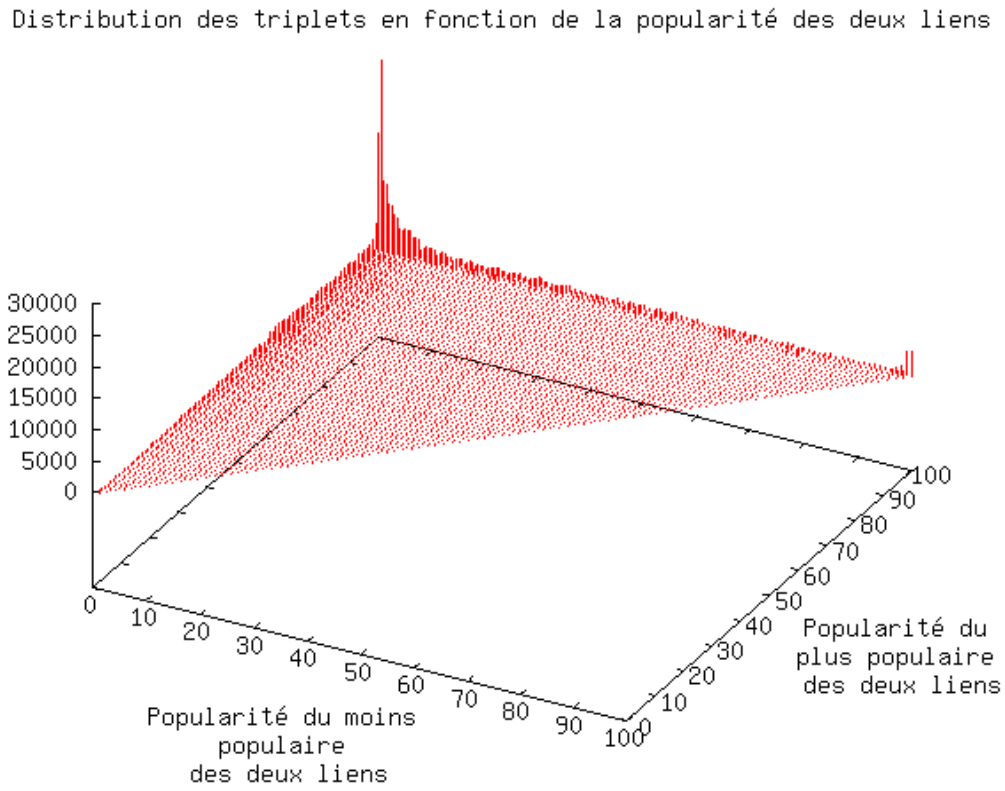


(b) tomographie $T_{2006-08}$



(c) tomographie $T_{2006-08}$

FIG. 2.15 – Répartition du nombre d'éléments topologiques observés en fonction de leur popularité.



(a) tomographie $T_{2006-08}$

FIG. 2.16 – Répartition du nombre de triplets d'AS en fonction de la popularité des liens.

d) Observation des atomes de politique de routage en fonction de la popularité des liens

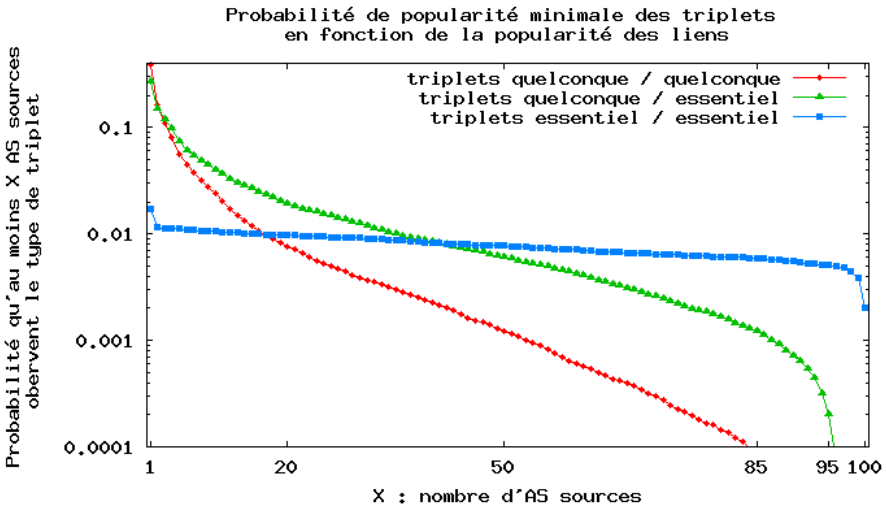
La figure 2.16 montre le nombre de triplets d'AS en fonction de la popularité des deux liens. Cette figure montre clairement qu'il existe trois types de triplets :

Triplets avec deux liens quelconques : Les triplets d'AS mesurés avec deux liens quelconques sont uniformément répartis par rapport à la popularité des deux liens.

Triplets avec un lien rare : Les triplets d'AS mesurés avec des liens rares sont en grand nombre par rapport aux triplets quelconques de liens mesurés par peu d'AS sources. La mesure de ces triplets est très sensible au nombre d'AS sources utilisés.

Triplets avec un lien essentiel : Les triplets d'AS mesurés avec un lien essentiel sont de deux types. Nous les distinguons ci-dessous.

Pour mieux distinguer les différents triplets mesurés, on s'intéresse maintenant à la probabilité qu'un triplet contienne deux liens d'une certaine popularité par rapport à la



(a) tomographie $T_{2006-08}$

FIG. 2.17 – Répartition des différentes popularités de triplets d'AS en fonction de la popularité des liens.

popularité du triplet figures 2.17 et 2.18, et dans le tableau 2.4.

Triplets →	rares	quelconques	essentiels
Liens (rare,*)	32.6%		
Liens (qcq,*)	67.6%	17.8%	
Liens (essentiel,*)	30.5%	10.5%	0.5%
Liens (rare,essentiel)	12.6%		
Liens (essentiel,essentiel)	0.6%	0.6%	0.5%

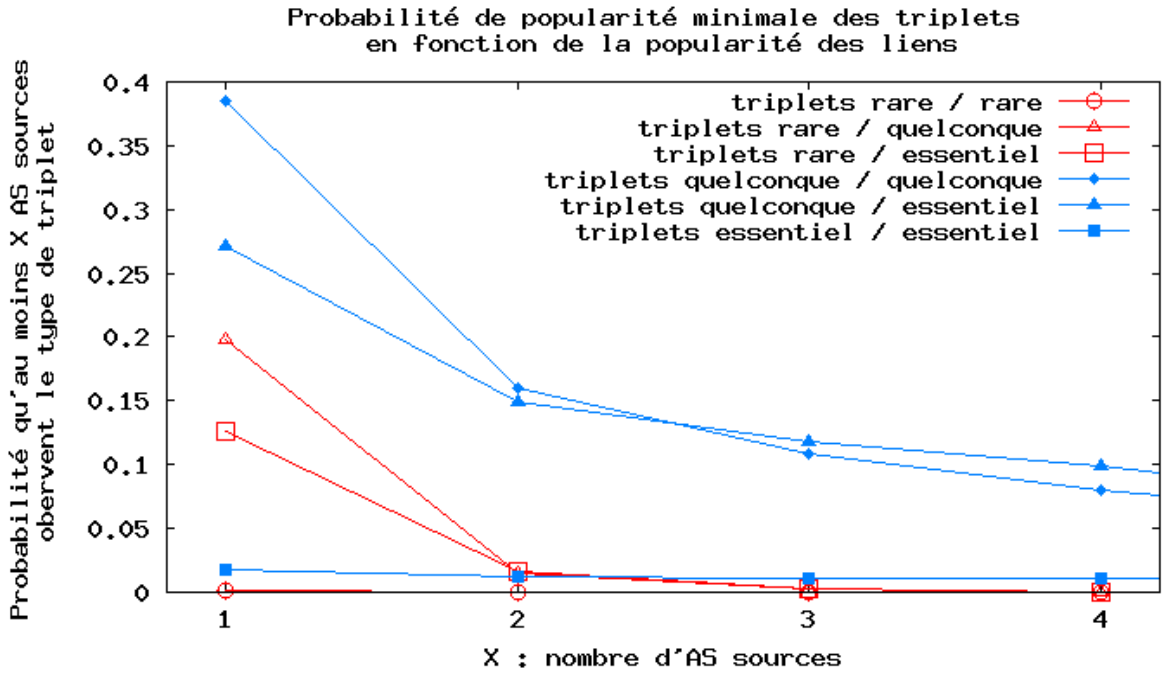
Nombre minimal d'AS sources	1	4	20	50	97
Triplets avec Liens (qcq,*)	60%	17.8%	2.5%	0.7%	0%
Triplets avec Liens (rare,essentiel)	12.6%	0%			
Triplets avec Liens (essentiel,essentiel)	1.7%	1.1%	0.96%	0.76%	0.5%

TAB. 2.4 – Popularité de certains triplets en fonction de la popularité des liens

Les triplets les plus mesurés sont des triplets qui font intervenir des liens quelconques : cela représente deux triplets sur trois. Il y a un triplet sur trois qui fait intervenir un lien rare, et il y a aussi un triplet sur trois qui fait intervenir un lien essentiel. Si on s'intéresse aux triplets qui font intervenir deux liens essentiels, on remarque qu'ils sont répartis dans les classes de triplets rares, quelconques et essentiels. La mesure avec une tomographie est donc d'autant plus juste que l'on augmente le nombre d'AS sources. Mais à partir d'un nombre significatif d'AS sources (par exemple 100), la vision d'AS sources supplémentaires n'apportera en moyenne que peu de triplets (essentiels,essentiels) et une

majorité de triplets avec des liens quelconques.

Le pourcentage du nombre de triplets visible d'un unique AS est de 64.6%. Un AS source supplémentaire ajouterait donc en moyenne $\frac{1}{100} * \frac{64}{100} \simeq 0.64\%$ de triplets en plus. Les atomes de politique de routage sont donc observés avec un biais potentiel car ajouter 25% d'AS sources en plus ajouterait 12% de triplets. Seulement 0.5% de tous les triplets (essentiels,essentiels) sont visibles d'un unique AS. Un AS source supplémentaire ajouterait donc en moyenne $\frac{1}{100} * \frac{0.5}{0.6+0.6+0.5} \simeq 0.30\%$ de triplets (essentiels,essentiels) en plus. Les atomes de politique de routage qui font intervenir deux liens essentiels sont observés plus justement que les triplets. En ajoutant 25% d'AS sources, on ajouterait en effet 6% de triplets avec deux liens essentiels.



(a) tomographie $T_{2006-08}$

FIG. 2.18 – Répartition des différentes popularités de triplets d'AS en fonction de la popularité des liens.

2.3 Analyse des matrices de politique de routage

Dans la littérature, la topologie de l'Internet a souvent été étudiée sous la forme d'un graphe inter-AS sans préfixes. On propose d'étudier l'effet de telles simplifications comparé à l'échelle de détail d'une tomographie. Pour cela, on analyse la propagation des préfixes de chaque AS origine. On montre que les politiques de routage ne sont pas définies à

l'échelle des préfixes. On tente ensuite d'identifier pour chaque AS des groupes de voisins similaires. On cherche alors à évaluer la véracité de l'hypothèse qui stipule que chaque AS règle sa politique de routage en fonction des accords économiques avec ses AS voisin. Cette hypothèse aboutit a priori à un grand nombre de routages similaires[17]. On montre qu'en effet, chaque AS traite ses voisins de façon très similaire.
On poursuit notre étude avec l'analyse de la hiérarchie du routage BGP. La connaissance d'un graphe inter-AS ou des chemins d'AS sans préfixes ne permet pas une détermination de cette hiérarchie en raison du biais de mesure du placement des AS-sondes. On introduit donc un nouvel indicateur qui conduit à une classification hiérarchique tenant compte des préfixes transités et du biais de mesure introduit par le placement des AS-sondes.

2.3.1 Routage des groupes de préfixes par AS origine

Une tomographie BGP conserve une association entre des chemins d'AS et les préfixes observés sur les chemins. On s'intéresse à savoir s'il est possible de supprimer cette association. Le principe vise à ne conserver que les chemins d'AS et les matrices de politique d'annonce pour représenter la connaissance des associations entre chemins et préfixes.

a) Chemins similaires des AS sources vers les préfixes d'un même AS origine

Soit un AS X dont la matrice de politique d'annonce n'est pas vide. Si les différents préfixes de l'AS X qui sont exportés vers un AS voisin Y sont traités de la même manière par Y, alors on peut penser que les voisins de Y, qui reçoivent des annonces BGP vers les préfixes de l'AS X, appliquent les mêmes règles de routage pour les différents préfixes.
La figure 2.19 montre le pourcentage d'AS origines en fonction du nombre d'AS sources qui observent exactement les mêmes chemins vers les différents préfixes des AS origines concernés. Un point (x, y) montre qu'il y a $y\%$ des AS origines pour lesquels un maximum de x AS sources disposent pour chacun de ces AS origines, des mêmes chemins vers l'ensemble de ses préfixes.

[17]Cette propriété pourrait être utilisée d'ailleurs pour configurer des agrégations de routes et ainsi améliorer la convergence du protocole BGP.

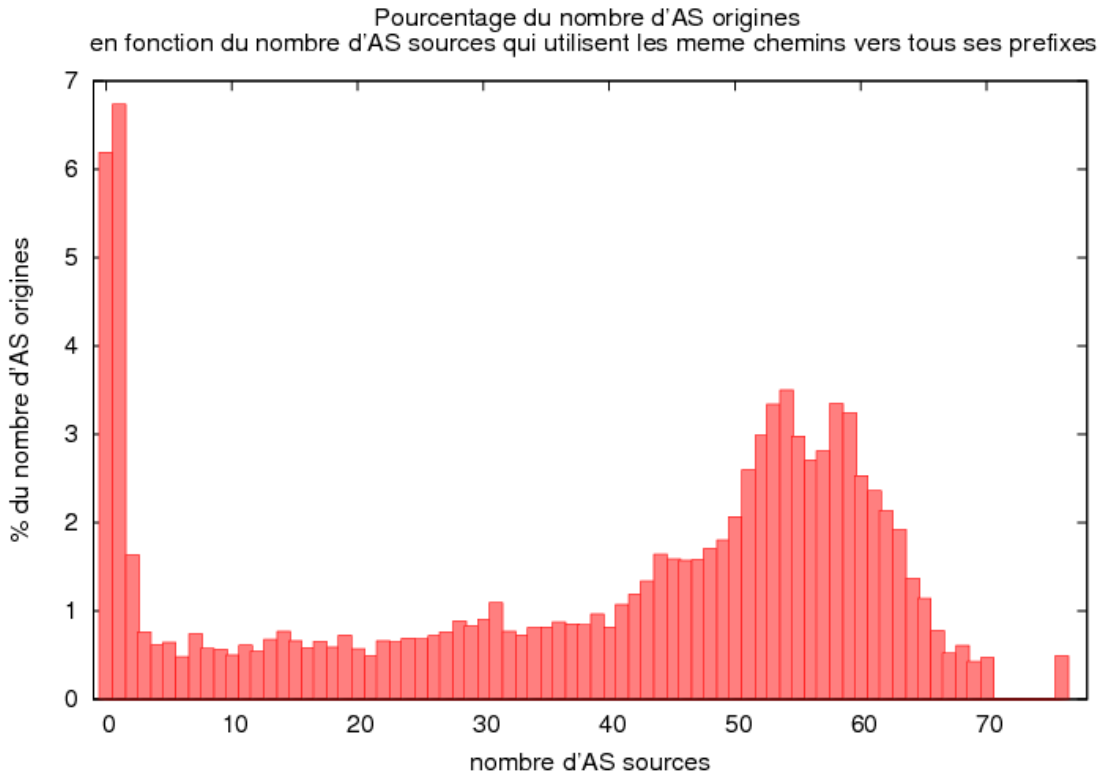


FIG. 2.19 – Nombre d'AS origines dont le même nombre d'AS sources observe exactement les mêmes chemins vers les différents préfixes de l'AS origine.

b) Chemins similaires des AS sources vers les préfixes d'un même AS origine distribués à un AS voisin

Dans la pratique, chaque AS origine sélectionne les voisins pour lesquels il annonce ses propres préfixes. On reporte figure 2.20, le nombre de couples (AS origine, AS voisin de l'AS origine) dont le même nombre d'AS sources observe exactement les mêmes chemins vers les différents préfixes de l'AS origine en passant par l'AS voisin. Un point (x, y) montre que $y\%$ des couples (AS origine, AS Y voisin de l'AS origine) sont tels que, pour chacun, un maximum de x AS sources utilisent les même chemins vers l'AS origine en passant par le voisin en avant dernier.

Les routes similaires observées des AS sources vers les AS origines, montrent qu'on ne peut pas clairement supposer que les préfixes d'un AS sont reçus avec les mêmes routes par un AS donné. Cependant, en tenant compte de la sélectivité des annonces de préfixes, on peut observer que la majorité des préfixes sont reçus avec la même route d'AS. Par conséquent, un AS configure généralement sa politique de routage indépendamment des différents préfixes annoncés par les voisins de ses voisins. Cette propriété empirique du

routage BGP permet d'obtenir une bonne approximation des cheminements AS-AS de préfixes grâces aux deux données suivantes :

– les cheminements AS-AS,

– les matrices de politique de routage d'annonce de chaque AS.

L'approximation du routage des préfixes à partir des matrices de politique d'annonce sera exploitée dans le chapitre 4 dans le modèle de graphe avec préfixes.

2.3.2 Singularité des matrices de politique de routage

On s'intéresse à partitionner les voisins d'un AS en fonction de ses deux matrices de politique de routage avec une méthode de clustering. On détermine pour chaque AS, des classes de voisins appelées "clusters". Chaque cluster regroupe les AS observés avec les mêmes éléments dans les deux matrices de politique de routage de l'AS considéré (on ne tient pas compte des poids des atomes). On met en évidence que parmi les AS voisins d'un même AS, il existe peu de classes d'après à l'identification des routages similaires dans les matrices de politique de routage.

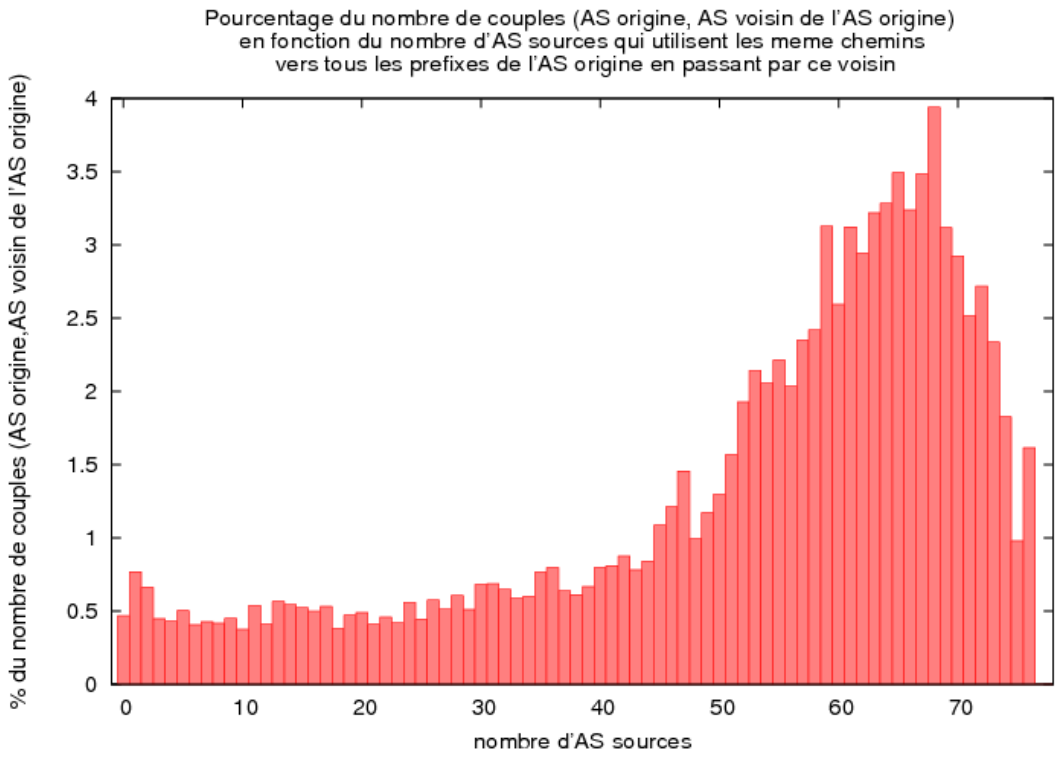


FIG. 2.20 – Nombre de couples (AS origine,AS voisin de l'AS origine) dont le même nombre d'AS sources observe exactement les mêmes chemins vers les différents préfixes de l'AS origine en passant par l'AS voisin de l'AS origine.

a) Clustering des matrices de politique de transit de chaque AS

Si les accords d'interconnexion sont uniquement des accords de peering ou des accords de transit, alors les matrices de politique de routage de transit sont supposées être constituées de trois parties. Dans ce cas, si on regroupe les colonnes et les lignes de chaque matrice de politique de transit en fonction du type de voisin, on obtient :

$$\forall X \in AS,\ M_{transit}(X) = \begin{array}{c} \\ (C) \\ (P) \\ (F) \end{array} \begin{array}{c} \begin{array}{ccc} (C) & (P) & (F) \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \end{array} \quad \text{où} \begin{cases} (C) \text{ sont les clients de } X \\ (P) \text{ sont les peers de } X \\ (F) \text{ sont les fournisseurs de } X \end{cases}$$

Soit un AS X. On définit la dissemblance $Dis_X^{transit}(Y,Z)$ perçue par X entre deux voisins Y et Z comme la somme des différences entre les lignes et les colonnes i_X (AS X) et i_Z (AS Z) de la matrice $M_{transit}(X)$.

$$Dis_X^{transit}(Y,Z) = \sum_{\substack{i=1 \\ i \notin \{i_Y, i_Z\}}}^{deg(X)} \frac{1}{deg(X)} . \left| (M_{transit}(X))_{i,i_Y} - (M_{transit}(X))_{i,i_Z} \right|$$

$$+ \sum_{\substack{i=1 \\ i \notin \{i_Y, i_Z\}}}^{deg(X)} \frac{1}{deg(X)} . \left| (M_{transit}(X))_{i_Y,i} - (M_{transit}(X))_{i_Z,i} \right|$$

b) Clustering des matrices de politique d'annonce de chaque AS

Les annonces sélectives de NLRI origines (représentées par les matrices d'annonce) sont effectuées soit de façon normale[18], soit pour faire du partage de charge, soit pour d'autres objectifs d'ingénierie de trafic. Dans tous les cas, on doit pourvoir observer un partitionnement simple de la matrice de politique d'annonce grâce au partitionnement des différents NLRI annoncés par un AS à ses voisins. Si on compare les NLRI annoncés aux AS voisins, on obtient en théorie une matrice formée de colonnes et de lignes qui peuvent être

[18] C'est le cas où un AS ne restreint pas l'ensemble des AS voisins auxquels il annonce ses préfixes, à part pour respecter les accords économiques.

similaires :

$$M_{annonce}(X) = \begin{array}{c} \\ \{NLRI_i\} \end{array} \begin{array}{c} \begin{array}{ccc} [X] & & \{Y_j\} \end{array} \\ \begin{pmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{pmatrix} \end{array}$$

Soit un AS X. On définit la dissemblance $Dis_X^{annonce}(Y,Z)$ perçue par X entre deux voisins Y et Z comme la somme des différences entre les colonnes i_X (ASX) et i_Z (ASZ) de la matrice $M_{annonce}(X)$.

$$Dis_X^{annonce}(Y,Z) = \sum_i \frac{1}{|NLRI_{origin}(X)|} \cdot \left| (M_{annonce}(X))_{i,i_Y} - (M_{annonce}(X))_{i,i_Z} \right|$$

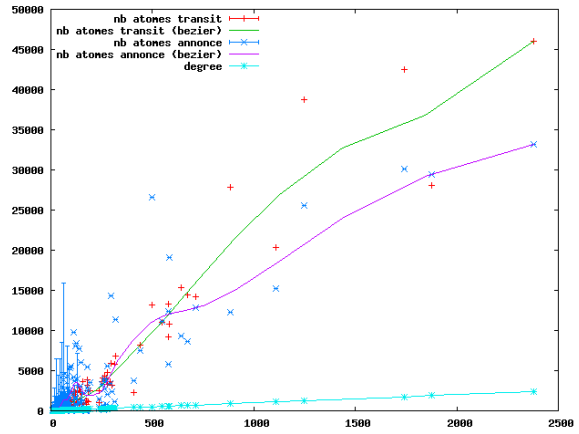
c) Clustering des matrices de politique de routage

En pratique, les matrices de politique de routage sont difficiles à observer en entier. Une tomographie BGP permet de reconstituer une partie significative des matrices. Pour illustrer le phénomène de similarité des voisins d'un AS, nous proposons d'identifier pour chaque AS, les AS voisins qui semblent avoir été traités de la même manière du point de vue des atomes de politique de routage de transit et d'annonce. La méthodologie de clustering va effectuer des regroupements d'AS en fonction d'une valeur de dissemblance. Pour chaque AS X, pour chaque paire d'AS voisins Y et Z, on définit $Dis_X(Y,Z)$ la dissemblance de routage de Y par rapport à Z :

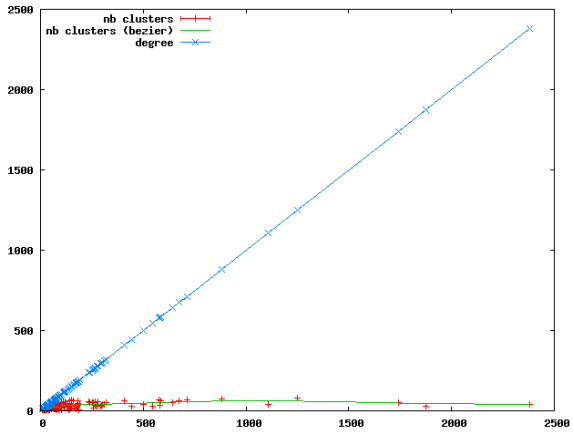
$$Dis_X = \begin{array}{c} \\ Y \end{array} \begin{array}{c} \begin{array}{ccc} (X) & & Z \end{array} \\ \begin{pmatrix} \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \\ \dots\dots\dots & Dis_x(Y,Z) & \dots\dots\dots \\ \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \end{pmatrix} \end{array}, \forall X \in AS$$

$$Dis_X(Y,Z) = \sqrt{\left(Dis_X^{transit}(Y,Z)\right)^2 + \left(Dis_X^{annonce}(Y,Z)\right)^2}$$

La matrice de dissemblance de chaque AS X, est exploitée avec l'algorithme reporté dans [48]. Le nombre de groupes de voisins obtenus pour chaque AS X est noté $cluster(X)$. La figure 2.21(b) montre le nombre de clusters obtenus pour chaque AS X en fonction de son degré. Les AS sont classés par degré croissant. La figure 2.21(a) montre le nombre



(a) Nombre d'atomes de transit et d'annonce en fonction du degré d'un AS



(b) Nombre de clusters en fonction du degré d'un AS

FIG. 2.21 – Taille des matrices de politique de routage et nombre de clusters en fonction du degré de chaque AS

d'atomes de transit et d'annonce minimal, maximal et moyen en fonction du degré d'un AS. La figure 2.22, utilisée avec une échelle log-log, synthétise les différences entre les ordres de grandeur suivants :

$$|atomes(X)| \gg |degre(X)| \gg |clusters(X)|, \quad \forall X \in AS$$

On peut donc supposer, que chaque AS n'a que très peu d'AS voisins traités de façon différente. Ce résultat peut être exploité pour justifier la définition des matrices de politique de routage qui remplacent la vision "chemins" de chaque AS obtenue avec une tomographie BGP.

2.3.3 Hiérarchie du routage interdomaine

Le réseau Internet est structuré en couches hiérarchiques. L'organisation topologique des AS est telle que le nombre de liens observés dans le graphe interdomaine est faible par rapport au nombre de noeuds, et les distances observées entre les paires origine-destination sont faibles (cf. figure 2.25).

On détermine ici les trois principaux niveaux hiérarchiques du routage par une nouvelle méthode qui utilise les matrices de politique de routage de transit pour déterminer les extrémités de cette hiérarchie, et qui utilise les atomes de politique de transit-préfixe en fonction de chaque AS source pour déterminer le coeur du réseau.

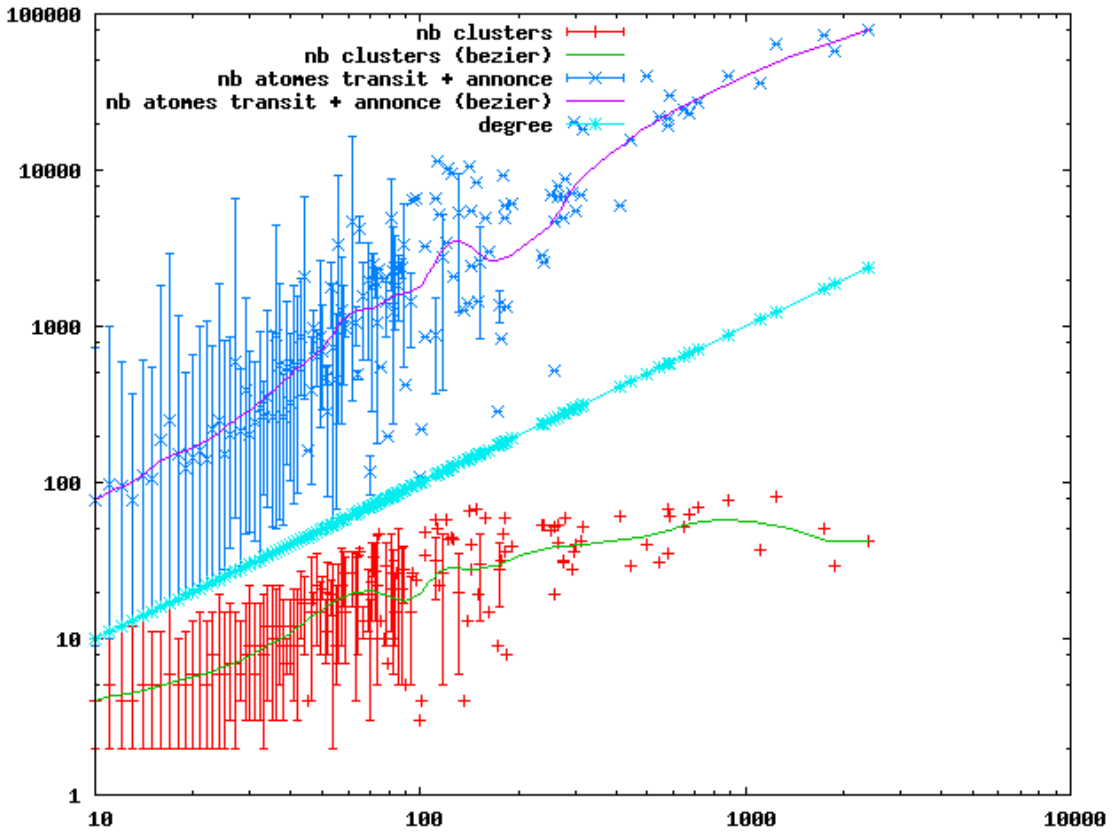


FIG. 2.22 – Nombre de clusters et nombre d'atomes de politique de routage d'un AS en fonction de son degré

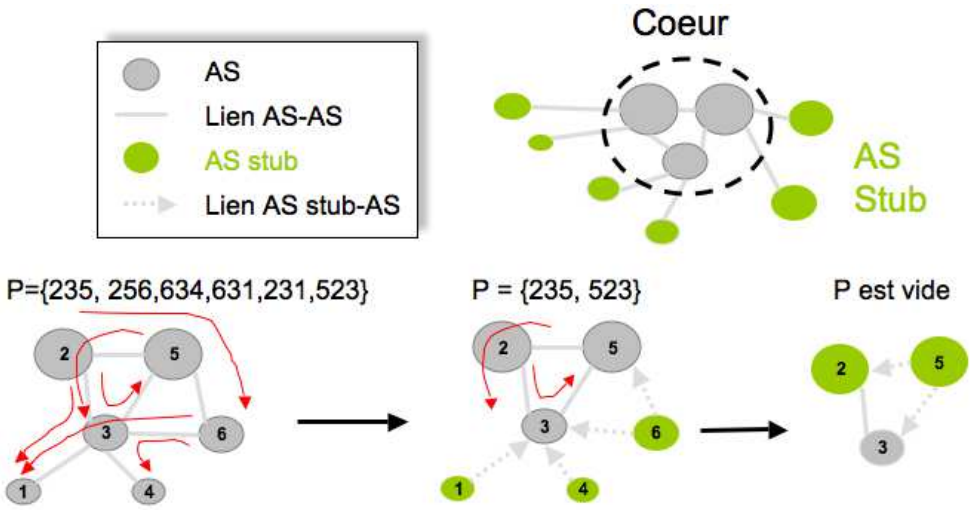


FIG. 2.23 – Illustration de la hiérarchie simple induite par les triplets

a) AS stubs d'ordre n

On note (R_{stub}) la phase de réduction suivante sur un ensemble de triplets :

Algorithme 1. *Réduction exacte* (R_{stub})

Entrée : triplets $\mathcal{T} = \left\{ (u_i^{(1)}, u_i^{(2)}, u_i^{(3)}) / \, i \in \{1, \ldots, m_0\} \right\}$

$i = 0$

$\mathcal{T}_i = \mathcal{T}$

Tant que $Stub_i = \{u \,/\, \nexists (v,u,w) \in \mathcal{T}_i\} \neq \varnothing$ faire :

$Core_i = \{u \,/\, \exists (v,u,w) \in \mathcal{T}_i\}$

$\forall u \in Stub_i,\ R_{stub}(u) = i$

$\mathcal{T}_{i+1} = \left\{ (u_i^{(1)}, u_i^{(2)}, u_i^{(3)}) \in \mathcal{T}_i \cap (Core_i \times Core_i \times Core_i) \right\}$

$i = i + 1$

fin tant que

Sortie : ordre R_{stub} : $\{u_i \in V_0\}_{i \in \{1,\ldots,n_0\}} \rightarrow \left\{ r_i \in \mathbb{N}^+ \right\}_{i \in \{1,\ldots,n_0\}}$

ordre maximal $r_{max} \geq 0$

fin de l'algorithme

En appliquant la procédure ci-dessus, on obtient une partition de l'ensemble des AS en classes d'AS stubs. Les AS éliminés par l'algorithme à chaque itération sont appelés AS stub d'ordre n, où n est le rang $R_{stub}(u)$ pour tout u un AS. Les AS issus de l'ensemble $Core_{r_{max}}$ sont appelés AS $core - triplets$.

b) Transit sur les éléments topologiques et hiérarchie du routage

La méthode de réduction (R_{stub}) permet d'identifier les AS aux extrémités des arbres de routage déduits des chemins de chaque AS source. Les AS core-triplets qui restent à la fin de l'algorithme sont en fait complètement entre-mêlés dans les triplets conservés. Il n'est donc pas possible de déterminer une structure plus précise des AS restants grâce aux triplets de l'ensemble $\mathcal{T}_{r_{max}}$. En effet les erreurs de routage économiques et les routage spécifiques observés pour très peu de préfixes biaisent le résultat. En particulier, certains AS autorisent les routages entre deux de leurs fournisseurs au mépris des règles économiques de routage de base. Ces AS apparaissent alors comme des AS de transit. Les indicateurs suivants sont inutilisables pour exhiber une structure correcte du coeur de l'Internet :

transit triplet : $_{(Y)} = |\{(X,Y,Z) \in \mathcal{T}\}|$, nombre de triplets d'AS symétriques indiquant un AS en transit.

transit degré : $_{(Y)} = \left| \left\{ (Y,X) / \begin{array}{c} \exists (X,Y,Z) \in \mathcal{T} \\ \text{ou } \exists (Z,X,Y) \in \mathcal{T} \\ \text{ou } \exists (Y,X,Z) \in \mathcal{T} \\ \text{ou } \exists (Z,Y,X) \in \mathcal{T} \end{array} \right) \in \mathcal{T} \right\} \right|$, nombre de voisins transités par un AS.

Pour correctement distinguer les AS restants, nous allons utiliser les matrices de politique transit-préfixe pour tenir compte des préfixes. On note $\mathcal{T}_p$ l'ensemble des triplets d'AS avec préfixes. Grâce à la connaissance des préfixes, les deux indicateurs suivants sont connus :

transit AS-préfixe : $_Y = |\{p / \exists (p,X,Y,Z) \in \mathcal{T}_p\}|$, nombre de préfixes transités par l'AS Y.

espace AS-préfixe : $e_Y = \left| \bigcup \{adresse \in p / \exists (p,X,Y,Z) \in \mathcal{T}_p\} \right|$, taille de l'espace d'adresse transité par l'AS Y.

Les indicateurs présentées ci-avant sont influencés par le placement des sondes et le nombre de routes apportées par chaque AS source comme le confirme la partie gauche de la figure 2.11. On introduit un indicateur pour discerner les AS du coeur du routage Internet et filtrer les AS trop proches de nos sondes ou ayant configuré du transit entre deux fournisseurs de plus haut niveau avec l'indicateur suivant :

pourcentage de l'espace d'adresses transitées moyen par AS source :

$$em_{(Y)} = \sum_{S \in AS_{sources}} \frac{\left| \bigcup \{@ \,/\, @ \in p\} \right|_{\exists (p,X,Y,Z) \in \mathcal{T}_p(S)}}{|AS_{sources}|}$$

L'indicateur $em_{(Y)}$ se calcule en comptabilisant pour les chemins de chaque AS source, la taille de l'espace d'adresses transités par l'AS Y .

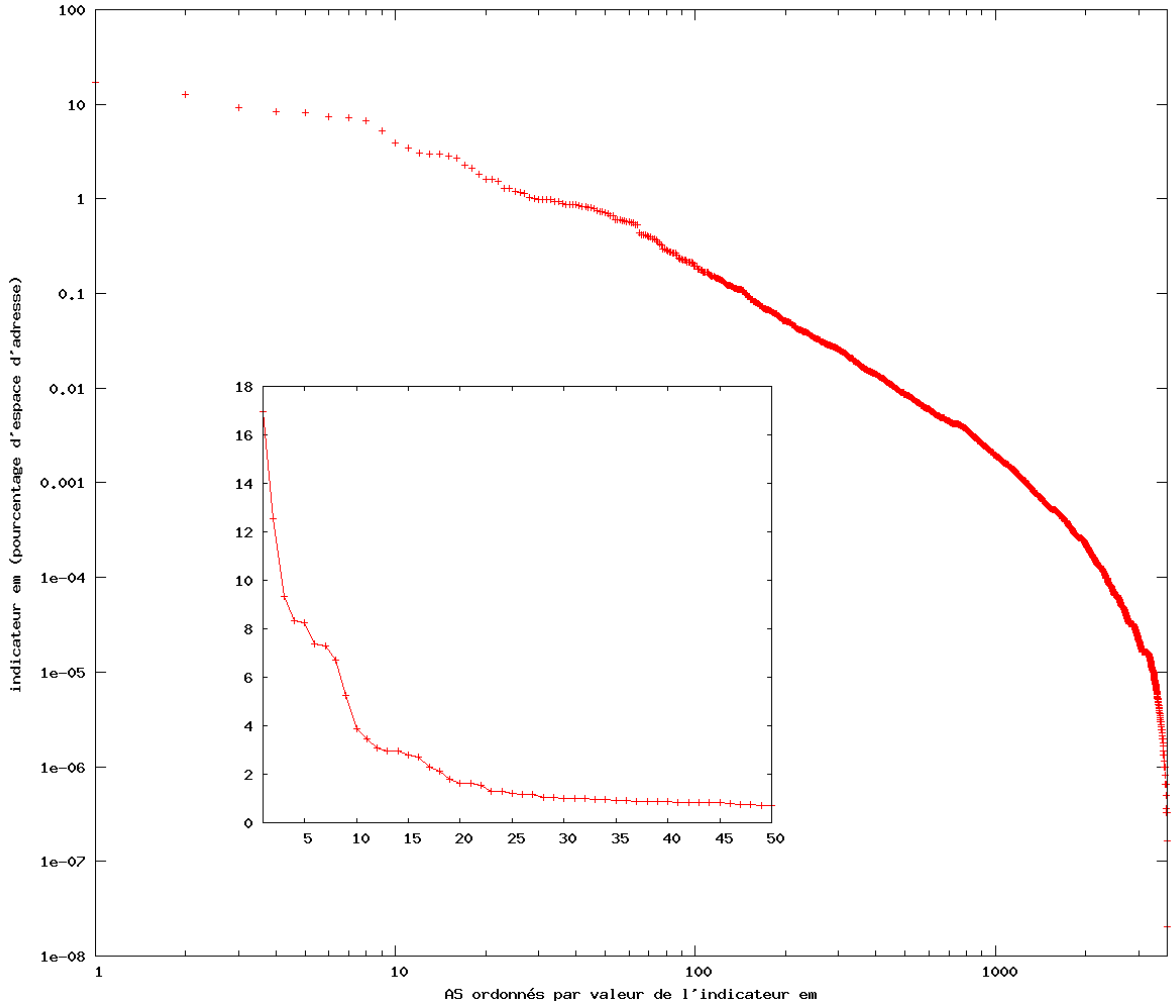


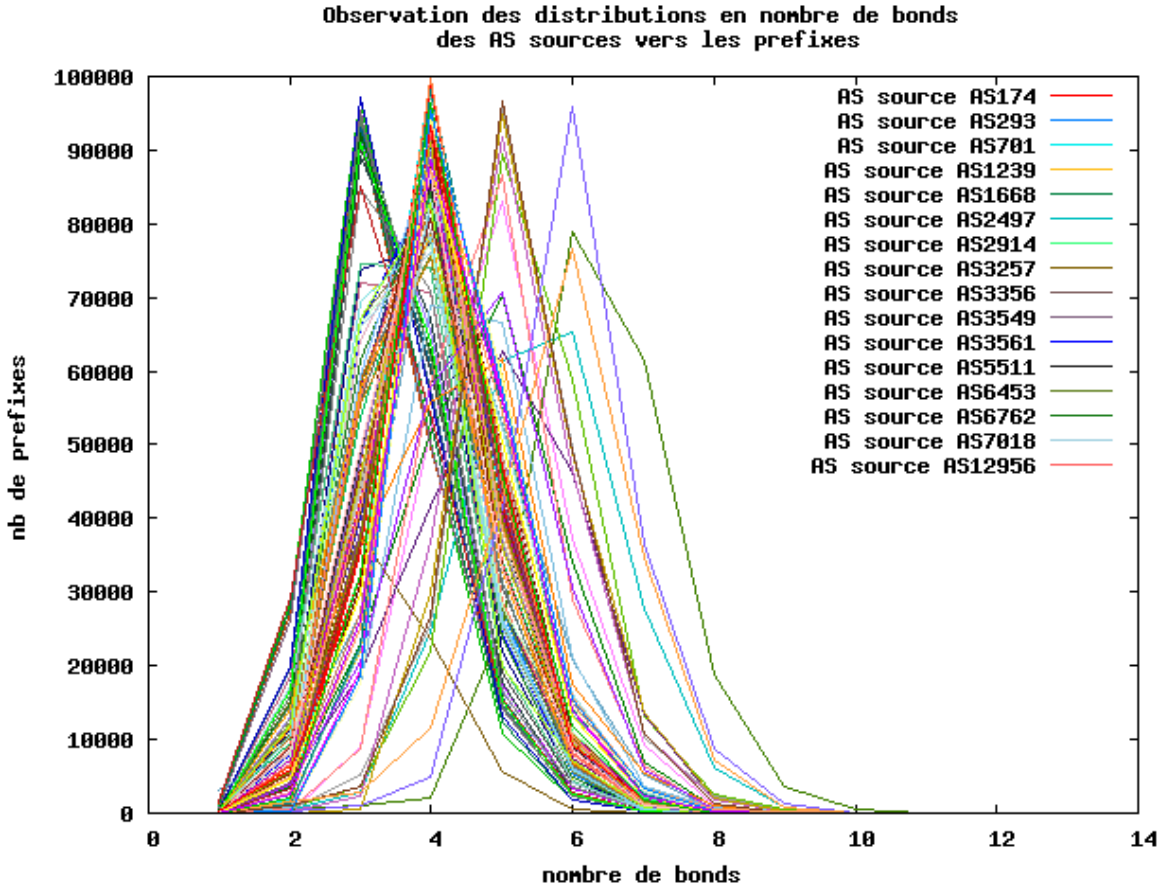
FIG. 2.24 – Pourcentages des espaces d'adresses transitées moyen par AS source pour la tomographie $T_{2006-08}$

Sur la figure 2.24 on peut voir que la distribution des valeurs de *em* suit une loi de puissance à partir du dixième AS jusqu'au 2000 ème. Cette caractéristique remarquable des valeurs de *em* permet d'établir un classement efficace pertinent des fournisseurs d'accès Internet.

c) Hiérarchie du routage en trois couches

Il est possible de mettre en évidence des niveaux hiérarchiques dans l'Internet. En effet, il existe des opérateurs Tiers-1 avec une connectivité globale gratuite (niveau le plus haut de la hiérarchie), et il existe des clients finaux avec une connectivité totalement payante (niveau le plus bas de la hiérarchie). On conjecture que l'Internet est constitué de trois

grandes couches. On peut vérifier par exemple sur la figure 2.25 que la distribution en nombre de sauts des AS sources très hauts dans la hiérarchie vers l'ensemble des préfixes forme une cloche avec un pic à la valeur 3. Cette valeur 3 correspond à des routes destinées aux préfixes de clients de clients ou aux clients de peers.



(a) Les AS de la légende sont ceux dont les distributions ont un pic à 3 bonds.

FIG. 2.25 – Distribution en nombre de sauts entre les AS sources et les préfixes

Voici la classification que nous proposons pour déterminer ces trois couches hiérarchiques :

AS Tiers-3 (Couche Stub d'ordre $n < r_{max}$) : un AS éliminé à l'itération n de l'algorithme (R_{stub}) fait partie de la couche hiérarchique numéro 3,

AS quasi-Tiers-1 : Les premiers AS classés par valeur de l'indicateur *em* sont les AS Tiers-1,

AS Tiers-2 : les autres AS.

Grâce à cette classification hiérarchique, on peut identifier des atomes de politique de routage économiquement invalides puisque l'on peut savoir si un AS de petite taille permet

le transit de certains préfixes entre deux AS de plus haut niveau. On peut aussi grâce à cette classification, décider d'un type d'accord d'interconnexion pour des liens inter-AS qui n'interviennent dans aucun triplet d'une tomographie.

2.4 Inférence de quelques propriétés économiques

La classification hiérarchique des AS est une première étape pour déterminer la structure hiérarchique de l'Internet. On montre ici comment déterminer les entités administratives responsables des AS et comment géolocaliser une grande partie des NLRI annoncés dans une tomographie BGP. On discute aussi de la manière dont un ranking précis des opérateurs peut être définit.

2.4.1 Connaissance des entités administratives des AS

Pour connaître l'entité administrative responsable d'un AS, il faut interroger les registres de routage. L'allocation des numéros d'AS est telle qu'un même AS peut être référencé par plusieurs registres. Les registres étant publiques, les difficultés rencontrées pour associer une entités administrative à un AS sont seulement d'ordre opérationnelles. Par contre, pour connaître les différents AS appartenant à une même entité administrative, on propose de procéder en plusieurs étapes :

Étape 1 : génération d'une liste de groupes d'AS. Dans chaque groupe, les différents AS appartiennent potentiellement à une même structure administrative. On a retenu trois méthodes différentes pour déduire de telles relations. Une première méthode consiste à identifier les AS administrés par les mêmes NOC[19] d'après les listes publiques disponibles sur le web [85]. Une seconde méthode consiste à télécharger les registres de routage complets et à en extraire les AS dont l'AS-NAME ou le nom de l'administration est similaire (même préfixe, même suffixe ou une distance de hamming[20] proche). Afin de trouver les candidats potentiels, on peut chercher les AS-NAME et les descriptions de chaque opérateur Tiers-1. Une troisième méthode consiste à extraire les AS en accord de sibling d'après des informations publiques déterminées par des algorithmes de la littérature (cf. [91]).

[19] Network Operation Center

[20] La distance de hamming entre deux chaînes d'éléments correspond au nombre de modifications élémentaires nécessaires pour transformer la première chaîne en deuxième chaîne.

Étape 2 : extraction des informations des registres de routage pour chaque AS de chaque groupe. Pour chaque groupe, les AS-NAME ou les descriptions de chaque AS deux à deux doivent être similaires d'après la distance de hamming. On peut ensuite vérifier manuellement pour chaque groupe, si les AS sont effectivement associés à la même administration.

2.4.2 Localisation des préfixes aux pays

Attribuer un pays à un AS est difficile. En effet un AS peut correspondre à un réseau d'étendue mondiale. Cependant pour ce qui concerne les NLRI, il est plus probable que les machines d'une même plage soient localisées dans un même pays[21]. Lorsqu'une plage d'adresse IP est allouée à une entité administrative, l'acquéreur du NLRI indique un seul pays dans le registre [56].

Les NLRI annoncés via le protocole BGP sont des plages découpées parmi les plages référencées dans les registres de routage. Pour savoir quel est le pays de référence d'un NLRI, il faut retrouver le parent le plus proche dans l'arbre d'inclusion des préfixes et pour lequel on connaît le pays de référence (voir figure 2.26). Le nombre de préfixes considérés par un routeur BGP peut atteindre 200 000. Manipuler l'arbre des préfixes et effectuer des recherches d'inclusion d'un préfixe dans cet arbre peuvent être très lents. On a défini une structure de données (cf. annexe D) qui permet d'obtenir d'excellentes performances. En effet, la structure exploite un double arbre qui permet la recherche et l'insertion d'un élément en $O(log(n))$. De plus, la recherche du préfixe le plus spécifique contenant un préfixe s'effectue en $O(log(n))$. On peut signaler que le temps de recherche du pays d'un préfixe par notre structure d'arbre est plus de 100 fois plus rapide que le temps d'une recherche par un parcours séquentiel des préfixes whois[22].

L'algorithme pour générer le pays de chaque NLRI est le suivant :

1. Télécharger les fichiers des registres de routage [56] fournissant une indication de nationalité pour les préfixes qu'ils enregistrent.
2. Construire l'arbre de recherche avec les préfixes téléchargés.
3. Pour chaque NLRI, rechercher le préfixe de l'arbre le plus spécifique qui le contient.

[21] A l'exception des plages d'adresses utilisées par les opérateurs pour leurs équipements, et les plages d'adresses publiques utilisées pour des réseaux distants interconnectés par réseaux privés virtuels (VPN). Il semblerait aussi que l'allocation des plages IP version 6 ne respecte pas notre hypothèse.

[22] Sur un ordinateur cadencé à 2Ghz avec 1Go de mémoire, cette recherche s'effectue en moins d'une seconde grâce à l'arbre.

Si ce préfixe n'est pas 0/0, alors associer le pays de référence du préfixe contenant le NLRI.

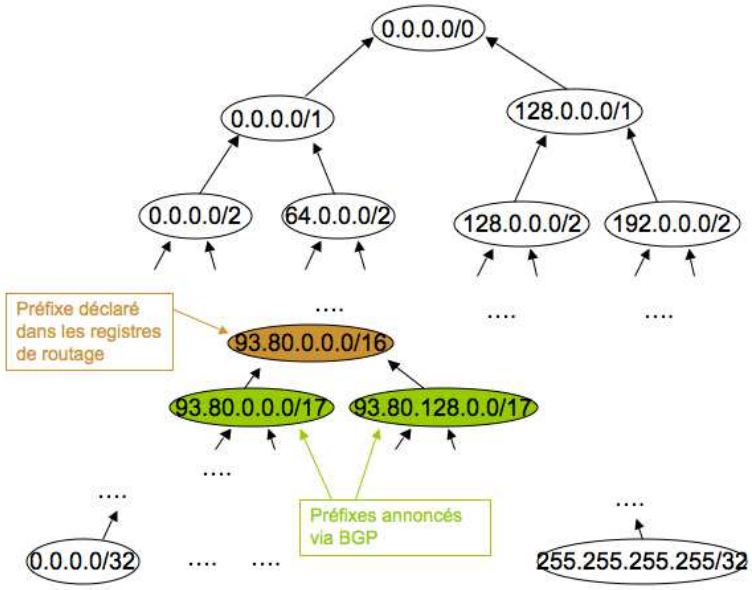


FIG. 2.26 – Découpage des préfixes déclarés dans les registres de routage pour les annonces via le protocole BGP

L'algorithme précédant se révèle très efficace comme indiqué dans le tableau 2.5. En effet,avec la seule information des registres de routage, entre 15 et 16.3% des NLRI BGP sont identifiés avec un pays d'enregistrement (voir colonne "préfixes BGP/whois" du tableau). Avec notre approche, on parvient à inférer 80% de préfixes supplémentaires (voir colonne "préfixes BGP inférés"), ce qui donne approximativement 95% des nationalités identifiées pour les NLRI.

Tomographie	préfixes BGP	préfixes BGP/whois	préfixes BGP inférés
$T_{2006-08}$(tous les filtres)	209 876	32 765 ($\simeq$ 15%)	168 347 ($\simeq$ 80%)

TAB. 2.5 – Application de l'algorithme d'inférence des nationalités des NLRI annoncés via le protocole BGP

Remarque : on peut utiliser cette géolocalisation des préfixes pour attribuer des nationalités possibles pour un AS en fonction de ses annonces. On peut aussi attribuer attribuer des nationalités supplémentaires à un AS lorsqu'il est connecté à des AS Stubs d'ordre 1 et mono-pays.

2.4.3 Classement des AS et marché économique des opérateurs

La structure hiérarchique du réseau Internet devrait être déterminée d'après les connectivités des AS vers les différents NLRIpublics propagés sur l'Internet. La connectivité de chaque AS vers les NLRI est dépend de tous les accords accords économiques d'interconnexion (les accords désignent les principales règles de propagation et de rejet de routes eBGP). Le positionnement d'un AS dans cette hiérarchie est complexe à définir. Un premier élément pour définir cette hiérarchie consiste à élaborer un classement (ranking) des AS en fonction de leur taille. Il existe une multitude de méthodes de classement (ranking) possibles : en fonction du nombre de préfixes annoncés et des agrégations [101], du degré d'un AS [176], de la centralité d'un AS [90], de la connectivité rich club d'un AS [201], du degré joint [120], du nombre de préfixes clients d'un AS [40, 95]...
Pour connaître précisément la connectivité d'un AS, il faut connaître ses routes vers les préfixes. Cela nécessite l'**inférence des routages** valides d'un AS vers le reste des préfixes ou des AS de l'Internet. Pour connaître les clients d'un AS, il faut **inférer les accords d'interconnexion** inter-AS. Le problème de ranking des AS, est un problème très flou mais les deux informations suivantes sont souvent nécessaire pour répondre à un grand nombre de questions :

Clients d'un AS : pour connaître les accords d'interconnexion entre AS, on utilise un des deux algorithmes présenté dans le chapitre 3.

Connectivité : pour les AS sources de nos tomographies, on connaît la connectivité vers presque tous les préfixes BGP (voir figure 2.25). Pour les autres AS, on utilise le modèle de graphe et l'algorithme présentés dans le chapitre 4 pour inférer les chemins d'AS vers l'ensemble des préfixes.

Chapitre 3

Inférence des accords d'interconnexion entre AS

Le problème d'inférence des accords d'interconnexion entre AS est un problème inverse. Il s'agit de déterminer la nature économique des relations d'interconnexion entre les AS de l'Internet d'après l'observation des données de routage. Les tomographies BGP décrites dans le chapitre précédent montrent différents cheminements interdomaines dont les séquences d'AS traversés dépendent des règles définies dans les politiques de routage des acteurs de l'Internet. En supposant qu'un unique accord d'interconnexion de type peering ou client/fournisseur est négocié entre deux AS et que les règles économiques qui découlent de ces accords sont correctement implémentées dans les politiques de routage, on peut en déduire une notion de validité économique pour les chemins d'AS [60]. Cette déduction est la clé du problème d'inférence des accords d'interconnexion. Celui-ci peut se définir comme un problème de recherche de labels sur les arcs du graphe des AS telle que la validité économique des cheminements observés soit maximisée [173]. Ce problème d'optimisation a été étudié sous la forme du problème MaxTOR[1]. Il est NP-complet même pour de petites tailles des chemins [15]. La taille des instances réelles du problème (plus de 20 000 AS et plus de 60 000 liens d'interconnexion) rend sa résolution très difficile. Dans la littérature, ce problème a d'abord été résolu avec des heuristiques dans [60, 173]. Sans considérer les accords de peering, il a été traité grâce à une formulation Max2Sat dans [14, 46]. Les solutions obtenues grâce aux algorithmes d'inférence sont difficiles à valider [156]. D'une part les accords d'interconnexion sont des informations propres à chaque opérateur et d'autre part la mesure du graphe logique des AS par le protocole BGP est

[1] Maximum Type Of Relationship

sensible au nombre et à la diversité des tables de routages utilisées en entrée [143]. La formulation du problème d'inférence des accords d'interconnexion en problème MaxTOR présente deux désavantages majeurs : elle n'est pas toujours juste vis à vis de la nature des mesures [190, 191] et elle est fortement sous-déterminée[2]. C'est en raison des limites de sa modélisation que l'on a du mettre en oeuvre une stratégie pointue de filtrage des données d'entrées décrite dans le chapitre précédent. Dans la littérature, la fonction objectif du problème a été modifiée dans [40] pour tenir compte du degré de chaque AS par rapport à ses AS clients potentiels. Par ailleurs, parmi toutes les solutions avec une même valeur optimale pour l'objectif, [15] propose de choisir les solutions avec le plus de relations de peering possible. Enfin dans [41], les auteurs proposent une résolution du problème en utilisant d'abord une relaxation SDP[3] appliquée sans les accords de peering et avec la fonction objectif modifiée, puis en utilisant une heuristique pour augmenter le nombre de relations de peering. Après validations de leurs solutions auprès des responsables de plus de 30 AS, ils ont montré qu'un nombre majoritaire des accords inférés étaient justes mais que les accords d'interconnexion entre les AS de grande taille sont difficiles à déterminer. Ces accords correspondent à un faible pourcentage du nombre total d'accords.

Dans ce chapitre, on propose de traiter le problème d'inférence des accords d'interconnexion entre AS en apportant de nouvelles contributions sur les points suivants :

- *Un filtrage pointu des données d'entrée* : dans le chapitre 2, on a montré comment obtenir un ensemble de chemins d'AS qui contient a priori beaucoup moins de routages temporaires que ceux directement extraits à partir de tables routages. On montre aussi dans le paragraphe 3.1.3 comment filtrer certains sous-chemins potentiellement invalides en exploitant une classification hiérarchiques des AS.
- *Une modélisation plus pertinente* : on propose de résoudre le problème MaxTOR construit à partir des sous-chemins de taille 3. Cette idée est l'extension naturelle des formulations 2-SAT proposées dans la littérature [14, 46] en tenant compte des accords de peering. On propose également de nouvelles contraintes permettant d'améliorer la qualité des solutions.
- *Une formulation CSP et sa résolution avec une méta-heuristique* [129] : dans 3.2.1, une formulation basée sur le modèle de satisfaction de contraintes est proposée puis résolue en 3.2.2 au moyen d'une méthode de recherche locale avec un mécanisme de recherche tabou. Cette méthode heuristique est ensuite raffinée grâce à une exploration guidée

[2]Cela a pour conséquence une dégénérescence des solutions optimales ou quasi-optimales possibles.
[3]Semi-Definite Programming

des solutions dans 3.2.3 et grâce à la détection des cycles de clients dans les solutions.

– *Une généralisation du problème d'optimisation* : le problème MaxTOR se généralise en un problème d'optimisation MaxVCAP. Dans ce nouveau problème, on considère un nombre arbitraire de labels, et on fixe des contraintes de validité entre labels successifs. Le problème MaxVCAP est très général et de nombreux problèmes peuvent s'écrire sous cette forme. Le problème MaxVCAP s'écrit de façon simple sous la forme d'un programme non linéaire en nombre entiers. Si on suppose qu'il n'y a que des chemins de taille 3 (cas particulier appelé MaxVCAP3), alors le problème s'écrit simplement sous la forme d'un programme quadratique en nombre entiers.
– *Formulation linéaire pour MaxVCAP* : on propose une méthode systématique pour obtenir une formulation du problème MaxVCAP en programme linéaire en nombre entiers (MIP). Cette méthode exploite en particulier l'algorithme détaillé en annexe C de la thèse qui permet le calcul des équations des facettes de polyèdres 0-1 de petite taille. Pour résoudre un problème MaxVCAP, il suffit donc de calculer sa formulation linéaire grâce à notre méthode et d'utiliser un algorithme de "Branch & Cut" pour le résoudre. En effet, la formulation linéaire en nombres entiers obtenue est plus facile à résoudre que la formulation non linéaire même si la taille du premier est plus importante.
– *Formulation linéaire compacte pour MaxTOR* : pour le problème MaxTOR (et la version MaxTOR3 avec des sous-chemins de taille 3), cas particulier de MaxVCAP, on propose plusieurs améliorations de la formulation MIP en supprimant la condition d'intégrité de certaines variables et en supprimant les contraintes inutiles.
– *Résolution exacte de MaxTOR3* [130] : tout d'abord on propose d'utiliser une méthode de réduction (R_{stub}) pour obtenir une formulation plus petite (on divise par 10 le nombre de variables de labels et on divise par 20 le nombre de sous-chemins de taille 3 exploités dans la formulation). Cette formulation plus petite permet de trouver le nombre optimum de chemins valides plus rapidement. On propose des coupes dites "gadget" du polyèdre des solutions du problème MaxTOR3 qui permettent de diviser par 100 le nombre de noeuds explorés dans l'algorithme de "Branch & Cut" du logiciel CPLEX.

La résolution exacte du problème MaxTOR3 jusqu'alors impossible à notre connaissance, devient possible grâce à notre proposition de formulation linéaire de MaxVCAP améliorée pour MaxTOR3 et grâce à notre réduction (R_{stub}) et à nos coupes "gadget". Nos différentes contributions permettent non seulement de résoudre le problème à l'optimum mais aussi d'améliorer la qualité potentielle des solutions grâce à de nouvelles contraintes. En fin de chapitre, on évalue la performance de nos méthodes de résolution. La validation des

solutions du problème d'inférence des accords d'interconnexion est impossible en pratique. On confronte donc nos résultats avec les informations connues en provenance des réseaux de France Télécom et avec des solutions calculées par plusieurs heuristiques connues de la littérature.

3.1 L'inférence des accords d'interconnexion

Les accords d'interconnexion bilatéraux établis entre les différents AS de l'Internet ne sont pas publics. Les fournisseurs d'accès Internet ont leur latitude pour établir et définir leurs modes de tarification. Le caractère transit ou peering de l'accord d'interconnexion, qui stipule en quelque sorte les différentes destinations accessibles[4] via la liaison d'interconnexion, est négocié en fonction de l'importance de chaque opérateur. Le problème d'inférence des accords d'interconnexion cherche à déterminer le caractère peering ou transit des contrats économiques établis entre les AS de l'Internet. Grâce aux chemins d'AS obtenus par tomographie BGP, on peut analyser des routages sous contraintes économiques valides. La validité des routages est exploitée pour reconstituer la structure économique initiale du réseau.

3.1.1 Les chemins économiquement valides

Pour définir le problème d'inférence des accords d'interconnexion entre AS, on introduit les notations et hypothèses dans a) et b). Puis on définit les contraintes économiques de routage en c) et enfin on montre comment inférer les différents AS appartenant à une même entreprise dans d) avant de présenter les formulations du problème dans 3.1.2.

a) Hypothèses et notations

Les accords d'interconnexion conclus deux à deux entre AS sont la plupart du temps de type transit (client/fournisseur) ou de type peering. En toute rigueur, il existe des accords d'interconnexion négociés pour accéder seulement à certains préfixes ou certains AS. Par exemple, il arrive que deux opérateurs n'établissent un accord de peering que pour les clients d'un continent ou d'un unique pays. Les règles dans les politiques de routage BGP sont alors spécifiques pour différents préfixes exportés par le même AS voisin. Même si

[4]On parle ici d'accessibilité au sens routage IP. Cela correspond aux différents réseaux NLRI annoncés dans les sessions eBGP entre les routeurs.

Label	Label symétrique
C2P	P2C
P2C	C2P
PEER	PEER
SIB	SIB

TAB. 3.1 – Symétrie des relations économiques.

le nombre de ces accords particuliers tend à augmenter aujourd'hui, ils restent encore très minoritaires par rapport aux accords de peering et aux accords clients-fournisseurs. Pour étudier le problème d'inférence des accords d'interconnexion, on fera l'hypothèse suivante :

Hypothèse 14. *Pour chaque paire d'AS interconnectés, l'accord d'interconnexion est unique. Il correspond aux deux relations économiques symétriques l'une de l'autre. Les types de relations possible sont les suivants :*

C2P *désigne une relation transit d'un AS client vers un autre AS,*

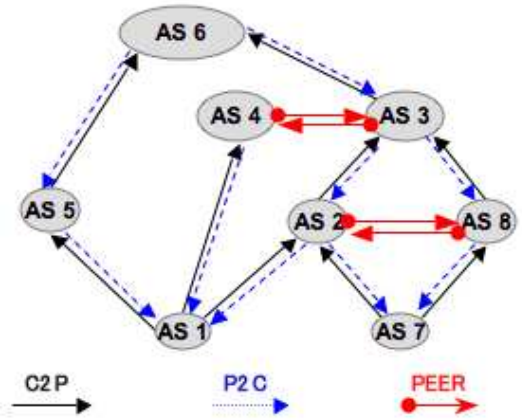
P2C *désigne une relation transit de fournisseur vers client,*

PEER *désigne une relation de peering entre deux AS,*

SIB *désigne une relation qui indique que deux AS appartiennent à une même entité administrative.*

On désignera par $\mathbb{A} = (C2P, P2C, PEER, SIB)$ l'ensemble des relations économiques possible entre 2 AS. Ces relations seront aussi appelées labels. A chaque label de l'ensemble $\mathbb{A}$, correspond en symétrie un label conformément au tableau 3.1. On notera $S(l)$ le label "symétrique" du label $l \in \mathbb{A}$.

– Remarque : deux opérateurs peuvent être connectés par le biais de plusieurs liaisons pour distribuer le trafic ou pour assurer la redondance des routages en cas panne. Dans ce cas l'hypothèse d'unicité des accords reste correcte. Il est pourtant possible que deux opérateurs décident de mettre en place différents accords en fonction de la localité géographique et des destinations accessibles d'une liaison. De telles politiques de routage ne seront pas considérées car elle sont définies à une échelle plus fine que celle notre modèle des relations uniques.



(a) Chaque arc porte un label qui correspond à l'accord économique du noeud source vers le noeud destination.

FIG. 3.1 – Exemple de graphe inter-AS

b) Modèle de graphe inter-AS

Considérons un graphe $G = (V, E)$ représentant le graphe orienté symétrique[5] des liaisons entre AS. L'ensemble $V = \{u_1, u_2, \ldots, u_n\}$ désigne les n AS publics du graphe. L'ensemble $E = (e_1, e_2, \ldots, e_{2p})$ désigne l'ensemble des liaisons dirigées entre les AS. A chaque interconnexion BGP entre deux AS u_i et u_j (avec $i \neq j$), on associe deux arcs symétriques $e_{ij} = \overrightarrow{(u_i, u_j)}$ et $e_{ji} = \overrightarrow{(u_j, u_i)}$ de l'ensemble E. Chaque paire d'arc symétrique correspond à un ensemble de sessions eBGP établies entre les routeurs des deux AS. A chaque arc du graphe G, on associe une unique relation économique de l'ensemble $\mathbb{A}$, conformément à l'hypothèse 14. On notera $L(e) = L_{i,j}$ la relation économique portée par un arc $e = \overrightarrow{(u_i, u_j)} \in E$. Le graphe G est tel que les relations économiques des arcs symétriques sont symétriques (exemple figure 3.1). On notera $deg(u)$ le nombre de noeuds adjacents à chaque AS u (égal au nombre d'arcs entrants et au nombre d'arcs sortants du noeud u).

c) Les chemins économiquement valides

Les contraintes de cheminement entre AS sont déduites des accords d'interconnexion. Ces contraintes sont traduites par des règles de propagation de messages BGP dans les politiques de routage. Pour étudier le problème d'inférence des accords d'interconnexion, on supposera que les règles suivantes sont universelles pour tous les AS :

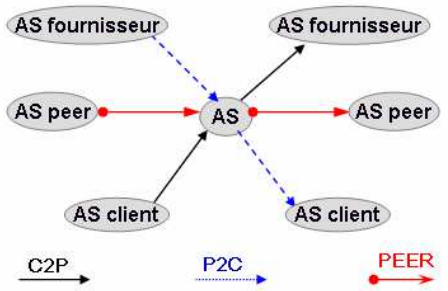
[5]Un graphe est symétrique lorsque pour chaque arc d'un noeud vers un autre il existe aussi un arc dans le sens inverse.

Hypothèse 15. *Les règles de propagation des messages BGP dans les politiques de routage incluent les règles suivantes :*

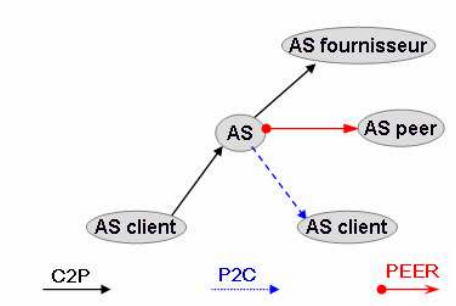
- *un message en provenance d'une liaison eBGP de peering (arc PEER) ne sera jamais redistribué par une liaison eBGP vers l'AS d'un partenaire de peering (arc PEER) ou d'un fournisseur (arc C2P),*
- *réciproquement, un message en provenance d'une liaison eBGP avec un fournisseur (arc P2C) ne sera jamais redistribué par une liaison eBGP vers l'AS d'un partenaire de peering (arc PEER) ou d'un fournisseur (arc C2P) .*

Ces hypothèses sont résumées dans la figure 3.2.

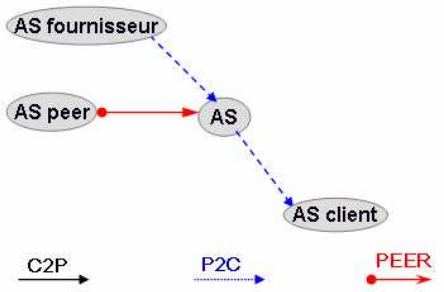
	C2P	PEER	P2C
C2P	1	1	1
PEER	0	0	1
P2C	0	0	1



(a) La matrice représente les successions de labels valides correspondant à des chemins de taille 3 passant par l'AS du mililieu de la figure de droite. L'élément du tableau est un 1 si la propagation de messages BGP est économiquement valide pour le chemin de taille 3 entre deux AS voisins de l'AS du milieu de la figure, et 0 sinon.



(b) Propagation des messages BGP en provenance de clients.



(c) Propagation des messages BGP en provenance de fournisseurs ou partenaires de peering.

FIG. 3.2 – Contraintes économiques pour la propagation des messages BGP entre deux AS

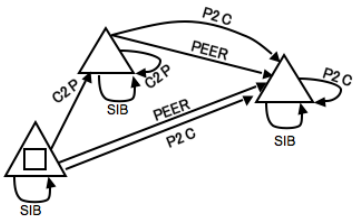


FIG. 3.3 – Automate de construction des labels de chemins valides

Lemme 16. *AS_PATH sans vallées*
Sous réserve des deux hypothèses 14 et 15, et si on néglige l'existence des accords de sibling (SIB), les attributs AS_PATH des messages BGP sont tels que les relations économiques qui se succèdent dans AS_PATH respectent l'expression régulière suivante :

$$C2P* \; PEER? \; P2C* \rightarrow \begin{cases} C2P - \ldots - C2P - PEER - P2C - \ldots - P2C \\ C2P - \ldots - C2P - P2C - \ldots - P2C \\ C2P - \ldots - C2P - PEER \\ C2P - \ldots - C2P \\ PEER - P2C - \ldots - P2C \\ PEER \\ P2C - \ldots - P2C \end{cases}$$

Preuve :
Considérons que l'ensemble $\mathbb{A}$ constitue un alphabet de symboles. La suite de labels des arcs d'un chemin élémentaire du graphe G constitue un mot d'un langage L construit à partir de l'alphabet $\mathbb{A}$. Le tableau 3.2 indique les différentes règles de construction du langage L. Il est constitué de mots correspondant à des séquences de labels d'arcs. L est le langage des labels des chemins élémentaires économiquement valides du graphe G. Considérons l'automate permettant la construction de ces mots figure 3.3. Puisque l'automate est fini, le théorème de Kleene assure que le langage est régulier. Finalement grâce à cet automate on peut vérifier par construction que les mots du langage respectent l'expression régulière $(C2P)* \; PEER? \; (P2C)*$.

□

Définition 17. *Chemin et triplet économiquement valide*
Soit un chemin élémentaire p du graphe G. Le chemin d'AS $p = (u_1, \ldots, u_n)$ *est économiquement valide si et seulement si p correspond à une suite de labels qui vérifie l'expression :* $(C2P)* \; PEER? \; (P2C)*$.

Lorsque p est valide, on dira que la paire de labels $\left(L(\overrightarrow{u_i, u_{i+1}}), L(\overrightarrow{u_{i+1}, u_{i+2}})\right)$ *est valide pour tout i,* $1 \leq i \leq n-2$.
Désormais, soit une paire d'arcs (e_{ij}, e_{jk}) *consécutifs dans un chemin, avec* $e_{ij} = \overrightarrow{(u_i, u_j)}$ *et* $e_{jk} = \overrightarrow{(u_j, u_k)}$. *Alors :*
la paire de labels $\left(L(\overrightarrow{u_i, u_j}), L(\overrightarrow{u_j, u_k})\right)$ *est valide si et seulement si le triplet* (u_i, u_j, u_k) *est valide*
si et seulement si $(L_{i,j}, L_{j,k}) \in \left\{ \begin{array}{c} (C2P, C2P), (C2P, PEER), (C2P, P2C), \\ (PEER, P2C), \\ (P2C, P2C) \end{array} \right\}$ *et si et seulement si* $L_{i,j} = C2P$ *ou* $L_{j,k} = P2C$.

On néglige l'existence des accords de sibling dans notre modèle de contrainte (cas particulier traité en d)). Sous nos hypothèses, un message BGP se propagera typiquement dans un premier ensemble d'AS par des liaisons de client à fournisseur, puis le message peut éventuellement traverser un lien de peering entre deux AS, enfin le message BGP est transmis dans un second ensemble d'AS (strictement distinct du premier) au travers de liaisons fournisseur vers client.

d) Cas des siblings

Les relations de sibling sont établies entre les AS appartenant à la même administration, mais pas systématiquement. Cette relation économique correspond à l'absence de contrainte sur la propagation des annonces BGP échangées entre les deux AS. Elle relâche en fait les contraintes sur les relations économiques entre AS voisins. En d'autres termes, si un triplet d'AS (X, Y, Z) met en jeu 2 AS X, Y (ou Y, Z) en relation de sibling, alors ce triplet est toujours valide. Plus précisément, si un chemin (A, X, Y, B) est observé alors que X et Y sont en relation de sibling, alors on peut conjecturer que la succession des deux relations économiques portées par les arcs $\overrightarrow{(A, X)}$ et $\overrightarrow{(Y, B)}$ est valide. Il est nécessaire de connaître les relations de sibling pour en déduire un jeu de contraintes prenant en compte la remarque précédente. Le cas échéant, on peut choisir d'appliquer l'une des deux procédures suivantes :

Procédure 1 : on déduit de la contrainte de succession pour les labels des arcs $\overrightarrow{(A, X)}$ et $\overrightarrow{(Y, B)}$ d'après tous les sous-AS_PATH (A, X, Y, B) où X et Y sont en relation de sibling.

Procédure 2 : suppression des triplets (X, Y, Z) mettant en jeu une relation de sibling entre X et Y ou entre Y et Z, puisqu'aucune contrainte de succession directe

ne peut être déduite.

La seconde possibilité a été retenue et les relations sibling sont déterminées avec l'algorithme dans 2.4.1. Une fois les relations de sibling inférées, on restreint l'ensemble des labels possibles à $\mathbb{L} = \mathbb{A}\backslash\{SIB\} = \{C2P, P2C, PEER\}$. Ne plus considérer les accords de sibling permet d'obtenir un problème de maximisation avec des contraintes binaires qui font intervenir seulement des arcs adjacents dans les chemins.

3.1.2 Formulations du problème

a) Formulation à base de chemins d'AS

La formulation originale du problème d'inférence des accords d'interconnexion entre AS se définit en maximisant la validité d'un ensemble de chemins d'AS d'une tomographie. Ce problème proposé initialement dans [60] a été formulé en [173] sous le nom MaxTOR. Soient un graphe symétrique G et un ensemble de chemins élémentaires P construit dans G.

MaxTOR : affecter un label $PEER$, $P2C$ ou $C2P$ à chaque arc du graphe pour maximiser le nombre de chemins de P vérifiant l'expression régulière :

$$C2P* \, PEER? \, P2C* \rightarrow \begin{cases} C2P - \ldots - C2P - PEER - P2C - \ldots - P2C \\ C2P - \ldots - C2P - P2C - \ldots - P2C \\ C2P - \ldots - C2P - PEER \\ C2P - \ldots - C2P \\ PEER - P2C - \ldots - P2C \\ PEER \\ P2C - \ldots - P2C \end{cases}$$

Le problème MaxTOR peut se voir comme un problème d'orientation partielle de graphe[6] (voir figure 3.4). Si on ne considère que les accords $P2C$ et $C2P$, le problème devient celui d'une orientation de graphe [14, 46] :

MaxTOR-simple : affecter un label $P2C$ ou $C2P$ à chaque arc du graphe tel que le nombre de chemins de P vérifiant l'expression régulière $C2P * P2C*$ soit maximum.

[6] L'orientation du graphe serait donnée par les accords C2P/P2C en orientant par exemple les liens des clients aux fournisseurs. Les arcs PEER correspondraient à des liens non orientés.

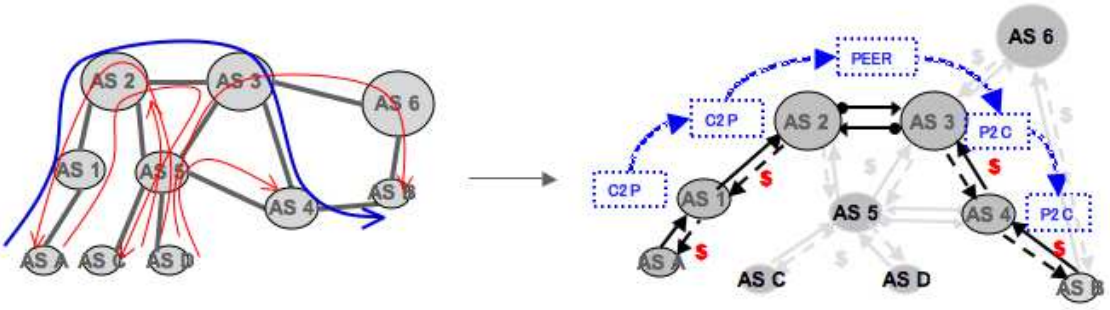


FIG. 3.4 – Exemple de problème d'inférence des accords d'interconnexion

– Remarque : aux problèmes de maximisation MaxTOR et MaxTOR-simple, on peut associer les problèmes suivants de décision :
  – l'ensemble P admet-il une orientation partielle telle que tous les chemins issus de P vérifient l'expression régulière $C2P* \; PEER? \; P2C*$ ou $C2P* \; P2C*$? Alors que les problèmes de décision peuvent être résolus en temps polynomial [14], les problèmes MaxTOR et MaxTOR-simple sont NP-difficiles[7] [15].

b) Formulation à base de triplets d'AS

Les problèmes de décision peuvent être ramenés à un problème SAT en exprimant la validité d'un chemin en fonction de la validité des paires d'arcs successives dans ce même chemin. En construisant un ensemble de clause binaires pour les labels des arcs adjacents dans les chemins, le problème de décision sans accord de peering devient un problème 2-SAT, décidable en temps polynomial [158]. Cette formulation 2-SAT peut être utilisé pour formuler les problèmes MaxTOR-simple et MaxTOR en utilisant les chemins de taille 3 en entrée au lieu de P.
Soient un graphe symétrique G et un ensemble de chemins élémentaires P du graphe G. On note P_3 l'ensemble des sous-chemins de taille 3 déduits de P. Les éléments de P_3 seront appelés *triplets d'AS :*

$$P_3 = \{(p_{i-1}, p_i, p_{i+1}) \,/\, \exists p = (p_1, \ldots, p_k) \in P,\, k \geq 3,\, 2 \leq i \leq k-1\}$$

Un triplet d'AS constitue un chemin valide si et seulement si le triplet d'AS symétrique constitue un chemin valide. Pour tout triplet d'AS distincts (u_i, u_j, u_k) de l'ensemble T,

[7]En effet, il est possible de réduire le problème du stable maximum au problème MaxTOR. De plus la difficulté du problème MaxTOR ne dépend pas de la taille des chemins de P.

(u_i, u_j, u_k) est valide *si et seulement si* (u_k, u_j, u_i) est valide. On introduit donc l'ensemble P_3' suivant :

$$P_3' = \left\{ p_3 = (u_i, u_j, u_k) \,/\, \begin{array}{c} p_3 \in P_3 \text{ si } i < k \\ (u_k, u_j, u_i) \in P_3 \text{ sinon.} \end{array} \right\}$$

Par construction, l'ensemble P_3' vérifie :

$$\forall u_i, u_j, u_k \in V, \left\{ \begin{array}{l} (u_i, u_j, u_k) \in P_3 \\ \text{ et} \\ (u_k, u_j, u_i) \in P_3 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} (u_i, u_j, u_k) \in P_3' \\ \text{ ou exclusif} \\ (u_k, u_j, u_i) \in P_3' \end{array} \right\}$$

L'ensemble P_3' permet d'avoir une bijection entre les différentes contraintes de successivité et les éléments de P_3'. On note également T l'ensemble des arcs adjacents issus des chemins de P_3' :

$$T = \bigcup_{(p_i,p_jp_k) \in P_3'} \left(\overrightarrow{(p_i, p_j)}, \overrightarrow{(p_j, p_k)} \right)$$

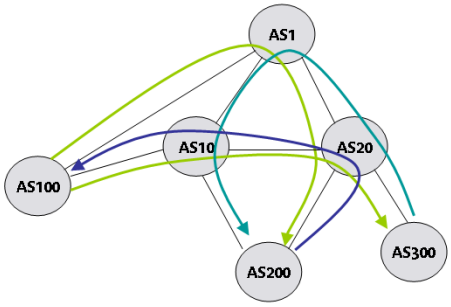
Les formulations étudiées dans [14, 40, 46] ne font pas intervenir les accords de peering. Le problème MaxTOR3-simple alors obtenu est un problème Max-2-SAT dans lequel il faut maximiser la validité d'un ensemble de clauses binaires. Chaque clause est définie sur deux variables de labels. Les variables sont binaires car chaque arc porte le label $P2C$ ou $C2P$. Les contraintes 2-SAT utilisées correspondent à l'utilisation de l'ensemble $P_3^{'}$ en entrée :

MaxTOR3-simple : affecter un label $P2C$ ou $C2P$ à chaque arc du graphe tel que le nombre de chemins de P_3' vérifiant l'expression régulière $C2P * P2C*$ soit maximum.

Le problème Max-2-SAT obtenu est un problème NP-difficile [111] (voir aussi les travaux dans [141, 142]). On propose l'extension suivante pour étudier l'inférence des accords d'interconnexion :

MaxTOR3 : affecter un label $PEER$, $P2C$ ou $C2P$ à chaque arc du graphe tel que le nombre de chemins de P_3' vérifiant l'expression régulière $C2P* \; PEER? \; P2C*$ soit maximum.

Un exemple de formulation MaxTOR3 est donné figure 3.5. Le problème MaxTOR3, peut être formulé grâce au modèle CSP de satisfaction de contraintes. Cette formulation,



(a) Une topologie interdomaine avec 4 AS_PATH

AS collecteur	AS_PATH (BGP)	P_3	P'_3
100	1 20 200	100 1 20 1 20 200	20 1 100 1 20 200
100	10 20 300	100 10 20 10 20 300	100 10 20 10 20 300
300	20 1 10 200	300 20 1 20 1 10 1 10 200	1 20 300 10 1 20 1 10 200
200	20 10 100	200 20 10 20 10 100	10 20 200 20 10 100

(b) Il y a 4 AS_PATH mesurés qui sont décomposés en 9 triplets d'AS. Ces chemins d'AS ont été collectés par 3 AS différents (100,200,300)

Variables : $\{L_{1,10},\ L_{1,20},\ L_{1,100},\ L_{10,20},\ L_{10,100},\ L_{10,200},\ L_{20,200},\ L_{20,300}\}$

$$(D_{dd}) \quad \left\{ \begin{array}{l} (L_{1,20}, L_{20,200}), \\ (L_{10,20}, L_{20,300}), \\ (L_{1,20}, L_{20,300}), \\ (L_{1,10}, L_{10,200}) \\ (L_{10,20}, L_{20,200}) \end{array} \right\} \in \left\{ \begin{array}{l} (C2P, C2P), (P2C, P2C), \\ (C2P, PEER), (PEER, P2C), \\ (C2P, P2C). \end{array} \right\}$$

$$(D_{di}) \quad \left\{ \begin{array}{l} (L_{1,100}, L_{1,20}), \\ (L_{10,100}, L_{10,20}), \\ (L_{1,20}, L_{1,10}), \\ (L_{10,20}, L_{10,100}) \end{array} \right\} \in \left\{ \begin{array}{l} (P2C, C2P), (C2P, P2C), \\ (P2C, PEER), (PEER, P2C), \\ (P2C, P2C) \end{array} \right\}$$

$$(D_{id}) \quad \left\{ \begin{array}{l} (L_{1,20}, L_{1,100}), \\ (L_{10,100}, L_{10,20}) \\ (L_{1,10}, L_{1,20}) \\ (L_{10,20}, L_{10,100}) \end{array} \right\} \in \left\{ \begin{array}{l} (P2C, P2C), (C2P, C2P), \\ (P2C, PEER), (PEER, C2P), \\ (P2C, C2P) \end{array} \right\}$$

(c) Variables et contraintes du modèle CSP

FIG. 3.5 – Exemple de problème MaxTOR3 pour l'inférence des accords d'interconnexion

d'abord proposée dans [127, 129], est reportée dans ce document en 3.2.

– Remarque : le problème MaxCSP est NP-difficile [158] dans le cas général. Les auteurs de [15] ont montré que MaxTOR3 et MaxTOR sont NP-difficiles et non approximables en temps polynomial à moins d'une constante près, même lorsque la taille des chemins en entrée est petite.

c) Dégénérescence des solutions

On peut remarquer que chaque solution du problème MaxTOR (ou MaxTOR3) qui comporte $2 \times n$ arcs de peering[8], peut être transformée en 3^n solutions satisfaisant le même nombre de chemins valides avec l'algorithme 2 (les éléments de preuve sont reportés dans le lemme 18). Parmi ces 3^n solutions, 2^n sont sans arcs de peering. Le nombre de solutions équivalentes à une solution donnée varie donc exponentiellement en fonction du nombre de liens de peering. L'algorithme suivant permet de générer des solutions équivalentes :

Algorithme 2. *Transformation des solutions MaxTOR*

Soit L une solution du problème MaxTOR.
Pour chaque couple d'arcs symétriques e_{ij}, e_{ji} tels que $L(e_{ij}) = PEER = L(e_{ji})$,
Transformer le label de l'arc e_{ij} librement en $P2C$ ou $C2P$,
ou garder la valeur $PEER$ (l'arc e_{ji} porte le label symétrique de l'arc e_{ij})
fin pour chaque

fin de l'algorithme

d) Maximisation du nombre de liens de peering d'une solution

Lemme 18. *Soit P un ensemble de chemins et un entier $n \leq |P|$, alors :*

Il existe une solution réalisable de MaxTOR avec au moins n chemins valides si et seulement si il existe une solution réalisable de MaxTOR-simple avec au moins n chemins valides.

Preuve :

sens direct : soit une solution x^* du problème MaxTOR dont au moins n chemins de P sont valides. Si x^* ne comporte pas de labels $PEER$, alors x^* est aussi

[8] Cela correspond à n accords bilatéraux de peering.

solution de MaxTOR-simple avec au moins n chemins valides. Sinon, il est possible de modifier chaque accord de peering de x^* sans violer de nouveaux chemins comme le montre les points suivants :

– En effet, soit un arc a portant l'accord $PEER$ dans x^*. Soit a' l'arc symétrique de a . Soient $p_1, \ldots, p_q$ les q chemins valides passant par l'arc a ou l'arc a'. De ces chemins on peut extraire les doublets d'arcs consécutifs passant par a ou a'.

$$T_a = \bigcup_{\substack{(u_i,u_j,u_k) \subset p \in \{p_1,\ldots,p_q\} \\ t.q. \begin{cases} a = \overrightarrow{(u_i,u_j)} ou\, a = \overrightarrow{(u_j,u_k)} \\ ou\, a = \overrightarrow{(u_j,u_i)} ou\, a = \overrightarrow{(u_k,u_j)} \end{cases}}} (u_i,u_j,u_k)$$

– Les doublets d'arcs de T_a sont valides car les chemins $\{p_1, \ldots, p_q\}$ sont valides. On peut donc écrire :

$$\exists (b,a) \subset p_i \, ou \, \exists (b,a') \subset p_i \quad \Rightarrow \quad L(b) = C2P$$

$$\exists (a,b) \subset p_i \, ou \, \exists (a',b) \subset p_i \quad \Rightarrow \quad L(b) = P2C$$

Les valeurs des labels des arcs b (contigus à a ou à a') assurent que tous les doublets soient valides quel que soit le label de l'arc a. On peut alors modifier la valeur de a en $C2P$ ou $P2C$ sans ajouter de nouveaux chemins invalides.

– De plus, on remarque que l'ensemble T_a des doublets d'arcs valides contenant un arc de peering a et son symétrique a', et l'ensemble T_b des doublets d'arcs valides contenant un autre arc de peering b et son symétrique b' sont disjoints car un chemin valide ne peut contenir qu'un seul arc de peering. Donc la procédure peut être appliquée arc de peering par arc de peering. La solution obtenue en modifiant x^* ne comporte plus d'accord de peering et devient une solution de MaxTOR-simple. Cette solution du problème MaxTOR-simple valide au moins n chemins de P.

sens réciproque : soit une solution optimale y^* du problème MaxTOR-simple avec au moins n chemins valides. Cette solution est solution de MaxTOR avec le même nombre de chemins valides. □

On peut en déduire que les solutions optimales du problème MaxTOR et du problème MaxTOR-simple comportent le même nombre de chemins valides. Ainsi pour résoudre le problème MaxTOR, on peut d'abord résoudre le problème MaxTOR-simple. On obtient une solution sans arc de peering. Puis, en partant de cette solution, on peut en chercher

une autre qui maximise le nombre de liens de peering sans augmenter le nombre de chemins invalides.

PEERING-DISCOVERY : soient un ensemble de chemins (éventuellement des chemins de taille 3 seulement) et une solution du problème MaxTOR (ou MaxTOR3). Maximiser le nombre de liens de peering dans la solution avec au moins le même nombre de chemins valides que dans la solution initiale.

Ce nouveau problème PEERING-DISCOVERY est aussi NP-difficile [15]. La difficulté réside dans la combinatoire du choix des arcs en relation de peering. Les problèmes MaxTOR et MaxTOR3 ont beaucoup de solutions conduisant au même nombre de chemins valides. Cette dégénérescence est brisée si on s'intéresse aux solutions lexicographiquement optimales[9] pour MaxTOR (ou MaxTOR3) puis pour PEERING-DISCOVERY.
On a donc montré qu'il était possible de modéliser le problème d'inférence des accords d'interconnexion avec ou sans accords de peering grâce à différents problèmes. Dans cette thèse, on s'intéressera au problème MaxTOR3 plus particulièrement. C'est le cas avec arcs de peering et sous-chemins de taille 3.

3.1.3 Amélioration de la qualité des solutions

Pour traiter le problème d'inférence des accords, on propose d'utiliser le modèle MaxTOR3 avec de nouvelles contraintes[10]. Les solutions maximisant le nombre d'arcs de peering ne sont pas forcément les meilleures solutions. On exploitera des propriétés économiques simples du réseau inter-AS afin de ne retenir que les solutions les plus probables. Les nouvelles contraintes ajoutées font l'objet de notre contribution dans la modélisation du problème.

a) Modèle MaxTOR3 pour l'inférence des accords d'interconnexion

L'inférence des accords d'interconnexion utilise les tables de routage BGP publiques comme données d'entrée. On obtient entre 2 millions et 8 millions de routes distinctes à partir des attributs AS_PATH extraits des tables BGP. Le problème d'optimisation MaxTOR est d'autant plus difficile à résoudre que le nombre de chemins augmente. Pour

[9] On parle d'optimalité lexicographique pour plusieurs critères lorsqu'on cherche des solutions d'abord optimales pour le premier critère, puis optimales pour le second critère...

[10] Le nombre exponentiel de solutions équivalentes vu en c) pour MaxTOR ou MaxTOR3 montre qu'il est nécessaire de définir d'autres familles de contraintes pour décider d'une solution parmi les solutions optimales.

réduire cette difficulté due à la taille des instances, on décide de traiter le problème d'inférence à l'aide du problème MaxTOR3. On cherche à maximiser le nombre de triplets d'AS de l'ensemble P'_3, au lieu de maximiser le nombre de chemins valides. Les différents chemins d'AS présents dans une tomographie dépendent fortement du placement des routeurs sondes. En utilisant les triplets d'AS on perd la vision des cheminements de bout en bout, mais on garde l'information essentielle permettant l'obtention des contraintes entre les relations économiques. La formulation MaxTOR3 semble donc plus naturelle que la formulation MaxTOR dans le sens où le biais de mesure induit par le placement des sondes BGP est moins important.

b) Filtrage des données d'entrée

Des filtres pour les chemins utilisés dans l'inférence des accords d'interconnexion sont utilisés dans la littérature [41,190,191]. On propose une détection plus complexe des chemins à supprimer grâce aux différents filtres mis en oeuvre pour construire les tomographies dans le chapitre 2. Le filtrage des routages BGP temporaires permet d'éliminer les politiques d'annonce temporaires (filtre *aat*) ainsi que les liens d'interconnexion temporaires (cf. filtre *lt*). Il en résulte que seulement les routages stables (grâce au filtre rs) avec des politiques d'annonce et les liens d'interconnexion stables sont retenus et considérés dans la tomographie servant de données d'entrée du problème d'inférence des accords.
La modélisation des relations économiques ne permet pas de prendre en compte certaines contraintes économiques réelles car un unique accord d'interconnexion existe entre deux AS (cf. hypothèse 14). De même nos hypothèses concernant les configurations des politiques de routage (cf. 15) peuvent se révéler fausses dans le cas de l'existence de routes BGP de secours (routes de "backup" [62]). En pratique, des politiques de routages très spécifiques peuvent avoir été configurées par les administrateurs des AS pour certains préfixes seulement. On supposera que ces pratiques restent temporaires et ne correspondent pas à des chemins BGP conservés dans une tomographie après l'application des différents filtres décrit dans le chapitre précédent.

c) Accords avec les AS de périphérie

Restriction des liens avec les AS stub

Un AS stub est par définition un AS qui ne fait pas de transit entre deux autres. Autrement dit, un AS stub ne propage jamais de messages BGP en provenance d'un AS vers un autre AS. Les AS stub sont des AS qui n'apparaissent jamais au milieu d'un triplet

d'AS[11]. Il n'est pas d'usage pour les AS sans capacités d'écoulement de vendre l'accès à leurs propres NLRI pour leur contenu. On peut donc interdire les relations $P2C$ d'un AS stub vers un autre AS conformément à la définition 19 et comme illustré sur la figure 3.6.

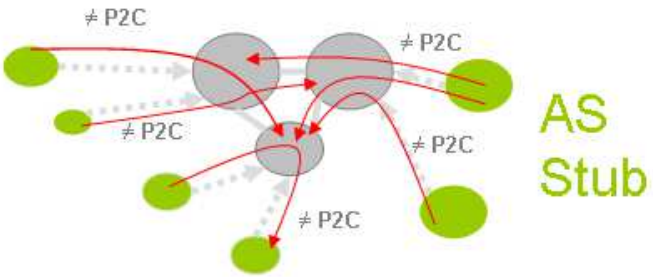


FIG. 3.6 – Illustration des contraintes C_{stub}

Définition 19. *Contraintes* (C_{stub})
On note Stub l'ensemble des AS v tels que pour chacun de ces AS il n'existe pas de triplets dans l'ensemble T de la forme (u, v, w).
Si il existe un arc $e = \overrightarrow{(v, u)}$ *dans l'ensemble E, alors* $L(e) \neq P2C$.
Réciproquement, si il existe un arc $e' = \overrightarrow{(u, v)}$, *alors* $L(e') \neq C2P$.

– Remarque : Lorsqu'on a peu d'attributs AS_PATH ou de peu de routeurs sondes BGP, certains AS peuvent apparaître stub alors qu'ils ne le sont pas.

Certains AS stubs ont pour des raisons qui leur sont propre des politiques de routage BGP économiquement incohérentes. Ils assurent le transit entre leurs fournisseurs pour un certain et faible nombre de plages d'adresses. Pour éviter de perturber nos modèles avec ce cas, on filtrera ces comportements qui semblent a priori invalides.

Restrictions en fonction de la hiérarchie du routage BGP

L'algorithme (R_{stub}) présenté page 77 pour déterminer le *rang stub* de chaque AS, permet d'obtenir plusieurs catégories d'AS stubs. Les AS stubs dont le rang est minimal, sont les AS stub classiques. L'ensemble des AS stubs d'ordre maximal r_{max} est un ensemble contenant le coeur du routage interdomaine. Une solution du problème d'inférence des accords d'interconnexion est cohérente si aucun AS stub d'ordre $n < r_{max}$ n'est fournisseur d'un AS du coeur de l'Internet. Ainsi, après détermination du coeur de l'Internet, on peut ajouter les contraintes (C^*_{stub}) suivantes :

[11]Un AS stub apparaît toujours en fin de chemin ou en début de chemin.

Définition 20. *Contraintes* (C^*_{stub})

(C^*_{stub}) $\quad \forall e = \overrightarrow{(u,v)} \in E,$

$$\begin{cases} R_{stub}(u) < R_{stub}(v) = r_{max} \text{ et } H(v) = 1 & \Rightarrow \quad L(e) \neq P2C \\ R_{stub}(v) < R_{stub}(u) = r_{max} \text{ et } H(u) = 1 & \Rightarrow \quad L(e) \neq C2P \end{cases}$$

En plus de restreindre les valeurs possibles pour les liens entre les AS stub et les AS présumés Tiers-1 (sans fournisseurs), les contraintes C^*_{stub} obligent les contraintes qui font intervenir un AS stub d'ordre n en transit entre deux AS Tiers-1 à être non valides. Ces contraintes permettent donc d'éviter la prise en compte de routages présumés économiquement invalides.

d) Connectivité globale de chaque AS

Certaines politiques de routage sont susceptibles d'être mal observées puisque le nombre de préfixes observés sur les différents liens logiques entre AS ou les différents triplets d'AS est faible (cf. figure 2.11) et puisque le nombre de triplets n'est pas fiable en raison du biais de mesure (cf. 2.2.2). Dans le problème d'inférence des accords d'interconnexion, certaines solutions peuvent indiquer des accords d'interconnexion qui ne permettraient pas à chaque d'AS d'obtenir la connectivité globale de l'Internet (accès à l'ensemble des préfixes IP). Par exemple, les AS avec un seul voisin peuvent apparaître (ces AS sont traités dans les contraintes (C_{stub})) en relation de peering avec ce voisin. Cette relation ne permet donc pas (sauf si l'AS voisin a la connectivité globale de l'Internet ce qui n'existe pas) à l'AS mono-homé d'avoir la connectivité globale : il n'a que la connectivité des clients de son peer.

En pratique, les AS "Tiers-1" sans fournisseurs sont peu nombreux. En effet, certains AS de grande taille utilisent un fournisseur pour joindre quelques NLRI qui ne sont pas joignables par liens de peering ou les liens de leurs clients. De plus pour des raisons diverses et parfois obscures (politiques, stratégiques, historiques ou économiques) certains AS de très grande taille ne sont pas interconnectés entre eux. Pour obtenir la connectivité globale, ces AS peuvent utiliser des relations d'interconnexion mixte : peering pour certains préfixes et transit payant pour d'autres[12]. Dans ces cas limites, le niveau de finesse de notre modèle ne permet pas de traiter ces relations économiques hybrides. Néanmoins

[12]Le service de transit est assuré par un des deux AS sur quelques sessions BGP et pour quelques NRLI seulement. Le reste des sessions BGP correspond à un accord de peering classique.

pour éviter des accords d'interconnexion invraisemblables dans les solutions fournies par nos algorithmes, on définit une contrainte pour que les AS non présumés Tiers-1 aient au moins un fournisseur :

Définition 21. *Contraintes d'accessibilité* (C_{acc})
Pour chaque AS $u \in V$*, si* u *n'est pas un AS Tiers-1* ($h(u) > 1$)*, alors :*

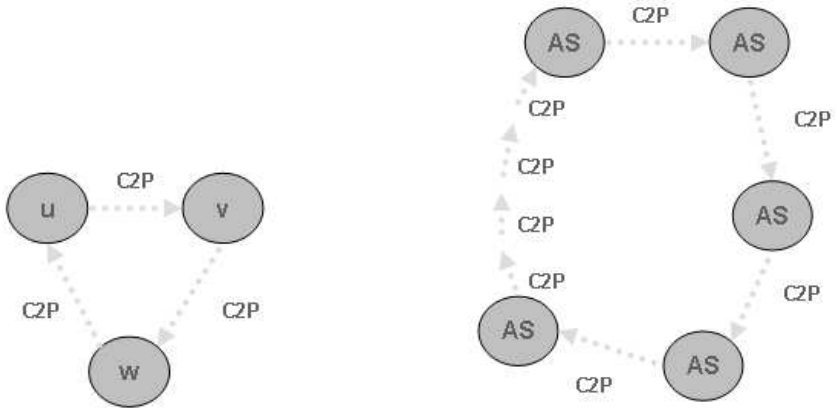
$$(C_{acc}) \qquad \left|\left\{e = \overrightarrow{(u,v)} \in E \,/\, L(e) = C2P\right\}\right| \geq 1$$

e) Cycles d'AS clients

Il est économiquement incohérent qu'un AS u soit client de l'AS v, sachant que l'AS v est lui même client de l'AS w et que l'AS w est client de u. Cette configuration économique reportée figure 3.7(a) correspond à un cycle de taille 3, avec les labels clients/fournisseurs : $\overrightarrow{L(u,v)} = \overrightarrow{L(v,w)} = \overrightarrow{L(w,u)} = C2P$. Plus généralement, on peut supposer que n'importe quelle configuration économique comportant un cycle d'AS clients/fournisseurs est incohérente. On recherchera des solutions du problème d'inférence des accords d'interconnexion qui ne contiennent pas de cycles d'AS avec successivement la même relation économique $C2P$ (le cas $P2C$ est symétrique). Ces nouvelles contraintes appelées (C_{cc}) sont illustrées figure 3.7(b).

Définition 22. *Contraintes* (C_{cc})
Si il existe un cycle d'AS distincts $\gamma = (u_{i_1}, u_{i_2}, \ldots, u_{i_n}, u_{i_1}) \in V^{n+1}$*, alors en notant* $u_{n+1} = u_1$ *:*

$$(C_{cc}) \qquad \exists i \leq n \,/\, \overrightarrow{L(u_{i_j}, u_{i_{j+1}})} \neq C2P$$



(a) Cycle de relations $C2P$ de taille 3. (b) Cycle de relations $C2P$ de taille quelconque.

FIG. 3.7 – Exemples de cycles de clients/fournisseurs incohérents

On a supposé les relations économique être parmi l'ensemble $\{C2P, PEER, P2C\}$, puis on a montré deux formulations basiques MaxTOR et MaxTOR3 pour le problème d'inférence des accords d'interconnexion. Comme expliqué en 3.1.3, on favorise la résolution du problème MaxTOR3 et on ajoute de nouvelles contraintes pour garantir des solutions de meilleure qualité que celles obtenues par les méthodes connues de la littérature. Dans la suite du chapitre, on présente une méthode de résolution heuristique en 3.2 et une méthode de résolution exacte en 3.3. La méthode heuristique permettra de prendre en compte les contraintes (C_{cc}) concernant l'existence de cycles de clients, alors que la méthode de résolution exacte ne le permettra pas en pratique. A l'inverse les contraintes d'accessibilité (C_{acc}) ne seront pas prises en compte dans la méthode heuristique. L'analyse et la validation des solutions calculées par nos algorithmes sont reportées en 3.4.

3.2 Formulation CSP et résolution heuristique

On présente ici une nouvelle formulation pour le problème MaxTOR3 (en 3.2.1) à base du modèle de satisfaction de contraintes CSP. Une fois sous la forme d'un problème de satisfaction maximale de contraintes, on propose un algorithme de recherche locale tabou dans 3.2.2 pour résoudre efficacement MaxTOR3 avec certaines des contraintes additionnelles proposées en 3.1.3. On raffine notre algorithme en 3.2.3 avec des critères heuristiques pour mieux explorer l'espace des solutions.

3.2.1 Modèle CSP

a) Formulation MaxCSP du problème MaxTOR3

Définition 23. *Modèle CSP pour l'inférence des accords d'interconnexion*

Variables : soit $L = \{L_{i,j} = L(\overrightarrow{u_i, u_j})/(\overrightarrow{u_i, u_j}) \in E,\ i < j\}$. *Chaque* $L_{i,j}$ *représente l'accord d'interconnexion entre deux AS* i *et* j *du graphe des AS. En particulier* $L(\overrightarrow{u_j, u_i}) = L_{j,i} = S(L_{i,j})$.

Domaines : soit $D = D_1 = D_2 = \ldots = D_{|L|} = \{C2P, P2C, PEER\} = \mathbb{L}$. *Le domaine* D_i *de chaque variable indique trois valeurs potentielles pour le label* $L(e_i)$ *de l'arc* e_i.

Contraintes : Pour chaque paire d'arcs consécutifs $(e_{ij}, e_{jk}) \in T$, *avec* $e_{ij} = (\overrightarrow{u_i, u_j})$ *et* $e_{jk} = (\overrightarrow{u_j, u_k})$. *Les variables introduites dans le modèle CSP dépendent du signe de* $(j - i)$ *et du signe de* $(k - j)$ *:*

$$
\begin{array}{ll}
(C_{dd}) & L_{i,j} = C2P \text{ ou } L_{j,k} = P2C, \quad si\ i < j \text{ et } j < k \\
(C_{di}) & L_{i,j} = C2P \text{ ou } L_{k,j} = C2P, \quad si\ i < j \text{ et } j > k \\
(C_{id}) & L_{j,i} = P2C \text{ ou } L_{j,k} = P2C, \quad si\ i > j \text{ et } j < k \\
(C_{ii}) & L_{j,i} = P2C \text{ ou } L_{k,j} = C2P, \quad si\ i > j \text{ et } j > k
\end{array}
$$

Dans la formulation CSP proposée, on s'assure que pour chaque arc e du graphe G, le label de l'arc symétrique soit égal au symétrique du label de e. En effet on introduit une seule variable pour chaque paire d'arcs symétriques.
Les contraintes de succession de labels introduites par les triplets d'AS, sont divisées en 4 familles de contraintes en fonction de l'orientation des deux arcs dans chaque triplet d'AS. On donne un exemple figure 3.4. Soit P un ensemble de chemins d'AS du graphe G. Soit P'_3 l'ensemble des triplets d'AS déduits des chemins de P (comme en b)). A chaque triplet d'AS de P'_3 correspond une contrainte parmi les 3 types (C_{dd}, C_{di}, C_{id}). Les contraintes de type (C_{ii}) disparaissent par définition de P'_3.

Définition 24. *Variable de validité d'un triplet d'AS*
Pour chaque chemin p *de l'ensemble* P'_3, *on définit une contrainte* c *parmi les trois types* (C_{dd}, C_{di}, C_{id}). *On définit une variable* $\delta_p \in \{0, 1\}$ *qui prend la valeur* 1 *si la contrainte* c *est satisfaite ou la valeur* 0 *si elle est violée.*

Le problème MaxTOR3 se formule comme un problème MaxCSP comme suit :

b) Formulation MaxCSP du problème MaxTOR3

$$\text{Maximiser} \sum_{p \in P'_3} \delta_p$$

$$\textit{tel que} \quad \forall e = \overrightarrow{(u_i, u_j)} \in E, \quad \begin{array}{ll} \text{si } i < j : & L_{ij} \in \mathbb{L} \text{ est une variable du modèle} \\ \text{si } j < i : & L_{ij} = S(L_{ji}) \text{ est déduit de la variable } L_{ji} \end{array}$$

$\forall p = (u_i, u_j, u_k) \in P'_3$, (on rappelle que $i < k$)

Si $i < j$ et $j < k$: (C_{dd}) $\delta_p = 1 \Leftrightarrow L_{ij} = C2P$ ou $L_{jk} = P2C$

Si $i < j$ et $j > k$: (C_{di}) $\delta_p = 1 \Leftrightarrow L_{ij} = C2P$ ou $L_{jk} = C2P$

Si $i > j$ et $j < k$: (C_{id}) $\delta_p = 1 \Leftrightarrow L_{ij} = P2C$ ou $L_{jk} = P2C$

Si $i > j$ et $j > k$: aucune contrainte

La modélisation du problème d'inférence des accords d'interconnexion en MaxCSP permet l'utilisation de nombreuses méthodes de recherches heuristiques. Les méthodes de recuit simulé [110], les algorithmes de recherche tabou [67], les algorithmes génétiques [12], et plus généralement les algorithmes de résolution du problème MaxCSP [36,37] sont des candidats pour inférer les accords d'interconnexion entre AS. La dégénérescence des solutions limite les performances de certains de ces algorithmes (le recuit simulé par exemple). On a opté pour une recherche tabou pour laquelle on appliquera des restrictions pour l'espace de recherche des solutions réalisables.

3.2.2 Un algorithme de recherche locale tabou

Pour résoudre le problème d'inférence des accords d'interconnexions MaxTOR3, on utilise une méta-heuristique de recherche locale expliquée en a). On exploite un algorithme glouton en b) avec un mécanisme tabou en e). Cet algorithme sera amélioré en 3.2.3.

a) Métaheuristique de recherche locale

Soit un problème de satisfaction de contraintes définit sur un ensemble de variables. A chaque variable correspond un domaine de définition (discret ou continu). Les contraintes du problème sont définies sur les variables et sont valides ou invalides suivant les valeurs des variables. Une solution quelconque du problème est un assignement à chaque variable d'une valeur possible issue de son domaine de définition.

Espace des solutions et voisinage

– Soit S l'**espace des solutions** possibles déterminé d'après les domaines d'admissibilité de chaque variable,

– on appellera **mouvement,** la modification d'un sous-ensemble de composantes d'une solution. Un mouvement permet d'explorer l'espace des solutions possibles en passant d'une solution à une autre,
– le **voisinage** d'une solution s noté $V(s)$ est l'ensemble des solutions accessibles obtenues avec un seul mouvement.

Fonction d'évaluation : la fonction d'évaluation est une somme pondérée de valeurs de pénalité associées aux variables du problème.

$$f(s) = \sum_{c \in (C)} w_c \times \delta_c(s), \begin{cases} \delta_c(s) = 1 \text{ si la contrainte est satisfaite} \\ \delta_c(s) = 0 \text{ sinon} \end{cases}$$

Une méthode de recherche locale est un parcours de l'espace des solutions à partir d'une solution initiale. La recherche s'effectue en modifiant, mouvement après mouvement, la solution courante pour maximiser (ou minimiser) la fonction d'évaluation. Une recherche locale exploite :
– Une solution initiale,
– le voisinage de chaque variable,
– les critères pour déterminer le mouvement sélectionné à chaque itération.
La recherche locale permet d'utiliser des heuristiques pour décider des mouvements possibles et de la sélection des mouvements sur la solution courante.

b) Choix de la fonction d'évaluation et du voisinage

Il n'est pas possible de déterminer si des contraintes de triplets d'AS sont plus significatives que d'autres au sens de la validité économique. Le poids w_c fixé pour chaque contrainte est fixé à 1. La fonction de pénalité f est égale à la somme du nombre de contraintes valides.
Une façon simple d'utiliser la recherche locale est de modifier variable par variable une solution. Deux solutions sont voisines si elles diffèrent en fonction d'une seule variable. Le voisinage est trié suivant les valeurs de la fonction f. La fonction f permet de guider la recherche vers les voisinages prometteurs. Soient s et s' deux solutions voisines. On note $\Delta_f = f(s') - f(s)$. Alors :

Si $\Delta_f > 0$: la solution s' est meilleure que la solution s

Sinon si $\Delta_f = 0$: la solution s' est équivalente à la solution s. Le nombre de contraintes violées est inchangé.

Sinon $(\Delta_f < 0)$: la solution s est meilleure que la solution s'

On utilise l'algorithme glouton qui mets à jour une solution en sélectionnant à chaque itération la variable à modifier et sa nouvelle valeur qui conduit à la meilleure fonction évaluation :

Algorithme glouton de recherche locale
s_0 : solution initiale

$s := s_0$

Tant que une solution complétement optimale n'est pas obtenue
et que le nombre d'itération n'est pas atteint
et que le temps maximum n'est pas écoulé
$(L_i, l') = Max_{s' \in V(s)} (f(s') - f(s))$
$s[L_i] := l'$
fin tant que

c) Solution initiale

Avec une méthode de recherche locale, on peut utiliser n'importe quelle sous-configuration comme solution de départ. On utilisera une solution initiale avec une valeur spéciale UNK. La valeur UNK conduit à des contraintes insatisfaites. Par abus de langage, on notera toujours $\mathbb{L}$ l'ensemble des valeurs possibles avec la valeur UNK incluse. On ne fera jamais de mouvements vers la valeur UNK.
On peut également fixer au préalable une partie de la solution. Pour cela, il suffit réduire le domaine de définition de chaque variable à l'unique valeur préalablement connue.

d) Contraintes additionnelles

Certaines des contraintes proposées dans 3.1.3 sont prises en compte de la façon suivante :

- Les contraintes (C_{stub}) (voir définition 19) sont incorporées en réduisant l'ensemble des valeurs possibles pour les arcs concernés. Pour chaque lien logique ASX-ASY avec $X < Y$ (numéro d'AS X et Y dans l'ordre croissant), si ASX est stub et l'AS Y n'est pas stub on supprime la valeur $P2C$ du domaine de définition de la variable $L\overrightarrow{(u_x, u_y)}$, et si AS$X$ n'est pas stub et si l'AS Y est stub, on supprime la valeur $C2P$ du domaine de définition de la variable $L\overrightarrow{(u_x, u_y)}$.
- Les contraintes (C_{cc}) (voir définition 22) sont incorporées dans le mécanisme de sélection du voisinage. Durant la recherche, un graphe des AS dirigé noté G_{cc} est maintenu à jour en fonction des arcs de label $C2P$ ou $P2C$ dans la solution courante. Pour chaque lien logique ASX-ASY avec $X < Y$, on ajoute l'arc $\overrightarrow{(ASX, ASY)}$ au graphe

G_{cc} si $L_{X,Y} = L(\overrightarrow{ASX, ASY}) = C2P$, ou l'arc $(\overrightarrow{ASY, ASX})$ au graphe G_{cc} si $L_{X,Y} = L(\overrightarrow{ASX, ASY}) = P2C$. Les mouvements qui introduisent un cycle dans le graphe sont supprimés du voisinage de chaque solution.

e) Mécanisme Tabou

Lorsqu'un minimum local est atteint, l'algorithme glouton répète un cycle d'itérations sans pouvoir obtenir une meilleure solution. Pour pouvoir échapper à des minima locaux, on utilise une liste glissante de mouvements interdits. Cette liste de mouvements, appelée liste tabou, permet à la recherche locale de pouvoir éviter certains sous-espaces de recherches dans lesquels l'algorithme glouton reste bloqué. A chaque itération, une variable L_i change de la valeur l vers la valeur l' et le couple (L_i, l) est enregistré dans la liste. Pour éviter le mouvement retour, le label l est interdit pour la variable L_i pendant un nombre limité d'itérations. La liste des mouvements tabous est gérée comme une pile FIFO[13] et sa taille peut être statique ou dynamique. Les deux techniques ont été expérimentées. La gestion statique de la liste a été retenue.

3.2.3 Améliorations du choix du voisinage

Un difficulté subsiste du fait de la dégénérescence des solutions. Il est en effet important de ne pas se laisser piéger dans un sous-ensemble de solutions de même coût (ayant le même nombre de contraintes valides, voir le lemme 18). De plus, il est intéressant de pouvoir choisir une solution (qui vérifie d'autres critères) parmi toutes les solutions de même coût. Le mécanisme tabou permet de sortir de sous-espaces de petite taille avec un minimum local, mais pas des sous-espaces de recherche où il existe au moins une solution locale sous-optimale avec beaucoup d'accords de peering. En effet le nombre de solutions équivalentes varie exponentiellement avec le nombre de liens de peering. Pour se concentrer sur l'obtention de bonnes solutions, on utilise un voisinage plus pertinent (de plus petite taille) et trié par une heuristique.

a) Réduction de l'espace des variables sélectionnées

Premièrement, on réduit le voisinage d'une solution en interdisant les mouvements sur des variables dont toutes les contraintes sont valides. En d'autres termes, tant qu'un

[13] First In First Out

accord d'interconnexion n'est impliqué dans aucun triplet invalide (toutes les contraintes de successivité l'impliquant sont satisfaites), alors aucune nouvelle valeur du label de cet accord d'interconnexion ne sera sélectionnée dans l'algorithme.

b) Tri du voisinage

Le tri du voisinage est important, car il définit la direction de recherche générale. La valeur UNK est une valeur "invalide". On privilégiera toujours les mouvements en provenance d'une variable avec la valeur UNK. Puis on définit un tri heuristique. La phase de tri est présentée ci-dessous :

UNKNOWN : Si la solution courante a des variables avec la valeur UNK, alors restreindre le voisinage aux changements de valeurs pour ces variables uniquement.

Ordre de réparation (RO) : préférer les mouvements dont la valeur Δ_f est la plus grande,

Ordre des valeurs (VO1) : en cas d'égalité, préférer les mouvements vers la valeur $PEER$ que les mouvements vers les valeurs $P2C$ et $C2P$,

Ordre des variables (VO2) : si il existe encore des mouvements à égalité, les variables $L_{i,j}$ sélectionnées vérifient :

$$\{L_{i,j}\} = ArgMax_{L_{i,j}}(deg(AS_i) + deg(AS_j) + \begin{array}{l} |\{(u_i,u_j,u_k) \in P'_3\}| \\ +|\{(u_j,u_i,u_k) \in P'_3\}| \\ +|\{(u_k,u_i,u_j) \in P'_3\}| \\ +|\{(u_k,u_j,u_i) \in P'_3\}| \end{array})$$

Tie break : s'il reste encore plusieurs mouvements possibles, on en sélectionne un au hasard.

La sélection (RO) est la sélection de l'algorithme glouton. La sélection (VO1) permet de répondre par une heuristique au problème $PEERING - DISCOVERY$. La sélection (VO2) tente de donner des valeurs aux variables fortement contraintes, et dont les AS du lien ont un fort degré. Les ordres (VO1) et (VO2) permettent de privilégier les accords de peering entre les AS de fort degré et avec beaucoup de triplets.

Après la présentation de la formulation MaxCSP du problème MaxTOR3 et de l'algorithme tabou amélioré pour le résoudre heuristiquement, on va étudier en 3.3 les problèmes MaxTOR et MaxTOR3 sous une forme généralisée. L'analyse et la validation des solutions calculées par nos algorithmes sont reportés dans 3.4.

3.3 Généralisation du problème d'inférence des accords d'interconnexion

Dans le problème d'inférence des accords d'interconnexion, on cherche à associer un unique label parmi ceux de $\mathbb{L} = \{C2P, P2C, PEER\}$ aux arcs d'un graphe symétrique, tels qu'un ensemble de chemins élémentaires du graphe (éventuellement de taille 3 seulement) respectent l'expression régulière déduite de la matrice de contraintes de successivité. On généralise le problème d'inférence des accords d'interconnexion, en définissant des labels arbitraires pour les arcs d'un graphe symétrique et une matrice de contraintes de succession de labels. Cette généralisation appelée MaxVCAP[14] permet de modéliser de nombreux problèmes comme par exemple le problème du stable maximum, le problème Max-SAT, ou le problème de K-coloration d'un graphe. On montre comment obtenir une formulation non-linéaire en nombre entiers, puis on propose de linéariser cette formulation pour obtenir un modèle de programmes linéaires en nombre entiers. Cette approche basée sur la linéarisation du modèle non-linéaire initial permet d'utiliser ensuite des algorithmes de résolution efficaces comme les méthodes de "Branch & Cut". Notre méthodologie applicable dans le cas général[15] est améliorée pour résoudre le cas particulier du problème d'inférence des accords d'interconnexion MaxTOR3. L'algorithme présenté dans l'annexe C joue un rôle important dans le procédé de linéarisation de la formulation d'un problème MaxVCAP.

3.3.1 Définition des problèmes VCAP

En généralisant le problème MaxTOR, on propose plusieurs formulations d'un nouveau problème qui consiste dans une recherche du label des arcs d'un graphe symétrique qui maximisent le nombre de chemins valides. La validité d'un chemin se définit comme la validité de toutes les paires d'arcs successifs dans celui-ci. On définit la validité d'une paire d'arc dans b) comme la succession valide des labels sur les deux arcs. Les nouveaux problèmes MaxVCAP et MaxVCAP3 sont présentées en c).

[14] Maximum Valid Consecutive Arc Pair

[15] sous réserve de pouvoir calculer les hyperplans nécessaires pour décrire l'enveloppe convexe d'un ensemble de points de l'espace $\{0,1\}^{|\mathbb{L}|}$.

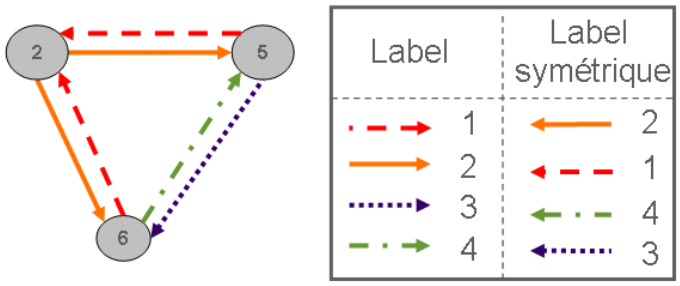


FIG. 3.8 – Graphe avec trois noeuds pour les problèmes VCAP

a) Modèle de graphe pour les problèmes MaxVCAP et MaxVCAP3

Soit un ensemble de labels quelconques noté $\mathbb{L} = \{L_i,\, i \in 1 \dots |\mathbb{L}|\}$. On suppose qu'il existe un label symétrique pour chaque label et on note $S(L_i)$ le label symétrique d'un label L_i : $\forall i \in \{1 \dots |\mathbb{L}|\}\,,\, \exists j \in \{1 \dots |\mathbb{L}|\} \;/\, S(L_i) = L_j$

On suppose également que tout label est le symétrique de son symétrique.

Soit un graphe $G = (V,\, E)$ orienté. L'ensemble $V = \{u_1, u_2, \dots, u_n\}$ désigne les n noeuds du graphe. L'ensemble $E = (e_1, e_2, \dots, e_{2p})$ désigne ses arcs. Chaque fois que deux noeuds u_i et u_j sont connectés (avec $i \neq j$) par un arc, il existe deux arcs $e_{ij} = \overrightarrow{(u_i, u_j)}$ et $e_{ji} = \overrightarrow{(u_j, u_i)}$ dans l'ensemble E. A chaque arc du graphe G, on associe un unique label de l'ensemble $\mathbb{L}$. On notera $L(e) = L_{i,j}$ le label porté par un arc $e = \overrightarrow{(u_i, u_j)} \in E$. Le graphe G est alors tel que les labels des arcs symétriques sont symétriques (voir figure 3.8) : $\forall e = \overrightarrow{(u_i, u_j)} \in E,\, L_{i,j} = S(L_{j,i})$

b) Contraintes VCAP dans le problème

Les contraintes de validité des chemins sont déterminées par la matrice de contraintes de succession $\mathbb{M} \in \mathcal{M}_{|\mathbb{L}|,|\mathbb{L}|}\{0,1\}$. La matrice $\mathbb{M}$ exprime la validité des différents couples de labels possibles :

Définition 25. *Matrice des contraintes de succession des labels*

$$\mathbb{M} = L_i \;\rightarrow \begin{pmatrix} & \overbrace{\quad}^{L_j} & \\ \dots & \dots & \dots \\ \dots & \mathbb{M}_{i,j} & \dots \\ \dots & \dots & \dots \end{pmatrix}$$

$$\forall i,j \in \{1 \dots |\mathbb{L}|\}^2,\ (\mathbb{M})_{i,j} = \begin{cases} 1 \textit{ si la succession des deux labels } L_i \textit{ et } L_j \textit{ est valide} \\ 0 \textit{ sinon} \end{cases}$$

La matrice $\mathbb{M}$ permet d'exprimer les contraintes de validité (C_{vcap}).

Définition 26. *Contraintes* (C_{vcap})
Soit un couple d'arcs consécutifs ($e_{i,j}$, $e_{j,k}$) *du graphe* G *avec* $e_{i,j} = \overrightarrow{(u_i, u_j)}$ *et* $e_{j,k} = \overrightarrow{(u_j, u_k)}$. *Alors on peut exprimer la validité du triplet* (u_i, u_j, u_k)*par les expressions suivantes :*

(C_{vcap}) *le triplet* (u_i, u_j, u_k) *est valide (la paire d'arc* ($e_{i,j}$, $e_{j,k}$) *est valide)*

$\Leftrightarrow$

(C_{vcap}) $(L_{i,j}, L_{j,k}) \in \{(L_x, L_y) / \mathbb{M}_{x,y} = 1\}$

Soit $p = (p_1, \dots, p_n) \in P$ *avec* $n \geq 3$. *On dit que le chemin* p *est valide si et seulement si les* $n-2$ *contraintes* (C_{vcap}) *déduites du chemin sont valides :*

(C_{vcap}) $\forall 1 \leq i \leq n-2,\ (L_{i,i+1}, L_{i+1,i+2}) \in \{(L_x, L_y) / \mathbb{M}_{x,y} = 1\}$

Les contraintes (C_{vcap}) vont permettre de formuler les problèmes MaxVCAP et MaxVCAP3.

c) Formulations des problèmes

Soit un ensemble de chemins P du graphe G. On note P_3 l'ensemble des sous-chemins de P de taille 3[16] :

$$P_3 = \{(p_{i-1}, p_i, p_{i+1}) / \exists p = (p_1, \dots, p_k) \in P,\ k \geq 3,\ 2 \leq i \leq k-1\}$$

$$\text{On pose } P'_3 = \left\{ p_3 = (u_i, u_j, u_k) / i < k \textit{ et } \left\{ \begin{array}{c} p_3 \in P_3 \\ ou \\ (u_k, u_j, u_i) \in P_3 \end{array} \right\} \right\}$$

On note T l'ensemble des arcs adjacents issus de tous les chemins de P'_3 :

$$T = \bigcup_{(p_i p_j p_k) \in P'_3} \left(\overrightarrow{(p_i, p_j)}, \overrightarrow{(p_j, p_k)} \right)$$

La définition d'un ensemble de chemins P'_3 permet d'obtenir une bijection entre les

[16] On entend ici par taille d'un chemin le nombre de sommets.

contraintes (C_{vcap}) et les chemins de P'_3. En effet :

$$\forall u_i, u_j, u_k \in V, \left\{ \begin{array}{l} (u_i, u_j, u_k) \in P_3 \\ \text{ et} \\ (u_k, u_j, u_i) \in P_3 \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} (u_i, u_j, u_k) \in P'_3 \\ \text{ ou (exclusif)} \\ (u_k, u_j, u_i) \in P'_3 \end{array} \right\}$$

MaxVCAP3 (Maximum Valid Consecutive Arc Pairs) Affecter un label issu de $\mathbb{L}$ à chaque arc du graphe G tel qu'un maximum de chemins de P'_3 vérifient les contraintes (C_{vcap}).

MAXVCAP (Maximum Valid Consecutive Arc Paths) Affecter un label issu de $\mathbb{L}$ à chaque arc du graphe G tel que le nombre de chemins de P vérifiant à la fois toutes les contraintes (C_{vcap}) soit maximum.

– Remarque : si on s'intéresse aux solutions avec un maximum d'arcs de label fixé, on peut définir la généralisation de PEERING-DISCOVERY comme le problème MaxLabelVCAP (Maximum Label Consecutive Arc Pairs).

d) Exemples de problèmes MaxVCAP

Ci-dessous on présente deux exemples de problèmes MaxVCAP.

Le problème SAT sous une forme MaxVCAP

Soit un ensemble de variables $x_i \in \{0,1\}$, $i \in \{1,\ldots,n\}$. Soit une formule logique $\mathcal{F}$ formée de m clauses sur les n variables : $\mathcal{F} = \wedge_{i=1}^{m}(\vee_{k\in\mathbb{I}_j} x_k)$.

Question SAT : $\mathcal{F}$ est-elle satisfiable ? Existe-t-il une affectation de chaque variable x_i à une valeur possible telle que la formule soit vérifiée.

Le problème SAT est NP-complet [34]. Un problème MaxVCAP peut être formulé avec les données suivantes :

– $\mathcal{F} = \wedge_{i=1}^{m}(C_j)$
– $\forall C_j = (\vee_{k\in\mathbb{I}_j} x_k)$, $j \in \{1,\ldots,m\}$, $\forall \mathbb{I}_j \subset \{1,\ldots,n\}$,
– $C_j = (x_{j_1} \wedge x_{j_2} \wedge \ldots \wedge x_{j_{|\mathbb{I}_j|}}), 1 \leq j_1 < j_2 < \ldots < j_{|\mathbb{I}_j|} \leq n$
– $V = \{u\} \cup \{u_i\}_{i\in\{1,\ldots,n\}}$, $E = \{\overrightarrow{(u_i, u)}, i \in \{1,\ldots,n\}\} \cup \{\overrightarrow{(u, u_i)}, i \in \{1,\ldots,n\}\})$
– $\mathbb{L}_1 = 0 = S(\mathbb{L}_1)$, $\mathbb{L}_2 = 1 = S(\mathbb{L}_2)$; $\mathbb{M} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
– $P = \left\{(u_{j_i}, u, u_{j_{i+1}}) \,/\, j \in \{1,\ldots,m\}, 1 \leq i < |\mathbb{I}_j|\right\}$

Le problème SAT a une solution si et seulement si le problème MaxVCAP a une solution optimale telle que $\sum_{p \in P} \delta_p = |P|$.

Problème de $k-$coloration de graphe

On appelle $k-$coloration d'un graphe l'association d'une couleur parmi k à chaque noeud d'un graphe telle que le nombre d'arcs connectant deux noeuds de même couleur soit minimal. On note $G_c = (V_c = \{v_1, \ldots, v_n\}, E_c)$ le graphe non orienté à colorier.
Un problème MaxVCAP peut être formulé grâce à k labels correspondant aux k couleurs. La couleur d'un noeud du graphe G_k est ici le label d'un arc d'un graphe intermédiaire $G = (V, E)$. L'ensemble des noeuds et des sommets est définit de la manière suivante :

– $V = \{u\} \cup \{u_i\}_{i \in \{1,\ldots,n\}}$; $\quad E = \{\overrightarrow{(u_i, u)},\ i \in \{1, \ldots, n\}\} \cup \{\overrightarrow{(u, u_i)},\ i \in \{1, \ldots, n\}\})$

Un label est associé à chaque couleur, et les contraintes induites entre couleurs sont placées dans la matrice $\mathbb{M}$.

– $\mathbb{L}_1 = 1 = S(\mathbb{L}_1), \ldots, \mathbb{L}_k = k = S(\mathbb{L}_k)$; $\mathbb{M} = \begin{pmatrix} 0 & 1 & \ldots & \ldots & 1 \\ 1 & 0 & 1 & \ldots & \ldots \\ \ldots & 1 & 0 & 1 & \ldots \\ \ldots & \ldots & 1 & 0 & 1 \\ 1 & \ldots & \ldots & 1 & 0 \end{pmatrix}$

Le problème de $k-$coloration revient ici à considérer tous les triplets de noeuds du graphe G possible (ce qui correspond à toutes les paires de noeuds adjacents dans le graphe initial).

– $P = \{(u_i, u, u_j) \,/\, (v_i, v_j) \in E_c,\ i < j\}$

3.3.2 Formulations en nombres entiers

On propose le codage en nombre entiers des labels des arcs en a). On montre d'abord comment obtenir des programmes en nombres entiers avec une fonction objectif quadratique dans b). On introduit ensuite une variable pour chaque contrainte (C_{vcap}) dans c). On montre alors comment obtenir des formulations linéaires pour les problèmes dans d). La linéarisation des contraintes (C_{vcap}) exploite l'algorithme présenté dans l'annexe C. Après cette étude des problèmes sous une forme générale, on donne les formulations précises du problème MaxTOR et MaxTOR3 dans 3.3.3.

a) Variables de label

L'encodage des variables de labels pour les arcs est libre. Soit c le nombre de variables binaires utilisées pour représenter un label. On définit la fonction Λ qui permet d'obtenir le vecteur de $\{0,1\}^c$ représentant un label :

$$\begin{array}{rcccc} \Lambda & : & \mathbb{L} & \longrightarrow & \{0,1\}^c \\ & & \mathbb{L}_i & \longmapsto & (\delta_1 \quad \dots \quad \delta_c) \end{array}$$

On notera abusivement S la fonction de symétrie définie sur $\{0,1\}^c$:

$$\begin{array}{rcccc} S & : & \Lambda(\mathbb{L}) \subset \{0,1\}^c & \rightarrow & \{0,1\}^c \\ & & \beta = (\delta_1 \quad \dots \quad \delta_c) & \longmapsto & S(\beta) = (\delta'_1 \quad \dots \quad \delta'_c) \end{array}$$

La taille c d'un système de codage binaire pour les éléments de $\mathbb{L}$ est bornée :

$$\frac{ln(|\mathbb{L}|)}{ln(2)} \leq c \leq |\mathbb{L}|$$

En effet, le système de codage ne peut représenter que 2^c valeurs différentes. Donc $|\mathbb{L}| \leq 2^c$. Le pire cas de codage correspond à associer une variable binaire pour chaque label. Chaque variable serait codée de la façon suivante :

$$\Lambda(\mathbb{L}_i) = (0, \dots 0, \underbrace{1}_{position\, i}, 0, \dots, 0)$$

On notera D comme l'ensemble des codes entiers représentant un label :

$$D = \{\Lambda(\mathbb{L}_i) / \mathbb{L}_i \in \mathbb{L}\} \subset \{0,1\}^c$$

Exemples avec le problème d'inférence des accords d'interconnexion :

- Si $c = 2$ et $\Lambda(P2C) = (0\,1)$ et $\Lambda(C2P) = (0\,0)$ et $\Lambda(PEER) = (1\,0)$, alors $D = \{(0\,0),(1\,0),(0\,1)\}$
- Si $c = 3$ et $\Lambda(P2C) = (1\,0\,0)$ et $\Lambda(C2P) = (0\,0\,1)$ et $\Lambda(PEER) = (0\,1\,0)$, alors $D = \{(1\,0\,0),(0\,1\,0),(0\,0\,1)\}$

On modélise donc chaque label par c variables binaires. Pour chaque paire d'arc symétriques, on associe c variables binaires. Pour tout arc $e = \overrightarrow{(u_i, u_j)} \in E$:

Si $i < j$: $\Lambda(L_{i,j}) = (\delta_1^{(i,j)}, \dots, \delta_c^{(i,j)}) \in \{0,1\}^c$définit c variables,

Si $i > j$: $L_{i,j} = S(L_{j,i}) \in \{0,1\}^c$ est déduit de la variable L_{ji}.

On introduit une seule variable pour chaque paire d'arcs symétriques. On définit ainsi une partition de l'ensemble $P'_3 = P_{dd} \cup P_{di} \cup P_{id}$ telle que :

$P_{dd} = \{p_3 = (u_i, u_j, u_k) \in P'_3 \,/\, i < j \text{ et } j < k\}$

$P_{di} = \{p_3 = (u_i, u_j, u_k) \in P'_3 \,/\, i < j \text{ et } j > k\}$
$P_{id} = \{p_3 = (u_i, u_j, u_k) \in P'_3 \,/\, i > j \text{ et } j < k\}$

On définit maintenant trois matrices $\mathbb{M}_{dd}$, $\mathbb{M}_{di}$ et $\mathbb{M}_{id}$ obtenues par réorganisation des éléments de la matrice $\mathbb{M}$:

(C_{dd}) $\mathbb{M}_{dd} = ((\mathbb{M}_{dd})_{x,y} = \mathbb{M}_{x,y})_{x,y \in \{1,\ldots,|\mathbb{L}|\}} = \mathbb{M}$

$$(C_{di}) \quad \mathbb{M}_{di} = ((\mathbb{M}_{di})_{x,y} = \mathbb{M}_{x,z}) \text{ tel que } \begin{cases} x, y \in \{1, \ldots, |\mathbb{L}|\} \\ L_z = S(L_y) \end{cases}$$

$$(C_{id}) \quad \mathbb{M}_{id} = ((\mathbb{M}_{id})_{x,y} = \mathbb{M}_{z,y}) \text{ tel que } \begin{cases} x, y \in \{1, \ldots, |\mathbb{L}|\} \\ L_z = S(L_x) \end{cases}$$

Pour le problème d'inférence des accords d'interconnexion, on obtient :

– Si $c = 3$ et $\mathbb{L}_1 = C2P$ et $\mathbb{L}_2 = P2C$ et $\mathbb{L}_3 = PEER$, alors

$$\mathbb{M} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{ et } \mathbb{M}_{di} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \text{ et } \mathbb{M}_{id} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

b) Formulations non linéaires

Si on associe le codage simple de taille $|\mathbb{L}|$ suivant :

$$\Lambda(\mathbb{L}_i) = (0, \ldots 0, \underbrace{1}_{position\, i}, 0, \ldots, 0), \forall 1 \leq i \leq |\mathbb{L}|$$

Alors, on peut obtenir les deux formulations quadratiques suivantes de MaxVCAP3 et de MaxVCAP.

Problème MaxVCAP3 :

$$\begin{aligned} \text{Maximiser} \quad & \sum_{(u_i,u_j,u_k) \in P_{dd}} (\delta_1^{(i,j)}, \ldots, \delta_{|\mathbb{L}|}^{(i,j)}) \times \mathbb{M}_{d,d} \times^t (\delta_1^{(j,k)}, \ldots, \delta_{|\mathbb{L}|}^{(j,k)}) \\ & + \sum_{(u_i,u_j,u_k) \in P_{di}} (\delta_1^{(i,j)}, \ldots, \delta_{|\mathbb{L}|}^{(i,j)}) \times \mathbb{M}_{d,i} \times^t (\delta_1^{(k,j)}, \ldots, \delta_{|\mathbb{L}|}^{(k,j)}) \\ & + \sum_{(u_i,u_j,u_k) \in P_{id}} (\delta_1^{(j,i)}, \ldots, \delta_{|\mathbb{L}|}^{(j,i)}) \times \mathbb{M}_{i,d} \times^t (\delta_1^{(j,k)}, \ldots, \delta_{|\mathbb{L}|}^{(j,k)}) \end{aligned}$$

$\forall e = \overrightarrow{(u_i, u_j)} \in E, \; i < j$

$$(C_{label}) \quad \sum_{1 \leq k \leq |\mathbb{L}|} \delta_k^{(i,j)} = 1$$

$$(\delta_1^{(i,j)}, \ldots, \delta_{|\mathbb{L}|}^{(i,j)}) \in \{0,1\}^{|\mathbb{L}|}$$

Problème MaxVCAP

Maximiser $\sum_{p \in P} \delta_p$

$\forall e = \overrightarrow{(u_i, u_j)} \in E,\ i < j$

(C_{label}) $\sum_{1 \leq k \leq |\mathbb{L}|} \delta_k^{(i,j)} = 1$

$(\delta_1^{(i,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(i,j)}) \in \{0,1\}^{|\mathbb{L}|}$

$\forall p \in P, \forall (u_i, u_j, u_k) \subset p,$

(C_{dd}) $\delta_p = (\delta_1^{(i,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(i,j)}) \times \mathbb{M}_{d,d} \times {}^t(\delta_1^{(j,k)}, \ldots,\ \delta_{|\mathbb{L}|}^{(j,k)})$, si $i < j$ et $j < k$

(C_{di}) $\delta_p = (\delta_1^{(i,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(i,j)}) \times \mathbb{M}_{d,i} \times {}^t(\delta_1^{(k,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(k,j)})$, si $i < j$ et $j > k$

(C_{id}) $\delta_p = (\delta_1^{(j,i)}, \ldots,\ \delta_{|\mathbb{L}|}^{(j,i)}) \times \mathbb{M}_{i,d} \times {}^t(\delta_1^{(j,k)}, \ldots,\ \delta_{|\mathbb{L}|}^{(j,k)})$, si $i > j$ et $j < k$

(C_{dd}) $\delta_p = (\delta_1^{(j,i)}, \ldots,\ \delta_{|\mathbb{L}|}^{(j,i)}) \times \mathbb{M}_{d,d} \times {}^t(\delta_1^{(k,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(k,j)})$, si $i > j$ et $j > k$

$\delta_p \in \{0,1\}$

c) Contraintes de validité de labels successifs

Pour chaque chemin p de l'ensemble P_3', on déduit une contrainte de type (C_{vcap}). On définit alors une variable de validité d'un triplet de noeud $\delta_p \in \{0,1\}$ qui prend la valeur 1 si la contrainte est satisfaite ou la valeur 0 si elle est violée.

Les contraintes de type (C_{vcap}) sont scindées en 3 familles suivant l'orientation des arcs dans chaque triplet. Soit (C_{vcap}) une contrainte pour une paire d'arcs consécutifs ($e_{i,j}$, $e_{j,k}$)$\in T$. En notant $e_{i,j} = \overrightarrow{(u_i, u_j)}$ et $e_{j,k} = \overrightarrow{(u_j, u_k)}$, il vient :

(C_{dd}) $\delta_c = \mathbb{M}_{x,y}$ avec $L_{i,j} = \mathbb{L}_x$ et $L_{j,k} = \mathbb{L}_y$, si $i < j$ et $j < k$

(C_{di}) $\delta_c = \mathbb{M}_{x,y}$ avec $L_{i,j} = \mathbb{L}_x$ et $S(L_{k,j}) = \mathbb{L}_y$, si $i < j$ et $j > k$

(C_{id}) $\delta_c = \mathbb{M}_{x,y}$ avec $S(L_{j,i}) = \mathbb{L}_x$ et $L_{j,k} = \mathbb{L}_y$, si $i > j$ et $j < k$

d) Formulations linéaires des problèmes MaxVCAP3 et MaxVCAP

On se replace dorénavant dans le cas général où un encodage Λ de taille c est fixé.

On pose pour tout $\mathcal{M} \in \mathcal{M}_{c,c}(\{0,1\})$, l'ensemble $\Phi(\mathcal{M}) = \{\phi_1, \ldots, \phi_p\}$ tel que :

$$\Phi(\mathcal{M}) = \left\{ (\alpha_1, \ldots, \alpha_{2*c}, \delta) \in D \times D \times \{0,1\} \ / \begin{array}{l} \Lambda(L_i) = (\alpha_1, \ldots, \alpha_c) \\ \Lambda(L_j) = (\alpha_{c+1}, \ldots, \alpha_{2*c}) \\ \delta = \mathcal{M}_{i,j} \end{array} \right\}$$

Si le codage Λ est tel que $c = |\mathbb{L}|$ comme dans b), alors $\Phi(\mathcal{M})$ peut s'écrire :

$$\Phi(\mathcal{M}) = \left\{ (\alpha_1, \ldots, \alpha_{2*c}, \delta) \in D \times D \times \{0,1\} \ / \ (\alpha_1, \ldots, \alpha_c) \times \mathbb{M} \times \begin{pmatrix} \alpha_{c+1} \\ \ldots \\ \alpha_{2*c} \end{pmatrix} = \delta \right\}$$

Soit $(u_i, u_j, u_k) \in P'_3$. Les contraintes (C_{vcap}) peuvent s'exprimer de la façon suivante (on ne montre que le cas $i < j < k$) :

(C_{dd}) $\qquad (\delta_1^{(i,j)}, \ldots, \delta_c^{(i,j)}, \delta_1^{(j,k)}, \ldots, \delta_c^{(j,k)}, \delta_{(u_i,u_j,u_k)}) \in \Phi(M_{dd})$

Le lemme suivant va permettre de linéariser la formulation des familles de contraintes (C_{dd}), (C_{di}) et (C_{id}) :

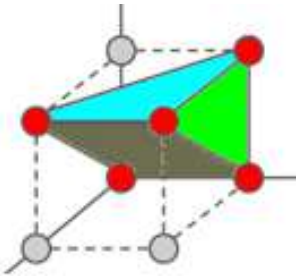
Lemme 27. *Un point de l'espace* $[0,1]^n$ *appartient à un ensemble de points entiers si et seulement si le point est entier et élément de l'enveloppe convexe des points.*
Autrement dit, $\forall m, \forall n, \forall \phi_1, \ldots, \phi_m \in \{0,1\}^n \times \ldots \times \{0,1\}^n$, $\forall \gamma \in [0,1]^n$,

$$\gamma \in \{\phi_1, \ldots, \phi_m\} \Leftrightarrow \gamma \in conv(\phi_1, \ldots, \phi_m) \cap \{0,1\}^n$$

où $conv(E)$ *désigne l'enveloppe convexe d'un ensemble d'éléments* E *définit de la manière suivante pour tout* $n \geq 1$ *et pour tout* $m \geq 1$ *:*

$$\forall E = (e_1, \ldots, e_m) \in (\{0,1\}^n)^m, conv(E) = \left\{ \beta \in [0,1]^n \ / \ \begin{array}{c} \forall 1 \leq i \leq n, \ \beta_i = \sum_{1 \leq k \leq m} \lambda_k . (e_k)_i, \\ (\lambda_1, \ldots, \lambda_m) \in [0,1]^m, \\ \sum_{1 \leq k \leq m} \lambda_k = 1 \end{array} \right\}$$

Preuve : [voir annexe C]. □



(a) Sur cette figure, la dimension est $n = 3$. Les points rouges correspondent aux $m = 5$ éléments $(\phi_1, \ldots, \phi_m) \in \{0,1\}^3 \times \ldots \times \{0,1\}^3$. L'enveloppe convexe correspond au volume dessiné. Les points gris sont les éléments de l'ensemble $\{0,1\}^n \setminus (\phi_1, \ldots, \phi_m)$.

FIG. 3.9 – Illustration du lemme 27

Le résultat du lemme est illustré dans la figure 3.9. Grâce au théorème de Minkowski-Weyl il existe une description linéaire de chaque enveloppe convexe des familles de points $\Phi(M_{dd}), \Phi(M_{di})$ et $\Phi(M_{id})$. Grâce au lemme on sait que l'enveloppe convexe permet d'obtenir une description nécessaire et suffisante de chaque famille de points.

On peut donc utiliser l'algorithme présenté dans l'annexe C pour déterminer les inéquations descriptives des trois polytopes $conv(\Phi(M_{dd})), conv(\Phi(M_{di}))$ et $conv(\Phi(M_{id}))$. On obtient donc une formulation linéaire pour chaque contrainte (C_{vcap}) déduite de l'ensemble P'_3. Pour $(u_i, u_j, u_k) \in P'_3$ avec $i < j < k$ on obtient :

$$(C_{dd}) \quad \mathbb{A}_{dd}. \begin{pmatrix} \delta_1^{(i,j)} \\ \ldots \\ \delta_c^{(i,j)} \\ \delta_1^{(j,k)} \\ \ldots \\ \delta_c^{(j,k)} \\ \delta_{(u_i,u_j,u_k)} \end{pmatrix} \leq \mathbb{B}_{dd},\ (\delta_1^{(i,j)}, \ldots, \delta_c^{(i,j)}, \delta_1^{(j,k)}, \ldots, \delta_c^{(j,k)}, \delta_{(u_i,u_j,u_k)}) \in \{0,1\}^{2c+1}$$

De façon complètement symétrique, on obtient quatre autres matrices $(\mathbb{A}_{di}, \mathbb{B}_{di})$ et $(\mathbb{A}_{id}, \mathbb{B}_{id})$ qui permettent de décrire respectivement les enveloppes convexes de $\Phi(M_{di})$ et de $\Phi(M_{id})$.

MaxVCAP3 (formulation PLNE)

Maximiser $\sum_{p \in P'_3} \delta_p$

$\forall e = \overrightarrow{(u_i, u_j)} \in E,\ i < j$

(C_{label}) $\sum_{1 \leq i \leq |\mathbb{L}|} \delta_i^{(i,j)} = 1$

$(\delta_1^{(i,j)}, \ldots,\ \delta_{|\mathbb{L}|}^{(i,j)}) \in \{0,1\}^{|\mathbb{L}|}$

$\forall (u_i, u_j, u_k) \in P'_3,$

$$(C_{dd}) \quad \mathbb{A}_{dd}. \begin{pmatrix} \delta_1^{(i,j)} \\ \ldots \\ \delta_c^{(i,j)} \\ \delta_1^{(j,k)} \\ \ldots \\ \delta_c^{(j,k)} \\ \delta_{(u_i,u_j,u_k)} \end{pmatrix} \leq \mathbb{B}_{dd}, \text{ si } i < j < k$$

$$(C_{di}) \quad \mathbb{A}_{di}. \begin{pmatrix} \delta_1^{(i,j)} \\ \ldots \\ \delta_c^{(i,j)} \\ \delta_1^{(k,j)} \\ \ldots \\ \delta_c^{(k,j)} \\ \delta_{(u_i,u_j,u_k)} \end{pmatrix} \leq \mathbb{B}_{di}, \text{ si } i < j \text{ et } j > k$$

$$(C_{id}) \quad \mathbb{A}_{id}. \begin{pmatrix} \delta_1^{(j,i)} \\ \ldots \\ \delta_c^{(j,i)} \\ \delta_1^{(j,k)} \\ \ldots \\ \delta_c^{(j,k)} \\ \delta_{(u_i,u_j,u_k)} \end{pmatrix} \leq \mathbb{B}_{id}, \text{ si } i > j \text{ et } j < k$$

$\delta_{(u_i,u_j,u_k)} \in \{0,1\}$

MaxVCAP (formulation PLNE)

$$\text{Maximiser} \quad \sum_{p \in P} \delta_p$$

$$\forall e = \overrightarrow{(u_i, u_j)} \in E, \;\; i < j$$

$$(C_{label}) \quad \sum_{1 \leq i \leq |\mathbb{L}|} \delta_i^{(i,j)} = 1$$

$$(\delta_1^{(i,j)}, \ldots, \; \delta_{|\mathbb{L}|}^{(i,j)}) \in \{0,1\}^{|\mathbb{L}|}$$

$$\forall p \in P, \forall (u_i, u_j, u_k) \subset p,$$

$$(C_{dd}) \quad \mathbb{A}_{dd}.^T(\delta_1^{(i,j)}, \ldots, \delta_c^{(i,j)}, \delta_1^{(j,k)}, \ldots, \delta_c^{(j,k)}, \delta_p) \leq \mathbb{B}_{dd}, \text{ si } i < j \text{ et } j < k$$

$$(C_{di}) \quad \mathbb{A}_{di}.^T(\delta_1^{(i,j)}, \ldots, \delta_c^{(i,j)}, \delta_1^{(k,j)}, \ldots, \delta_c^{(k,j)}, \delta_p) \leq \mathbb{B}_{di}, \text{ si } i < j \text{ et } j > k$$

$$(C_{id}) \quad \mathbb{A}_{id}.^T(\delta_1^{(j,i)}, \ldots, \delta_c^{(j,i)}, \delta_1^{(j,k)}, \ldots, \delta_c^{(j,k)}, \delta_p) \leq \mathbb{B}_{id}, \text{ si } i > j \text{ et } j < k$$

$$(C_{dd}) \quad \mathbb{A}_{dd}.^T(\delta_1^{(j,i)}, \ldots, \delta_c^{(j,i)}, \delta_1^{(k,j)}, \ldots, \delta_c^{(k,j)}, \delta_p) \leq \mathbb{B}_{dd}, \text{ si } i > j \text{ et } j > k$$

$$\delta_p \in \{0,1\}$$

Les deux formulations PLNE sont possibles sous réserve de pouvoir connaître la description de l'enveloppe convexe des trois ensembles de points. La connaissance d'une des trois descriptions est d'ailleurs suffisante pour en déduire les autres descriptions par symétrie lorsque le codage est de taille maximale $c = |\mathbb{L}|$.

3.3.3 Formulation et résolution exacte de MaxTOR3 et MaxTOR

Pour étudier les problèmes MaxTOR3 et MaxTOR sous leur forme exacte, on utilise les labels suivants : $\mathbb{L}_1 = C2P$, $\mathbb{L}_2 = P2C$, $\mathbb{L}_3 = PEER$.

L'encodage choisit est :

$c = 3$, $\Lambda(C2P) = (1,0,0)$, $\Lambda(P2C) = (0,1,0)$, $\Lambda(PEER) = (0,0,1)$.

$$\mathbb{M}_{dd} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \mathbb{M}_{di} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \; \mathbb{M}_{id} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

On utilise aussi les notations suivantes :

$\delta_1^{(u,v)} = \delta_{C2P}^{(u,v)}$, $\delta_2^{(u,v)} = \delta_{P2C}^{(u,v)}$, $\delta_3^{(u,v)} = \delta_{PEER}^{(u,v)}$, $\forall (u,v), u < v$

a) Formulation MIP

L'algorithme d'énumération des équations des facettes d'un polyèdre entier reporté dans l'annexe C permet de générer efficacement les inéquations utilisées dans les contraintes (C_{vcap}).

Pour la matrice $\mathbb{M}_{dd}$ de MaxTOR3, on obtient la description en dimension 7 suivante (valable pour toutes les contraintes (C_{dd})) :
$conv(\Phi(M_{dd})) = \{x = (x_1, \ldots, x_7) \in \mathbb{R}^7 /$

$$\begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 \\ -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & -1 & 0 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ -1 \end{pmatrix} \}$$

Cela veut dire que $\forall (u, v, w) \in P_3'$, si $u < v < w$ alors :

$$(C_{dd}) \quad \begin{matrix} 1: \\ 2: \\ 3: \\ 4: \\ 5: \\ 6: \\ 7: \\ 8: \\ 9: \\ 10: \\ 11: \\ 12: \end{matrix} \begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & -1 & -1 & -1 & 0 \\ -1 & -1 & -1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \delta_{C2P}^{(u,v)} \\ \delta_{P2C}^{(u,v)} \\ \delta_{PEER}^{(u,v)} \\ \delta_{C2P}^{(v,w)} \\ \delta_{P2C}^{(v,w)} \\ \delta_{PEER}^{(v,w)} \\ \delta_{(u,v,w)} \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Les inéquations générées dans le cas $u < v < w$, numérotées de 1 à 12, peuvent être découpées en trois familles :

[1→9] inéquations de label : la variable de triplet $\delta_{(u,v,w)}$ n'apparaît dans aucune des inéquations. Chaque inéquation fournie peut donc être déduite des contraintes (C_{label}) et (C_{label}), et des contraintes $\delta_j^{(u,v)} \in \{0,1\}$, $\delta_j^{(v,w)} \in \{0,1\}$, $j \in \{1,2,3\}$.

[10→11] inéquations superflues : toute solution optimale qui vérifie la contrainte [12] $(-\delta_{C2P}^{(u,v)} - \delta_{P2C}^{(v,w)} + \delta_{(u,v,w)} \leq 0)$ vérifie aussi les deux contraintes [10] $(\delta_{C2P}^{(u,v)} \leq \delta_{(u,v,w)})$ et [11] $(\delta_{P2C}^{(v,w)} \leq \delta_{(u,v,w)})$. En effet, si $\delta_{C2P}^{(u,v)} = 1$ ou $\delta_{P2C}^{(v,w)} = 1$ alors $\delta_{(u,v,w)} = 1$ (en raison de la fonction objectif et car $\delta_{P2C}^{(v,w)} \geq 0$ et $\delta_{C2P}^{(u,v)} \geq 0$), ce qui montre que les deux contraintes sont vérifiées dans les trois cas. Sinon $\delta_{C2P}^{(u,v)} = 0$ et $\delta_{P2C}^{(v,w)} = 0$ et alors les deux contraintes sont redondantes par rapport au fait que $\delta_{(u,v,w)} \geq 0$.

D'après les remarques précédentes, la famille des 12 inéquations peut être réduite à une seule inéquation[17]. Dans notre cas $i < j < k$, on obtient[18] :

$(C_{dd}) \quad -\delta_{C2P}^{(u,v)} - \delta_{P2C}^{(v,w)} + \delta_{(u,v,w)} \leq 0,\ \forall (u,v,w) \in P_{dd}$

En calculant les ensembles $conv(\Phi(M_{di}))$ et $conv(\Phi(M_{id}))$ et en supprimant les contraintes redondantes comme précédemment, on obtient la formulation suivante du problème MaxTOR3[19] :

MaxTOR3 (formulation PLNE)

[17]Les inéquations de labels et les inéquations non nécessaires peuvent être retirées de la formulation MIP sans changer les conditions d'optimalité du problème.

[18]Intuitivement, on retrouve qu'un triplet (u,v,w) est valide *si et seulement si* u est client de v *ou* w est client de v.

[19]la déduction des contraintes (C_{di}) et (C_{id}) peut aussi s'établir grâce à la symétrie des labels.

$$\text{Maximiser} \quad \sum_{p \in P'_3} \delta_p$$

$$\forall e = \overrightarrow{(u_i, u_j)} \in E, \;\; i < j$$

$$(C_{label}) \quad \delta^{(i,j)}_{C2P} + \delta^{(i,j)}_{P2C} + \delta^{(i,j)}_{PEER} = 1$$

$$0 \leq \delta^{(i,j)}_{PEER} \leq 1$$

$$(\delta^{(i,j)}_{C2P}, \delta^{(i,j)}_{P2C}) \in \{0,1\}^2$$

$$\forall (u_i, u_j, u_k) \in P'_3,$$

$$(C_{dd}) \quad -\delta^{(i,j)}_{C2P} - \delta^{(j,k)}_{P2C} + \delta_{(u_i,u_j,u_k)} \leq 0, \text{ si } i < j < k$$

$$(C_{di}) \quad -\delta^{(i,j)}_{C2P} - \delta^{(k,j)}_{C2P} + \delta_{(u_i,u_j,u_k)} \leq 0, \text{ si } i < j \text{ et } j > k$$

$$(C_{id}) \quad -\delta^{(j,i)}_{P2C} - \delta^{(j,k)}_{P2C} + \delta_{(u_i,u_j,u_k)} \leq 0, \text{ si } i > j \text{ et } j < k$$

$$0 \leq \delta_{(u_i,u_j,u_k)} \leq 1$$

Notons que l'on peut relaxer les variables définissant les labels $PEER$. Les variables de label $PEER$ ne sont mises en jeu que dans des contraintes (C_{label}). Si les variables $\delta^{(i,j)}_{C2P}$ et $\delta^{(i,j)}_{P2C}$ sont entières, cela suffit pour que la variable $\delta^{(i,j)}_{PEER}$ soit entière grâce à la contrainte (C_{label}).

Notons aussi la relaxation continue des variables de validité des triplets d'AS. Les variables de triplets d'AS sont nécessairement entières dans une solution optimale. En effet, chaque variable de triplet $\delta_{(u,v,w)}$ sature la contrainte de domaine $\delta_{(u,v,w)} \leq 1$ et/ou la contrainte (C_{vcap}) associée qui selon les variables de label des liens (u, v) et (v, w) vaut 0, 1 ou 2.

b) Méthode de réduction hiérarchique pour MaxTOR3

L'algorithme (R_{stub}) présenté en fin de chapitre 2, permet en procédant à un pré-traitement de réduire la taille du problème (nombre de variables et nombre de contraintes) sans perdre la condition d'optimalité. Ce pré-traitement consiste à fixer à chaque itération de l'algorithme (R_{stub}) le label $C2P$ aux arcs reliant les noeuds stubs aux noeuds du coeur. Ces arcs disparaissent à chaque itération. Les arcs reliant deux noeuds du coeur qui disparaissent suite à la suppression de triplets, prennent la valeur PEER. A la première itération, les arcs reliant deux noeuds de l'ensemble Stub sont placés à la valeur $PEER$ car ils ne sont mis en jeu dans aucune contrainte (C_{vcap}) du problème.

Algorithme 3. *Réduction exacte (R_{stub}) pour MaxTOR3*

Entrée : triplets $T_{in} = \{(u,v,w) \in P'_3 / u_i < w_i\ \}$
liens $E_{in} = \{(u,v) \in E\ / u < v\}$

Répéter

$Core := \{u\ /\ \exists (v,u,w) \in T_{in}\}$

$Stub := \{u\ /\ \nexists (v,u,w) \in T_{in}\}$

$T_{out} := \{(u,v,w) \in T_{in} \cap (Core^3)\}$

$$E_{out} := \left\{ (u,v) \in\ Core^2 \bigcap E_{in} \quad / \exists w \in Core,\ \text{avec} \begin{array}{c} (u,v,w) \in T_{out} \\ \text{ou } (v,u,w) \in T_{out} \\ \text{ou } (w,u,v) \in T_{out} \\ \text{ou } (w,v,u) \in T_{out} \end{array} \right\}$$

$$\forall (u,v) \in E_{in} \backslash \{E_{out}\},\ \left\{ \begin{array}{l} \mathbb{L}[(u,v)] := C2P \text{ si } u \in Stub \text{ et } v \in Core \\ \mathbb{L}[(u,v)] := P2C \text{ si } u \in Core \text{ et } v \in Stub \\ \mathbb{L}[(u,v)] := PEER \text{ sinon} \end{array} \right.$$

$E_{in} := E_{out}$

$T_{in} := T_{out}$

Tant que $Stub \neq \varnothing$

Sortie : arcs $\mathbb{L} : \{(u,v)\} \rightarrow \{C2P, P2C, PEER\}$
l'ensemble $Core$ calculé dans la dernière itération
ensemble de triplets T_{in} et de liens E_{in}

fin de l'algorithme

Étant donné un ensemble de triplets P'_3, on peut utiliser la procédure de réduction (R_{stub}) pour obtenir une formulation plus compacte du problème MaxTOR3. Cette formulation PLNE beaucoup plus petite que le problème initial va nous servir à trouver plus rapidement la valeur optimale du nombre de triplets valides. Avec cette solution et la valeur objective à l'optimum, on peut résoudre dans un second temps le problème non-réduit en ajoutant une contrainte sur la somme du nombre de triplets égale à cette valeur optimale (cf. algorithme 7). Si nous introduisons de nouvelles contraintes au problème, cela permet de résoudre le problème complet avec la connaissance préalable de la valeur maximum admissible du nombre de triplets valides.

c) Coupes "gadget"

Nous appelons "gadget" la donnée d'une configuration spécifique d'un ensemble de triplets contigus. La description du polyèdre des solutions entières du problème est incomplète

avec nos contraintes (C_{vcap}). En effet, si on résout le problème MaxTOR3 sous sa forme relaxée (chaque variable de décision prend une valeur continue dans $[0, 1]$), la solution obtenue du programme linéaire est fractionnaire. Cette solution fractionnaire ne viole aucune des contraintes (C_{vcap}) mais demeure néanmoins une solution non réalisable du problème MaxTOR3 où les variables de décision doivent être binaires. La résolution d'un problème formulé en nombre entiers est d'autant plus efficace que la description du polyèdre des solutions admissibles est précise. En général, le plus important est de pouvoir mettre en évidence les facettes du polyèdre actives à l'optimum. Nous introduisons donc une nouvelle famille de coupes qui écartent des solutions fractionnaires du problème relaxé. Ces coupes sont appelées **"gadget"** et sont déduites de l'analyse d'un nouveau graphe G_{vcap} qui est lui-même construit à partir des contraintes (C_{vcap}). Les coupes gadget ne définissent pas forcément des facettes du polyèdre des solutions admissibles mais ces coupes ne peuvent pas être déduites simplement des contraintes (C_{vcap}) et (C_{label}). Dans un premier temps on s'intéresse aux coupes gadget sur des solutions réalisables, et dans un second temps on montre comment déterminer des contraintes gadget violées par des solutions fractionnaires.

Graphe des contraintes VCAP pour MaxTOR3

Soit un graphe des AS symétrique noté G et P'_3 un ensemble de chemins de taille 3. On définit un graphe orienté $G_{vcap} = (V_{vcap}, E_{vcap})$ appelé graphe des contraintes VCAP et déterminé en fonction de P'_3. V_{vcap} désigne l'ensemble des noeuds et E_{vcap} désigne l'ensemble des arcs de ce graphe. Chaque sommet du graphe G_{vcap} correspond à un arc du graphe G (deux sommets dans V_{vcap} pour deux arcs symétriques) :
$V_{vcap} = \left\{ a_{u,v} / \overrightarrow{(u,v)} \in G \right\}$
Chaque chemin de P'_3 introduit deux arcs dans le graphe G_{vcap} . Soit en tenant compte de tous les triplets :
$E_{vcap} = \left\{ \overrightarrow{(a_{u,v}, a_{v,w})} \,/\, (u,v,w) \in P'_3 \text{ ou } (w,v,u) \in P'_3 \right\}$
Un exemple de graphe des AS est donné figure 3.10(a) et le bi-graphe des contraintes correspondant figure 3.10(b). On peut donner une signification intuitive à chaque arc $\overrightarrow{(a_{u,v}, a_{v,w})}$ du graphe G_{vcap} : si l'arc $\overrightarrow{(u,v)}$ porte le label $P2C$ ou $PEER$, alors la contrainte C_{vcap} déduite du chemin (u,v,w) ou du chemin (w,v,u) n'est satisfaite que si le label de l'arc $\overrightarrow{(v,w)}$ est $P2C$. Le lemme suivant généralise cette observation intuitive :

Lemme 28. *Condition de validité complète des triplets d'un chemin élémentaire du graphe* G_{vcap}

Soit un chemin élémentaire $p_{vcap} = a_{u_1,u_2}, a_{u_2,u_3}, ..., a_{u_{n-1},u_n}$ *du graphe* G_{vcap} *tel que* $n \geq 3$. *Chaque arc du chemin* p_{vcap} *correspond à un élément de l'ensemble* P'_3. *On note* δ_1,...,δ_{n-2} *les variables de validité des contraintes* C_{vcap} *associées à chacun des éléments de* P'_3 *déduit de chaque arc de* p_{vcap}. *Le chemin* p_{vcap} *est supposé tel que pour touti, j éléments distincts de* $\{1, n\}$, *si* $i \neq j$ *alors* $u_i \neq u_j$. *On a :*

Si $L(\overrightarrow{(u_1, u_2)}) \in \{P2C, PEER\}$ *et* $\delta_k = 1$, $\forall k \in \{1, ..., n-2\}$

Alors $L(\overrightarrow{(u_j, u_{j+1})}) = P2C$, $\forall j \in \{2, ..., n-1\}$

Preuve :

Supposons $\forall k \in \{1, ..., n-2\}, \delta_k = 1$ et $L(\overrightarrow{(u_1, u_2)}) \in \{P2C, PEER\}$. On associe au chemin p_{vcap} du graphe G_{vcap} le chemin $p = (u_{1,}, ..., u_n)$ du graphe G. D'après l'hypothèse $\forall k \in \{1, ..., n-2\}, \delta_k = 1$, on obtient :

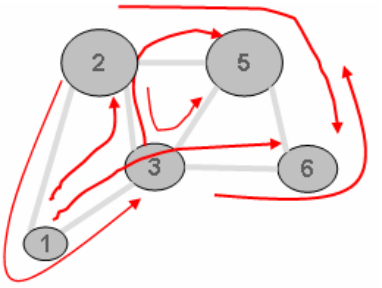
$\forall j \in \{1, ..., n-2\}$, $L(\overrightarrow{(u_j, u_{j+1})}) = C2P$ ou $L(\overrightarrow{(u_{j+1}, u_{j+2})}) = P2C$

Sachant que $L(\overrightarrow{(u_1, u_2)}) \neq C2P$, on obtient $L(\overrightarrow{(u_2, u_3)}) = P2C$. Par récurence simple :

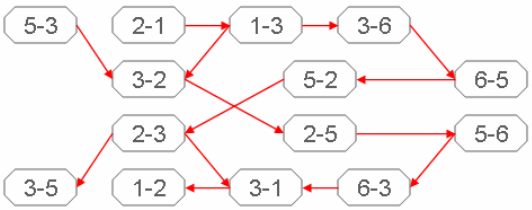
$\forall j \in \{1, ..., n-2\}, L(\overrightarrow{(u_{j+1}, u_{j+2})}) = P2C$

□

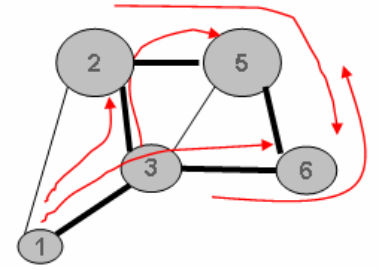
Les contraintes gadget sont une application simple du lemme précédent. Tout chemin élémentaire dans le graphe G_{vcap} entre un noeud $a_{u,v}$ et le noeud $a_{v,u}$ définit un gadget. S'il existe un chemin élémentaire $p_{vcap} = a_{u_1,u_2}, a_{u_2,u_3}, ..., a_{u_{n-1},u_2} a_{u_2,u_1}$ du graphe G_{vcap}, alors les contraintes C_{vcap} définies par p_{vcap} ne peuvent pas être toutes valides *si* $L(\overrightarrow{u_1, u_2}) \neq C2P$.



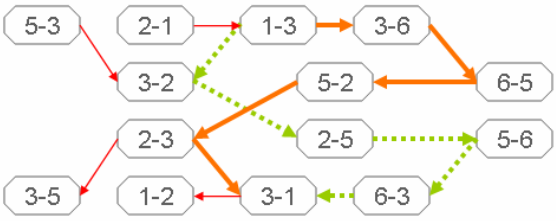
(a) graphe des AS. Le graphe est représenté sans les arcs symétriques pour simplifier la figure.



(b) graphe utilisé pour détecter un gadget



(c) graphe des AS avec les triplets du gadget. Le graphe est représenté sans les arcs symétriques pour simplifier la figure.



(d) détection d'un gadget

FIG. 3.10 – Détection d'une contrainte gadget

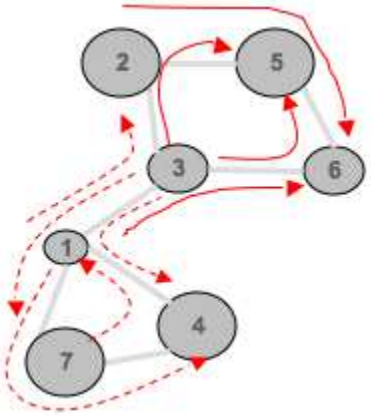
Contraintes (C_{gadget})

Supposons qu'il existe un chemin $\gamma = (u_{i_1}, u_{i_2}, \ldots, u_{i_n}, u_{i_2}u_{i_1})$ dans le graphe G_{vcap} tel que deux noeuds u_i et u_j sont distincts quelquesoit $i, j \in \{1, \ldots, n\}$ avec $i \neq j$. Alors :

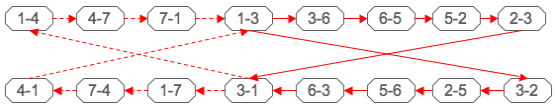
$$(C_{gadget}) \quad \sum_{1 \leq j \leq n-2} \delta_{(u_{i_j}, u_{i_{j+1}}, u_{i_{j+2}})} + \delta_{(u_{i_{n-1}}, u_{i_n}, u_{i_2})} + \delta_{(u_{i_n}, u_{i_2}, u_{i_1})} - \delta_{C2P}^{(u_{i_1}, u_{i_2})} \leq n - 1$$

La figure 3.10(c) montre un graphe des AS avec un gadget et la figure 3.10(d) montre le chemin p_{vcap} correspondant dans le bi-graphe des contraintes (il existe en fait deux chemins de la sorte en raison des symétries).

Lorsque des contraintes gadget existent dans les deux sens pour une paire origine-destination, on peut obtenir de nouvelles contraintes "chestnut" dont la configuration fut remarquée dans [15].



(a) graphe des AS avec des triplets qui forment un double gadget pour le lien entre 1 et 3. Le graphe est représenté sans les arcs symétriques pour simplifier la figure.



(b) graphe G_{vcap} utilisé pour détecter un double-gadget

FIG. 3.11 – Double contrainte gadget, appelée chestnut

Double-contrainte (C_{gadget})

Soient deux AS $u_{i_1} < u_{i_2}$ adjacents $(u_{i_1}, u_{i_2}) \in E$. Supposons qu'il existe deux chemins $\gamma_{1\to2} = (u_{i_1}, u_{i_2}, \ldots, u_{i_n}, u_{i_2}, u_{i_1})$ et $\gamma_{2\to1} = (u'_{i_1}, u'_{i_2}, u'_{i_3}, \ldots, u'_{i_{n'}}, u'_{i_2}, u'_{i_1})$ formant deux contraintes gadget. On note $u_{i_2} = u_{i_{n+1}}$ et $u_{i_1} = u_{i_{n+2}}$ et $u'_{i_1} = u_{i_2}$ et $u'_{i_2} = u_{i_1}$. On a alors les deux contraintes gadget suivantes :

$$(C_{gadget})_{1\to2} \qquad \sum_{1\le j\le n} \delta_{(u_{i_j}, u_{i_{j+1}}, u_{i_{j+2}})} - \delta_{C2P}^{(u_{i_1}, u_{i_2})} \le n - 1$$

$$(C_{gadget})_{2\to1} \qquad \sum_{1\le j\le n'} \delta_{(u'_{i_j}, u'_{i_{j+1}}, u'_{i_{j+2}})} - \delta_{C2P}^{(u'_{i_1}, u'_{i_2})} \le n' - 1$$

Sachant que $\delta_{C2P}^{(u'_{i_1}, u'_{i_2})} = \delta_{P2C}^{(u_{i_1}, u_{i_2})}$ et en utilisant (C_{label}) pour l'arc (u_{i_1}, u_{i_2}) on obtient :

$$(C_{chestnut}) \quad \sum_{1\le j\le n} \delta_{(u_{i_j}, u_{i_{j+1}}, u_{i_{j+2}})} + \sum_{1\le j\le n'} \delta_{(u'_{i_j}, u'_{i_{j+1}}, u'_{i_{j+2}})} + \delta_{PEER}^{(u_{i_1}, u_{i_2})} \le n + n' - 1$$

Dans ce cas de double gagdet, au moins un triplet est invalide parmi ceux des deux gadgets. Si l'arc (u_{i_1}, u_{i_2}) porte la label $PEER$ alors deux triplets sont au moins invalides. Voir figure 3.11.

Recherches de coupes gadget pour une solution fractionnaire

Pour trouver des contraintes gadget violées par une solution fractionnaire du problème MaxTOR3, on va chercher un plus court chemin dans le bi-graphe des contraintes avec des poids w^* fixés sur les arcs du graphe des contraintes de la façon suivante :

$$w^*(e) = 1 - \delta_{(u,v,w)} \in \{0,1\},\ \forall e = \overrightarrow{(a_{u,v}, a_{v,w})} \in E_{vcap}$$

L'algorithme suivant permet l'énumération de tels gadgets en calculant pour chaque paire d'arcs symétriques, les K-plus courts chemins dans le graphe des contraintes VCAP entre les deux noeuds correspondant aux deux arcs symétriques considérés. En calculant K-plus courts chemins dans le bi-graphe des contraintes G_{vcap} pour chaque lien inter-AS, on obtient au maximum $K * \frac{|E|}{2}$ gadgets potentiels.

Algorithme 4. *Enumération de contraintes gadgets*

Construire le graphe G_{vcap} d'après l'ensemble P_3' et $\{\delta_p,\ p \in P_3'\}$

Pour chaque $(u,v) \in E, u < v$,

Calculer $k-$plus courts chemins de $a_{u,v}$ à $a_{v,u}$ dans G_{vcap}

$(W_k^*, \Pi_k) := k\_PlusCourtsChemins(G_{vcap}, w^*, a_{u,v}, a_{v,u})$,

Notation : $\Pi_i = (w_0 = u, w_1 = v, w_1, ..., w_n, w_{n+1} = v, w_{n+2} = u),\ \forall 1 \leq i \leq k$

Notation : $W^*(\Pi_1) = Min_{\exists \Pi}\{W^*(\Pi) = \sum_{1 \leq i \leq n+1} (1 - \delta_{(w_{i-1}, w_i, w_{i+1})}), n \geq 2\} \leq W^*(\Pi_2) \leq ...$

Pour chaque $i \in \{1, \ldots k\}$

Si le chemin est élémentaire *et* $W_i^* + \delta_{C2P}^{(u,v)} < 1$, *alors* :

$$\sum_{\substack{1 \leq i \leq n+1 \\ w_{i-1} < w_{i+1}}} \delta_{(w_i, w_{i+1}, w_{i+2})} + \sum_{\substack{1 \leq i \leq n+1 \\ w_{i-1} > w_{i+1}}} \delta_{(w_{i+2}, w_{i+1}, w_i)} - \delta_{C2P}^{(u,v)} \leq n$$

fin pour chaque

fin pour chaque

fin de l'algorithme

– Remarque : l'algorithme de K-plus-courts chemins que nous avons utilisé dans l'algorithme 4 est celui proposé par Eppstein [44, 45]. Il est tout à fait possible d'utiliser d'autres algorithmes permettant d'énumérer des plus courts chemins dans un graphe.

Par définition des poids w^*, les gadgets déduits des K-plus-courts chemins dans (G_{vcap}) sont les plus favorables pour identifier une contrainte (C_{gadget}) violée pour une solution fractionnaire donnée. Le nombre de plus courts chemins K, décidant du nombre maximum de chemins calculés pour chaque arc dans la boucle de l'algorithme, permet de paramétrer la génération des contraintes gadgets pour couper une solution fractionnaire du problème MaxTOR3.

d) Recherche des cycles d'AS clients

La recherche de cycles d'AS clients dans une solution réalisable du problème consiste à trouver un cycle dans le sous-graphe G_{C2P} formé des arcs avec les labels $C2P$ uniquement puisque le graphe des AS avec les labels est symétrique. On note $G_{C2P} = (V, E_{C2P})$ le graphe tel que l'ensemble des arcs est $E_{C2P} = \{e \in E / L(e) = C2P\}$.
Pour trouver un cycle on utilise une méthode de recherche usuellement employée dans la littérature. Cette méthode consiste à rechercher un cycle qui passe par un arc donné en utilisant un algorithme de plus courts chemin sur le graphe sans cet arc. Pour exhiber plusieurs cycles, on peut itérer sur l'ensemble des arcs et énumérer k-plus-courts chemins pour chaque arc considéré. Pour traiter le cas des solutions fractionnaires, on utilise le graphe G avec un seul arc symétrique parmi les deux présents dans G. Les poids sur les arcs sont fixés à la valeur $w_{C2P} = 1 - \delta_{C2P}$ pour trouver les cycles les plus susceptibles de violer une contrainte (C_{cc}).

Algorithme 5. *Enumération de cycles d'AS clients*

Construire le graphe $G_{C2P} = (V, \varnothing)$
Pour chaque $\overrightarrow{(u,v)} \in E$, $u < v$,
ajouter l'arc $\overrightarrow{(u,v)}$ au graphe G_{C2P} et fixer $w_{C2P}\overrightarrow{(u,v)} = 1 - \delta_{C2P}^{(u,v)}$
fin pour chaque
Pour chaque $\overrightarrow{(u,v)} \in E_{C2P}$,
retirer l'arc $\overrightarrow{(u,v)}$ du graphe G_{C2P}
$(W_k, \beta_k) := k\_PlusCourtsChemins(G_{C2P}, w_{C2P}, v, u)$
Notation : $\beta_i = (v, \beta_i(2), ..., u)$, $\forall 1 \leq i \leq k$ les plus courts chemins
Notation : $W_1 = W^*(\beta_1) = Min_{\exists \beta}\{W^*(\beta) = \sum_{1 \leq j \leq |\beta|-1} (1 - \delta_{C2P}^{(\beta_1(j), \beta_1(j+1))})\} \leq W_2 \leq ...$
Pour chaque chemin β_i tel que et $1 \leq i \leq k$,
Si β_i est élémentaire *et* $W_i < \delta_{C2P}^{(u,v)}$, *alors* :
$$\sum_{1 \leq j \leq |\beta_i|-1} \delta_{C2P}^{(\beta_i(j), \beta_i(j+1))} + \delta_{C2P}^{(u,v)} \leq |\beta| - 1$$
fin si
fin pour chaque
insérer l'arc $\overrightarrow{(u,v)}$ dans le graphe G_{C2P}
fin pour chaque

fin de l'algorithme

Cycle de triplets d'AS

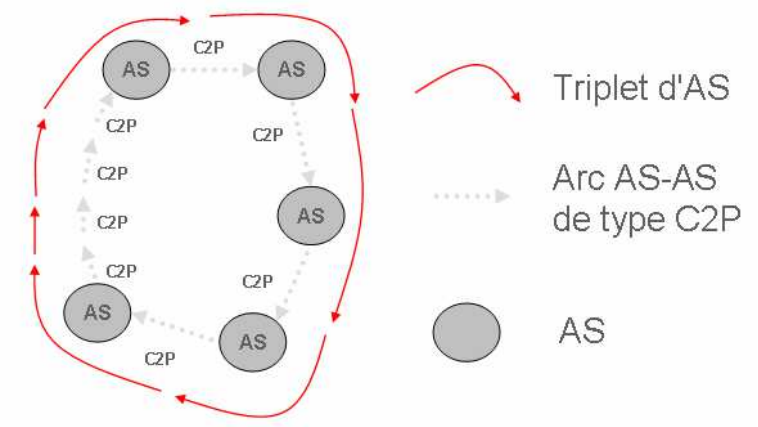
On dit qu'un ensemble de triplets forme un cycle de bout-en-bout pour un cycle élémentaire d'AS $\gamma = (u_1, u_2, \ldots, u_n, u_1)$ dans le graphe lorsque pour tout $i \in \{1, \ldots, n\}$ il existe un triplet (u_i, u_{i+1}, u_{i+2}) ou (u_{i+2}, u_{i+1}, u_i) dans P_3' (en notant $u_{n+1} = u_1$ et $u_{n+2} = u_2$). Lorsqu'il existe un tel cycle de triplets d'AS alors l'interdiction des cycles de clients oblige au moins un des triplets du cycle à être invalide (cf. figure 3.12(a)).

Définition 29. *Contraintes* (C_{cct})
Supposons qu'il existe un chemin $\gamma = (u_1, u_2, \ldots, u_n, u_1)$ *dans le graphe* G_{vcap} *tel que deux noeuds* u_i *et* u_j *sont distincts quelquesoit* $i, j \in \{1, \ldots, n\}$ *avec* $i \neq j$. *Supposons de plus qu'il existe un ensemble de triplets* $T_\gamma \subset P_3'$ *formant un cycle de bout-en-bout pour* γ. *Autrement dit en notant* $(u_{n+1} = u_1, u_{n+2} = u_2)$, *pour tout* $j \in \{1, \ldots, n\}$, *si* $i_j < i_{j+2}$ *alors* $(u_{i_j}, u_{i_{j+1}}, u_{i_{j+2}}) \in T_\gamma$ *sinon* $(u_{i_j}, u_{i_{j+1}}, u_{i_{j+2}}) \in T_\gamma$.

Si *on interdit les cycles d'AS clients,*

Et si *il existe un cycle élémentaire d'AS* γ *et un ensemble de triplets* T_γ *formant un cycle de bout-en-bout pour le cycle* γ

Alors (C_{cct}) $\displaystyle\sum_{p \in T_\gamma} \delta_p \leq |T_\gamma| - 1$



(a) Cycle de relations $C2P$ cumulé avec un cycle de triplets. Dans ce cas précis, l'interdiction de cycles de relations $C2P$ oblige au moins un triplet du cycle à être invalide.

FIG. 3.12 – Cycle de triplets

e) Résolution de MaxTOR3

Pour résoudre le problème d'inférence avec les formulations exactes, nous avons choisi d'utiliser un algorithme de "Branch & Cut". Ci-dessous on donne le schéma de principe d'un tel algorithme.

Algorithme 6. *principe de l'algorithme Branch & Cut utilisé*

Entrée : le programme linéaire en nombre entiers noté Q_0
$Pile := \{Q_0\}$
Tant que $Pile \neq \varnothing$
    Choisir le problème Q dans la pile des sous-problèmes
    Ajouter des coupes pertinentes pour le problème Q
    résoudre la relaxation linéaire du problème Q : solution s_{LP}
    Si la valeur de l'objectif est inférieure à la meilleure borne connue
        Continuer à la prochaine itération
    Sinon si la solution s_{LP} est non réalisable :
        Choisir une variable entière fractionnaire X
        $Pile := Pile \bigcup \{Q \cup \{X = 0\}\} \bigcup \{Q \cup \{X = 1\}\}$
    Sinon si la solution S_{LP} est réalisable (toutes les variables sont entières) :
        $S^* := S_{LP}$, S_{LP} est la meilleure solution réalisable connue
        Supprimer les problèmes devenus inutiles dans la pile
    fin si
fin tant que
Sortie : une solution optimale entière S^*

fin de l'algorithme

Pour inférer les accords d'interconnexion entre AS, on utilise l'algorithme 7 en exploitant à deux reprises l'algorithme de "Branch & Cut" 6.

Algorithme 7. *Inférence des accords*

Itération 1 : recherche du nombre maximal de triplets valides sur le problème réduit

Déterminer la réduction du problème initial

Déterminer les contraintes (C_{label}) et (C_{vcap}) du problème réduit

Fixer la fonction objectif $\sum_{p_{core} \in R_{stub}(P'_3)} \delta_{p_{core}}$

s^*_{red} Résoudre MaxTOR3 réduit pour connaître le nombre maximal de triplets valides

Itération 2 : résolution de MaxTOR3 amélioré

Déterminer les contraintes (C_{label}) et (C_{vcap}) du problème initial.

Fixer le nombre maximum de triplets valides avec un nouvelle contrainte linéaire

Générer les contraintes supplémentaires (C^*_{stub}) et (C_{acc}).

Fixer la fonction objectif $\sum_{p \in P'_3} \delta_p$

Résoudre MaxTOR3 en utilisant comme solution initiale s^*_{red}

fin de l'algorithme

Les contraintes gadget sont pré-calculées avant la résolution et incorporées dans la formulation par l'algorithme 6 à chaque noeud de l'arbre de branchement. Les gadget sont importants dans l'itération 1 de l'algorithme 7 pour accélérer la résolution de MaxTOR3 réduit. Dans l'itération 2 de l'algorithme 7, il est préférable de ne pas générer trop de coupes gadget en raison de la taille du problème et du graphe.

Les contraintes (C_{cc}) et (C_{cct}) sont très nombreuses et elles ne sont pas utilisés dans l'algorithme 7. Il n'est pas pratique d'énumérer toutes les contraintes dans la formulation car seules les contraintes actives à l'optimum sont utiles. Nous avons incorporé la recherche des coupes (C_{cc}) et (C_{cct}) dans le parcours de l'arbre de branchement de l'algorithme de résolution. Malheureusement la résolution à l'optimum n'a pas été possible et on ne reporte aucun résultat concernant l'utilisation de ces coupes.

– Remarque : il est possible d'ajouter une troisième itération pour maximiser le nombre d'arcs avec la valeur PEER. Cette résolution du problème PEERING-DISCOVERY est très difficile, même en disposant de la connaissance du nombre optimal de triplets valides et même en disposant des contraintes supplémentaires (C_{acc}) et (C^*_{stub}). Le temps d'exécution de cette troisième itération prend typiquement plusieurs semaines pour un PC 2Ghz avec 2Go de mémoire (lorsque les données d'entrées sont en quantité intéressante).

f) Performance de notre méthode exacte

Avant de comparer les résultats obtenus par les algorithmes de la littérature avec ceux proposés dans cette thèse, nous allons mettre en évidence les apports de notre contribution vis à vis de la résolution de MaxTOR3. On utilise pour cela volontairement une tomographie élémentaire de faible taille permettant une résolution de la forme non réduite : un jeux de données constitué d'un ensemble de tables de routage collectées le même jour (le 15 novembre 2003). Ce jeu de données n'est qu'un exemple représentatif qui met clairement en évidence les améliorations proposées dans cette thèse. Voici les formulations testées :

$COMP$: formulation exacte complète de MaxTOR3 sans réduction du nombre de contraintes superflues (contraintes $[1 \rightarrow 12]$ dans 3.3.3).

RED : formulation exacte réduite avec (R_{stub}) de MaxTOR3 sans réduction du nombre de contraintes superflues (contraintes $[1 \rightarrow 12]$ dans 3.3.3).

$COMP_{compact}$: formulation exacte de MaxTOR3 complète avec réduction du nombre de contraintes superflues (on garde seulement la contrainte [12]).

$RED_{compact}$: formulation exacte réduite avec (R_{stub}) de MaxTOR3 avec réduction du nombre de contraintes superflues : on garde seulement la contrainte [12].

$RED^{gadget}_{compact}$: formulation exacte réduite avec (R_{stub}) de MaxTOR3 avec réduction du nombre de contraintes superflues : on garde seulement la contrainte [12]. On ajoute en plus les contraintes gadget générées avec un seul plus court chemin dans l'algorithme 4 ($K = 1$).

Pour ces formulations, on reporte les grandeurs suivantes :

N_a : nombre de labels dans le programme linéaire (le nombre de variables est triple)

N_t : nombre de triplets exploités par le programme linéaire (égal au nombre de variables de triplets)

C_t : nombre de contraintes pour les triplets.

$P_{ligne}, P_{col}, P_{nz}, P_{time}$: correspondent respectivement au nombre de lignes, au nombre de colonnes supprimées pendant l'étape de pré-traitement du logiciel CPLEX, au nombre de non-zéros dans la matrice des contraintes et au temps de calcul passé dans la phase de pré-traitement.

R_{time} : temps de relaxation initial du problème (option "emphasis feasibility and optimality at presolve" de CPLEX).

	N_a	N_t	C_t	P_{lig}	P_{col}	P_{nz}
$COMP$	65626	242289	726867	731959	287416	1711299
RED	5376	13677	41031	42048	19837	98790
$COMP_{compact}$	65626	242289	242289	6016	7060	18048
$RED_{compact}$	5376	13677	13677	6016	7060	18048
$RED_{compact}^{gadget}$	5376	13677	13677	6016	7060	18048

	P_{time}(s)	R_{time}(s)	gap_{init}	gap_{sol}	IT_{sol}	IT_{opt}	$temps(s)$
$COMP$	29.25	1470.63	2.73%	0.01%	12879	13474	321773.67
RED	0.81	4.81	23.16%	0.06%	2841	8508	7424.27
$COMP_{compact}$	4.30	0.25	0.97%	0.02%	16	38	3274.70
$RED_{compact}$	0.22	0.32	23.56%	0.02%	4222	5709	471.30
$RED_{compact}^{gadget}$	0.22	0.31	23.56%	0.01%	420	480	288.00

TAB. 3.2 – Évaluation de l'efficacité des améliorations de résolution proposées pour MaxTOR3

gap_{init} : gap entre la meilleure solution entière et la pire des solutions fractionnaires lorsqu'une première solution entière est trouvée.

gap_{sol} : gap entre la solution optimale et la pire des solutions fractionnaires lorsque la solution optimale est trouvée.

IT_{opt}, IT_{sol} : Nombre d'itérations total et nombre d'itérations nécessaires pour trouver la solution optimale (sans prouver son optimalité) dans l'algorithme de Branch & Cut de CPLEX.

nb_{PEER} : nombre d'arcs de peering dans la solution optimale trouvée.

$temps$: temps total pour la résolution en comptant celui du logiciel CPLEX (prétraitement et exécution de l'algorithme de Branch & Cut), du chargement des données et de la génération des coupes gadget lorsqu'elles sont utilisées.

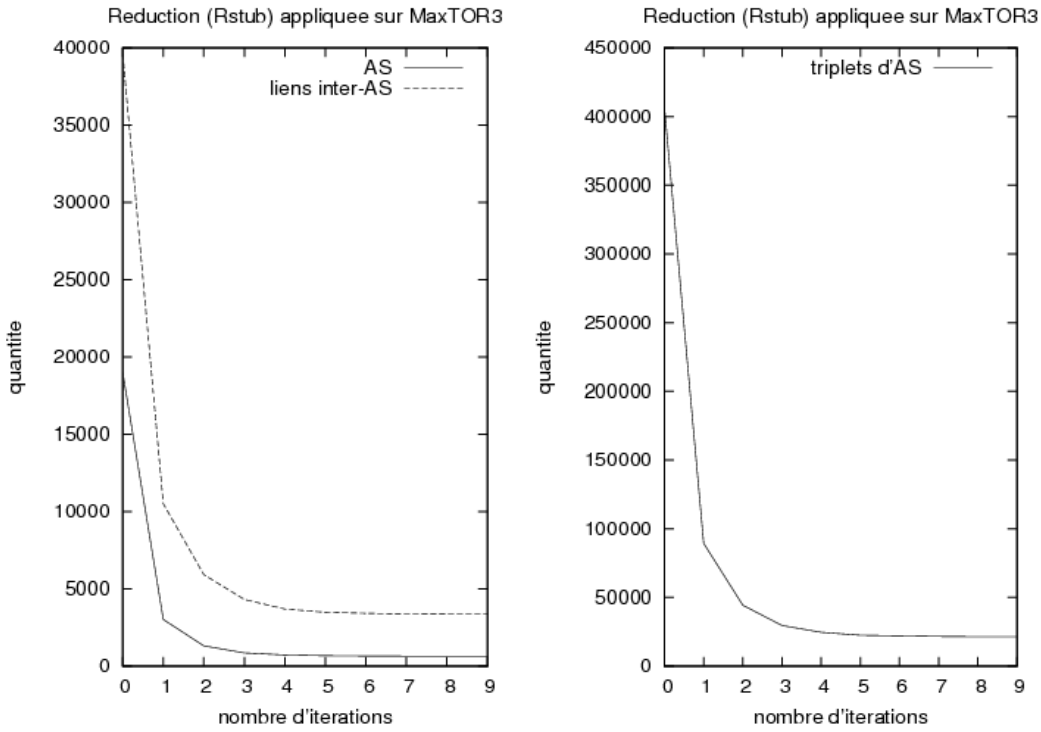


FIG. 3.13 – Effet de l'algorithme de réduction (R_{stub}) sur le jeu le test.

Signalons que le jeu de données est constitué de 613391 chemins d'AS distincts, de 242777 triplets d'AS, de 16635 AS et de 65626 arcs symétriques. Les résultats sont reportés dans le tableau 3.2. On observe que nos améliorations permettent de diviser par ~ 1000 le temps de résolution total. La réduction du problème, la suppression des contraintes inutiles et les coupes gadget sont en fait des améliorations dont les effets se multiplient.

On va maintenant évaluer la pertinence du paramétrage des coupes gadget pré-calculées pour aider la résolution. Les tests sont effectués sur un autre jeu de données (une tomographie de Janvier 2005). On utilise les tables de routage pour un seul jour comme pour l'exemple précédent. On prend différentes valeurs de K-plus-courts chemins pour l'algorithme d'énumération des coupes gadget. Le problème est résolu sous sa forme réduite et sans les contraintes inutiles. On pourra remarquer que les figures indiquent pour certaines valeurs de K un processus de résolution interrompu (pour des raisons de temps). Nous avons suivi l'effet des indicateurs suivants :

Taille du problème (figure 3.13) : dès la troisième itération, la réduction du problème diminue considérablement sa taille. Il y a environ 10 fois moins d'arcs et de triplets d'AS après la réduction.

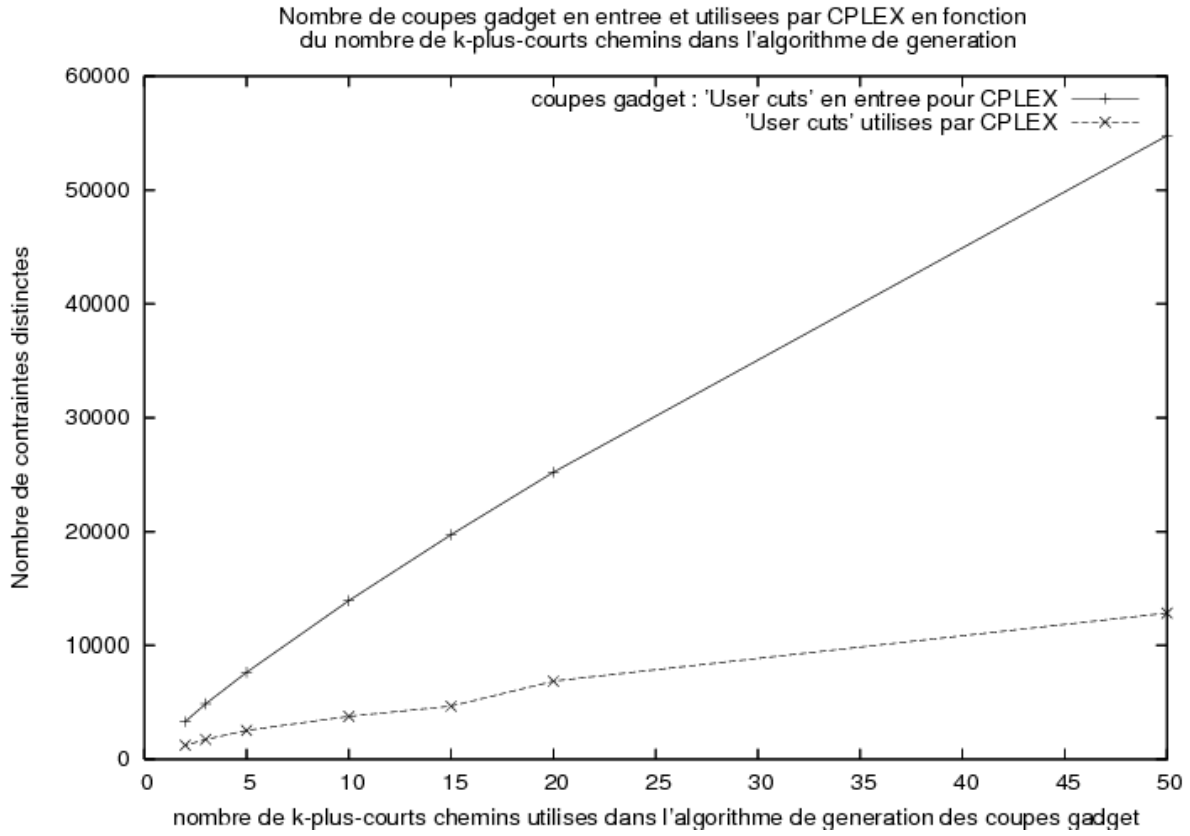


FIG. 3.14 – Nombre de coupes gadget utilisées pour le jeu de test

Coupes Gadget : sur la figure 3.14 on compare le nombre de coupes gadget générées par notre algorithme avec le nombre de "User Cuts" utilisées par l'algorithme du logiciel CPLEX. On peut remarquer qu'une part significative des coupes est toujours introduite par l'algorithme dans la formulation mais que cette part diminue quand on ajoute trop de coupes en entrée. Le logiciel CPLEX est efficace pour incorporer les coupes gadget lorsque ces dernières sont en nombre restreint.

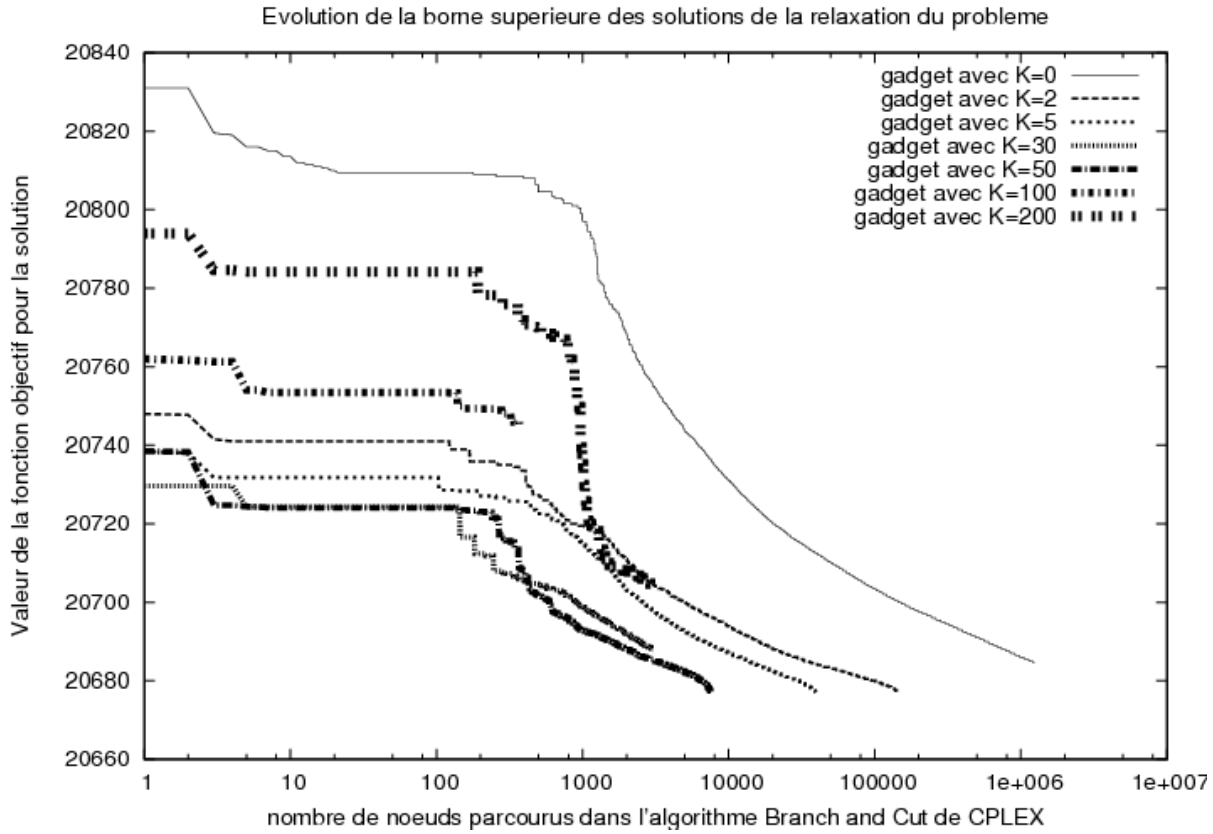


FIG. 3.15 – Évolution de la borne supérieure des solutions du problème relaxé en fonction du nombre d'itérations de CPLEX et du nombre de K-plus-courts chemins utilisés pour l'énumération des coupes gadget

Solutions fractionnaires (figure 3.15) : l'introduction des coupes gadget joue un rôle très significatif dans le nombre de noeuds explorés par l'algorithme de Branch & Cut (le nombre d'itérations est en échelle log sur la figure). Jusqu'à $k = 30$ on observe une amélioration de la borne maximum déduite après le prétraitement. Malheureusement, pour de grandes valeurs de k, le nombre de coupes générées est tel que le logiciel CPLEX ne semble pas introduire les meilleures coupes que l'on propose en entrée. Attention sur la figure 3.15, pour certaines valeurs de K la résolution a été interrompue.

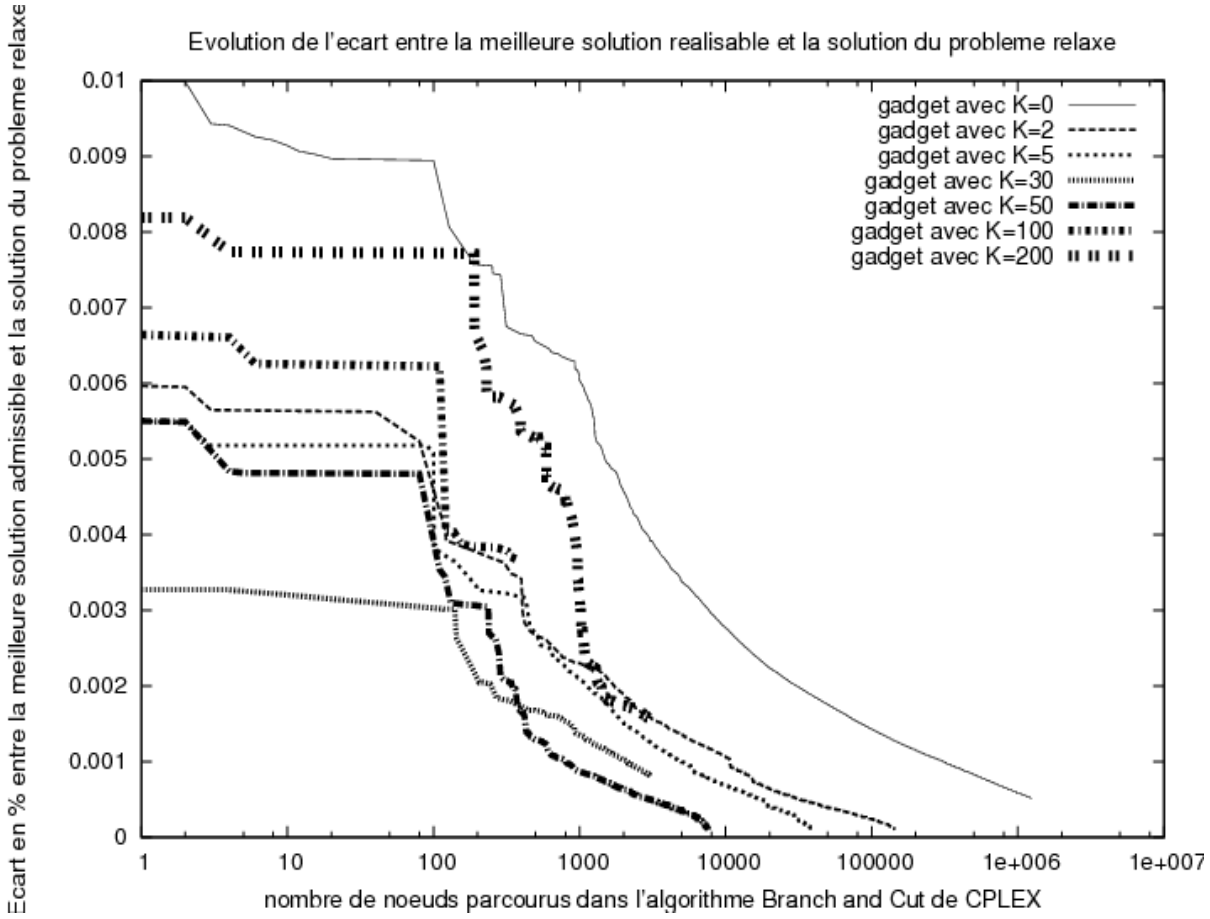


FIG. 3.16 – Évolution de la valeur du gap de résolution au cours de l'algorithme Branch & Cut en fonction du nombre d'itérations de CPLEX et de différents valeurs du nombre de K-plus-courts chemins pour l'énumération des coupes gadget

Gap d'optimalité (figure 3.16) : l'introduction des coupes gadget joue un rôle très significatif dans la taille du gap entre la meilleure solution entière et la pire des solutions fractionnaires. Attention sur la figure 3.16, pour certaines valeurs de K la résolution a été interrompue. Le meilleur compromis semble être pour la valeur $K = 30$ dans cet exemple.

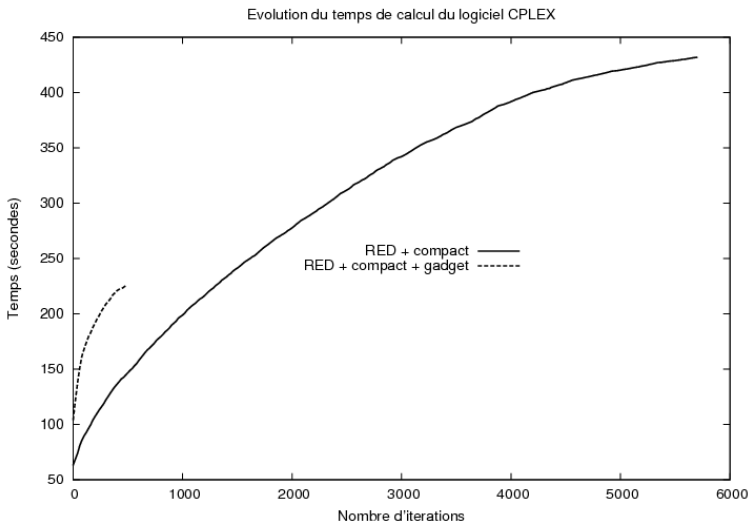


FIG. 3.17 – Effet de l'utilisation des coupes gadget sur le temps d'exécution du logiciel CPLEX.

Grâce à ces figures, on voit que les coupes gadget sont particulièrement efficaces pour améliorer le nombre de noeuds explorés dans l'arbre de l'algorithme de Branch & Cut mis en oeuvre par CPLEX. Le calcul effectué à chaque noeud est différent lorsque l'on utilise les coupes gadgets, mais le temps de traitement n'est pas beaucoup plus grand que sans ces coupes (voir figure 3.17). Les coupes gadgets sont donc très pertinentes dans la résolution du problème d'inférence des accords sous la forme d'un programme linéaire en nombre entiers. Dans le cas d'instances de grande taille, leur rôle est essentiel pour la recherche d'une solution optimale.

3.4 Résultats numériques

Les résultats numériques présentés ci-après mettent en évidence trois de nos contributions dans l'inférence des accords d'interconnexion entre AS :

Filtrage : dans le chapitre précédent différents filtres ont été proposés pour calculer une tomographie stable expurgée d'évènements temporaires qui se sont déroulés pendant la période d'observation. On montre l'impact de ces filtres dans la qualité des résultats obtenus.

Modélisation : le problème d'inférence a d'abord été modélisé sous la forme d'un MaxCSP. Puis il a été formalisé sous la forme d'un programme linéaire en nombre entiers (PLNE). Des nouvelles contraintes ont été introduites pour améliorer le réalisme des solutions : les contraintes (C_{cc}) pour la formulation MaxCSP et les contraintes (C^*_{stub}) et (C_{acc}) pour le modèle en nombre entiers.

Méthodes de résolution : nous avons résolu le problème MaxCSP avec une méta-heuristique de recherche locale avec un voisinage réduit (sans cycles d'AS clients (C_{cc})) exploré de façon guidée. Nous avons résolu le modèle en nombre entiers avec un algorithme de type "Branch & Cut" où nous avons ajouté de nouvelles coupes "gadget" et une phase de recherche de borne optimale sur une version réduite du modèle.

La résolution du problème avec les modèles PLNE n'a pas été possible en utilisant les contraintes d'interdiction de cycles d'AS clients (C_{cc}) et (C_{cct}). La lenteur de la résolution est due notamment au nombre exponentiel de cycles possibles dans le graphe des AS.

3.4.1 Données d'entrée

Nous utilisons pour nos calculs d'inférence différents ensembles de chemins, qui sont détaillés dans le tableau 3.3. On parle de chemins ou de triplets stables lorsque l'on sélectionne respectivement les chemins ou les triplets qui apparaissent pour toutes les dates utilisées dans un ensemble de données. Les jeux de données sont construits pour différentes périodes pour lesquelles on dispose des accords d'interconnexion de l'AS Opentransit (5511) du groupe France Télécom. On utilise cette connaissance pour valider partiellement les résultats.

Numéro	jeux de données	durée (jours)
1	messages de mise à jour BGP	30
2	tables de routage	1
3	tomographie avec les filtres définis dans le chapitre 2	31
4	méthode proposée par Gao et al. [190]	5
5	Sélection des chemins stables dans le temps (Dimitropoulos et al. [41])	4
6	Sélection des triplets stables dans le temps	9

Numéro du jeux	Tables BGP Route-Views	RIS	Traces "show ip bgp" Looking glass	Update BGP Route-Views & RIS
1				X
2	X	X	X	
3	X	X		
4	X	X		
5	X			
6	X			

TAB. 3.3 – Caractéristique des données d'entrée pour les tests d'inférence

3.4.2 Algorithmes utilisés

a) Algorithmes de l'état de l'art

Les algorithmes qui ne traitent que le problème MaxTOR-simple ou MaxTOR3-simple (sans prise en compte des accords de peering) ne sont pas utilisés. Pour les heuristiques permettant de résoudre le problème MaxTOR, nous avons utilisé :

GAO [60] : une méthode heuristique qui utilise les degrés successifs des AS dans les chemins pour en déduire une orientation a priori des arcs. Nous avons fixé les paramètres avec les valeurs préconisées par les auteurs.

SARK [173] : une méthode heuristique qui utilise la vision parallèle de chaque AS source pour en déduire une orientation des arcs d'après les routages de chaque AS source vers les AS. On emploie cet algorithme avec des AS sources qui ont au moins 1 000 routes chacun. Nous avons fixé les paramètres avec les valeurs par défaut.

PTE [190] : une méthode heuristique de résolution avec un pré-traitement des AS_PATH en entrée (voir jeux de données de type 4 dans le tableau 3.3). Cette méthode utilise l'algorithme *GAO* en fin de résolution.

CAIDA [91] : méthode la plus récente de la littérature. Cette méthode combine la résolution du problème MaxTOR3-simple par une relaxation SDP et la résolution

heuristique du problème *PEERING-DISCOVERY*. Les solutions ont été validées à une date précise grâce à des données réelles en provenance d'environ 30 AS.

b) Algorithmes proposés pour résoudre le problème MaxTOR3

Tout d'abord, signalons qu'une méthode de réduction et de pré-traitement des données en entrée ont été proposés pour les formulations CSP et VCAP de MaxTOR3.
Nous utilisons deux algorithmes CSP :

CSP : ce premier algorithme n'incorpore ni la recherche de cycles de clients ni les tris du voisinage $VO1$ et $VO2$.

CSP^* : celui-ci utilise les tris heuristiques $VO1$ et $VO2$ et détecte la présence des cycles d'AS clients à chaque itération.

Pour la résolution exacte des programmes linéaires en nombre entiers, nous avons reporté les résultats pour les paramétrages suivants du problème MaxTOR3 :

MIP_1 : formulation MIP minimale réduite par (R_{stub}) avec les contraintes (C_{label}) et (C_{vcap}).

MIP_2 : formulation MIP minimale avec les contraintes (C_{label}) et (C_{vcap}).

MIP_2^* : formulation MIP avec les contraintes supplémentaires (C_{acc}), (C_{stub}), (C^*_{stub}).

L'algorithme de Branch & Cut utilisé pour résoudre $MaxTOR3$ est celui du logiciel CPLEX version 10.0.

3.4.3 Comparaison des algorithmes

4 périodes de temps ont été sélectionnées : fin 2003, début 2005, début 2006 et août 2006. Pour ces quatre périodes nous avons téléchargé une solution pour l'algorithme "$CAIDA$" et on a utilisé comme référence les accords de l'AS5511 du groupe France Télécom. Pour ces quatre périodes plusieurs jeux de données ont été construits. Les détails sont reportés dans les tableaux 3.4 et 3.5. La période la plus couverte correspond au début de l'année 2005 pour laquelle on dispose de jeux de données de type 1, 2, 3, 4, 5 et 6. Le jeux de données de type 1 est particulier puisqu'il correspond à des messages de mise à jour BGP. Le graphe obtenu est plus grand, ainsi que le nombre de triplets et de chemins d'AS. Sauf exception de ce dernier jeu de données (UP_{2005} de type 1), le nombre d'AS_PATH sans répétition, le nombre de triplets et le nombre d'AS et de liens sont des indicateurs de la

Entrée	type	période
TA2003	(2)	15 Nov 2003
TA2005	(2)	29 Jan 2005
PS2005	(5)	27→31 Jan 2005
TO2005	(3)	1→31 Jan 2005
GA2005	(4)	27→31 Jan 2005
TS2005	(6)	20→29 Jan 2005
UP2005	(1)	1→31 Mar 2005
GA2006	(4)	27→31 Jan 2005
TO2006	(3)	1→31 Aou 2006

TAB. 3.4 – Périodes de temps des jeux de données utilisés

Entrée	AS_PATH sans rép. tous	AS_PATH sans rép. taille 2	Triplets	graphe des AS AS	graphe des AS liens	Visibilité AS5511 (%)	Visibilité CAIDA (%)
TA2003	613 502	7841	242793	16345	33409	87.31	93.7
TA2005	2 154 061	29134	558027	19165	48044	88.29	98.73
PS2005	1 709 333	18748	498149	19051	46536	87.8	97.95
TO2005	1822484	17593	515979	18881	47112	88.29	96.5
GA2005	1 058 422	13853	409696	19085	41396	88.29	96.92
TS2005	475053	0	475046	18902	43210	87.8	95.21
UP2005	8577708	32396	878507	19813	56899	90.24	98.63
GA2006	1241555	16617	445056	21771	46744	64.87	97.33
TO2006	2974582	26603	759103	22988	59986	87.7	98.87

TAB. 3.5 – Principales caractéristiques des jeux de données utilisés

taille d'un jeu de test qui dépendent directement de la diversité des données (le nombre d'AS sources) et de la taille de la fenêtre de temps. Pour chaque jeu de données, on a calculé le pourcentage de liens inter-AS visibles parmi ceux de la solution $CAIDA$ et des accords connus de l'AS5511 (partie droite du tableau 3.5).

De chaque jeu de données, on extrait un modèle CSP pour lequel certains arcs n'apparaissent que dans des chemins de taille 2. Ces arcs sans contrainte (C_{vcap}) sont quelquefois en très grand nombre. Ceci justifie l'utilisation de contraintes du type (C_{acc}) ou (C^*_{stub}). Les modèles CSP sont reportés dans le tableau 3.6.

On compare les solutions avec les indicateurs suivants pour la proportion des accords :

$C2P$: le nombre d'accords de transit.

$PEER$: le nombre d'accords de peering.

SIB/UNK : le nombre d'accords de sibling ou nombre d'accords UNK dans le cas de

Entrée	modèles CSP		
	variables	variables sans contrainte	contraintes (C_{vcap})
TA2003	32822	587	242305
TA2005	45339	2704	554700
PS2005	43992	2544	495917
TO2005	44581	2531	513346
GA2005	40033	1363	407735
TS2005	43210	0	472948
UP2005	54379	2517	865918
GA2006	45342	1402	442918
TA2006	56474	3512	754260

TAB. 3.6 – Modèles CSP pour les jeux de données utilisés pour les comparaisons

l'algorithme "$SARK$".

Pour la validation théorique des solutions, on utilise les indicateurs suivants :

P : le nombre de chemins valides.

T : le nombre de triplets valides (triplets avec sibling inclus).

$TSIB$: le nombre de triplets dont un ou deux des accords sont sibling SIB.

Pour comparer avec les solutions connues, nous avons utilisé les deux indicateurs suivants :

M : le pourcentage d'accords corrects pour l'AS 5511 parmi les accords visibles.

C : le pourcentage d'accords égaux à ceux de la solution $CAIDA$ pour la même période de temps.

Les valeurs de ces indicateurs pour les différents jeux de test et algorithmes de résolution sont reportés dans les deux tableaux 3.7 et 3.8 ainsi que sur les figures 3.18, 3.19, 3.20 et 3.21.

On peut d'abord remarquer que notre méthode CSP^* est quasi ex-aequo voire meilleure sur tous les jeux de données : nombre de chemins valides (indicateur P figure 3.18), nombre de triplets d'AS valides (indicateur T figure 3.19), nombre d'accords corrects pour l'AS 5511 (indicateur M figure 3.20) et nombre d'accords identiques à ceux de la solution de référence $CAIDA$. Du point de vue de l'optimisation du nombre de chemins ou de triplets valides (P ou T), on obtient des solutions plus proches de l'optimum que les différentes heuristiques de la littérature. Même si les informations en provenance de l'AS5511 ne suffisent pas pour garantir la qualité globale des solutions obtenues par nos algorithmes, on constate que pour l'AS 5511 (un des 15 plus importants) nos méthodes reconnaissent

Dataset	*PEER*	*C2P*	*SIB/UNK*	*M*	*T*	*TSIB*	*P*	*C*
MIP_1								
TO2005	3334	41146	91	71.8	99.9	1.0	99.4	90.9
TO2006	4568	51802	86	75.0	99.9	0.7	98.7	90.8
MIP_2								
TO2006	485	57349	84	75.0	99.8	0.7	87.2	88.7
MIP_2^*								
TO2005	295	44185	91	70.7	99.9	1.0	99.3	88.9
TO2006	4568	51802	86	75.0	99.9	0.7	98.7	90.8
CSP*								
TA2003	3836	29573	0	95.0	99.9	1.0	99.4	95.8
TA2005	11485	36558	0	94.5	99.9	1.0	99.2	95.6
PS2005	10944	35592	0	95.0	99.9	1.0	99.4	95.1
TO2005	11155	35957	0	94.5	99.9	1.0	99.3	95.2
GA2005	5981	35415	0	93.9	99.9	0.7	99.4	95.6
TS2005	8034	35176	0	95.6	99.9	1.1	99.9	95.7
UP2005	14488	42408	0	89.2	99.8	0.7	95.1	94.8
GA2006	6285	40459	0	95.5	99.9	0.6	99.3	95.3
TO2006	12069	44387	0	96.8	99.9	0.7	99.9	95.3
CAIDA								
TA2003	3358	29429	149	81.3	90.2	1.8	97.6	100.0
TA2005	3384	34405	170	87.7	75.7	1.3	95.2	100.0
PS2005	3384	34405	170	87.7	76.5	1.4	95.5	100.0
TO2005	4545	53253	415	87.7	75.6	1.4	95.0	100.0
GA2005	3384	34405	170	87.7	83.4	1.5	97.7	100.0
TS2005	3384	34405	170	87.7	76.6	1.5	99.6	100.0
UP2005	3384	34405	170	87.7	65.5	0.8	86.3	100.0
GA2006	4001	39697	232	81.2	85.7	0.9	97.2	100.0
TO2006	4175	42110	247	83.2	72.5	0.9	99.5	100.0
CSP								
TA2003	5012	27709	92	72.7	99.9	1.0	99.5	73.4
TA2005	5808	39425	92	72.9	99.9	1.0	99.3	88.4
PS2005	5852	38042	87	72.8	99.9	1.0	99.2	86.4
TO2005	5758	38725	88	73.5	99.9	1.0	99.5	86.9
GA2005	3322	36609	90	73.7	99.9	0.7	99.4	89.0
TS2005	5701	37422	87	73.3	99.9	1.1	99.9	87.4
UP2005	8753	45505	93	70.5	99.8	0.7	95.0	86.3
GA2006	4358	40876	82	72.5	99.9	0.6	99.3	86.7
TO2006	8027	48346	83	76.3	99.9	0.7	99.9	88.1

TAB. 3.7 – (1/2) Comparaison des algorithmes sur différents jeux de test

Dataset	*PEER*	*C2P*	*SIB/UNK*	*M*	*T*	*TSIB*	*P*	*C*
GAO								
TA2003	2034	30964	411	76.5	94.9	4.9	100.0	93.4
TA2005	7572	39429	1033	69.6	92.1	6.0	100.0	92.6
PS2005	7078	38684	774	71.1	94.1	5.0	100.0	92.9
TO2005	7215	39039	858	71.8	93.0	5.7	100.0	92.8
GA2005	3880	36916	597	75.7	93.7	5.0	100.0	94.1
TS2005	5703	36743	764	75.0	96.7	3.5	100.0	92.0
UP2005	8277	44464	4147	56.8	59.7	22.6	100.0	89.0
GA2006	3840	42219	681	81.2	92.9	5.4	100.0	94.1
TO2006	8573	46850	1033	71.8	96.2	3.4	100.0	91.6
PTE								
TA2003	19184	14205	20	54.7	70.8	1.0	26.6	51.0
TA2005	31144	16846	44	47.5	71.1	1.0	16.9	50.3
PS2005	29844	16657	35	46.1	71.0	1.1	17.9	50.3
TO2005	30283	16793	36	46.4	71.1	1.1	17.5	50.5
GA2005	24692	16673	28	48.6	73.3	0.7	26.0	49.2
TS2005	26765	16440	5	47.2	71.4	1.1	71.8	50.1
UP2005	38553	18118	217	48.1	67.6	1.2	6.6	50.5
GA2006	27905	18791	44	42.1	71.8	0.7	24.6	47.6
T02006	36155	20301	0	38.5	72.3	0.7	72.6	49.4
SARK								
TA2003	14046	19162	201	81.0	72.7	1.0	25.6	50.6
TA2005	27026	20441	576	19.3	70.6	0.9	30.2	48.5
PS2005	18169	27687	680	84.4	75.0	1.0	19.4	51.1
TO2005	24557	22286	269	84.5	70.5	1.0	30.5	46.8
GA2005	18593	22452	351	83.4	77.6	0.6	35.3	54.0
TS2005	12289	12421	18500	23.3	43.5	1.1	72.9	31.2
UP2005	30425	26156	315	21.1	72.2	0.6	17.1	51.8
GA2006	21095	25330	319	83.5	79.7	0.6	36.7	54.6
TO2006	15238	14784	26434	14.7	23.7	0.7	60.0	16.5

TAB. 3.8 – (2/2) Comparaison des algorithmes sur différents jeux de test

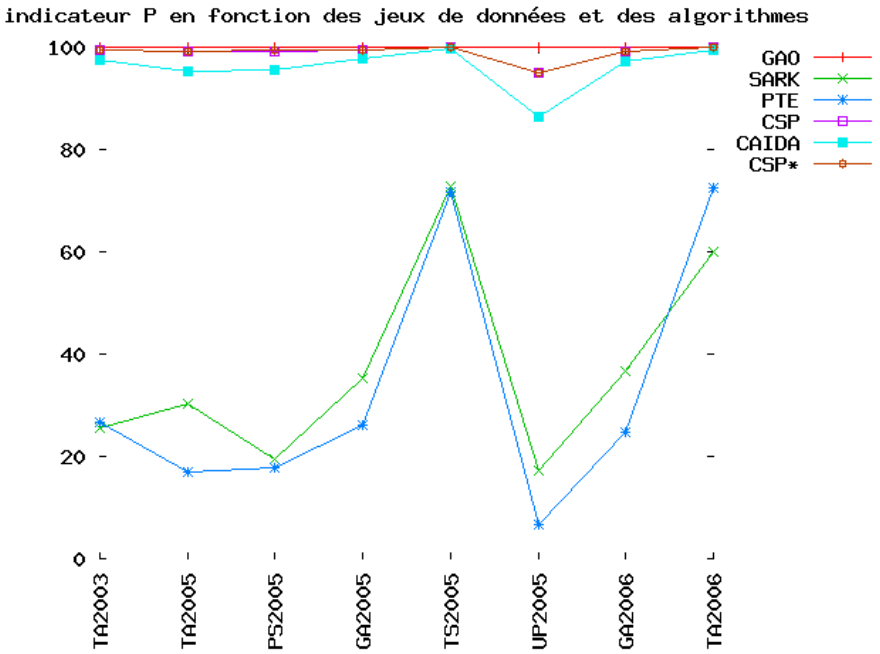


FIG. 3.18 – Valeurs de l'indicateur P pour les différentes solutions des algorithmes en fonction des jeux de test

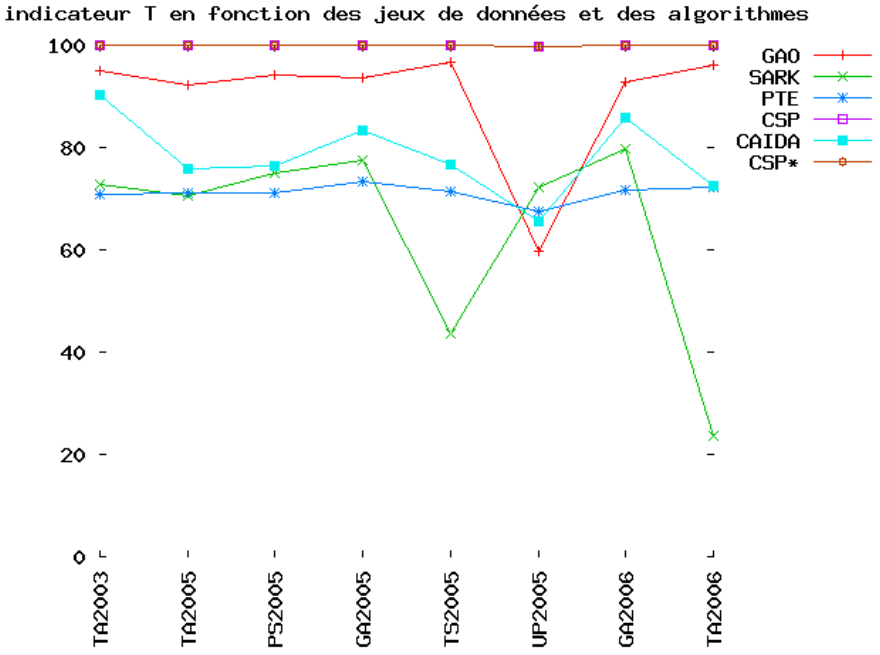


FIG. 3.19 – Valeurs de l'indicateur T pour les différentes solutions des algorithmes en fonction des jeux de test

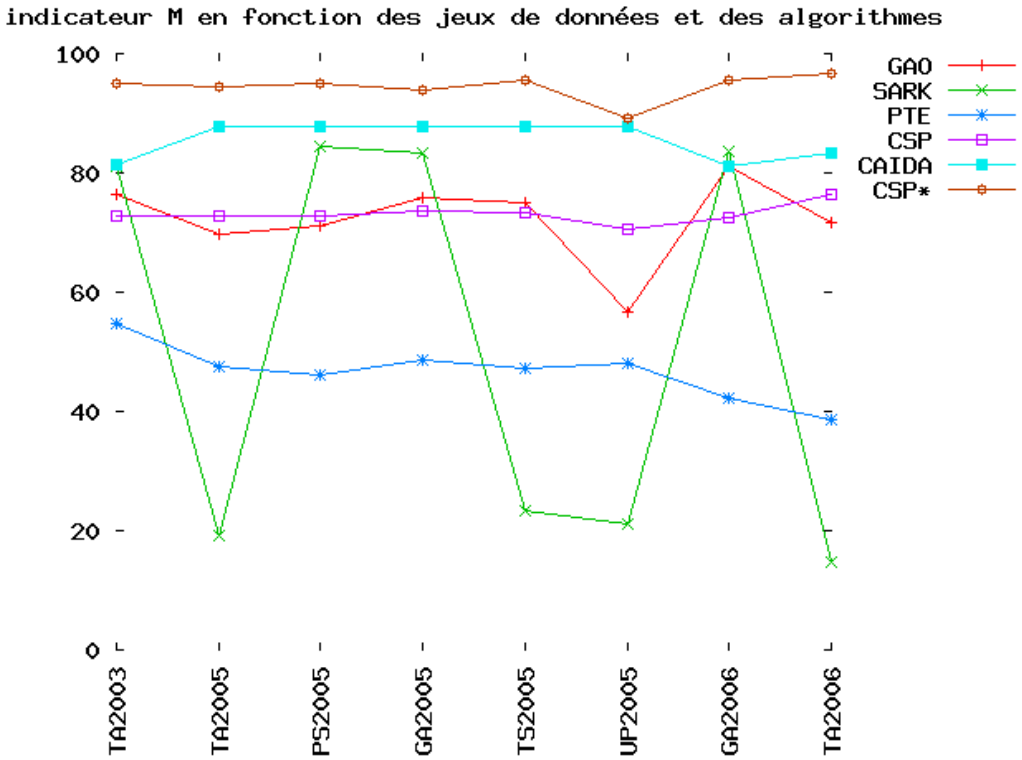


FIG. 3.20 – Valeurs de l'indicateur M pour les différentes solutions des algorithmes en fonction des jeux de test

une plus grande part des accords que les autres heuristiques de la littérature. De plus, en comparant nos solutions à celles fournies par l'algorithme $CAIDA$, on remarque que les solutions obtenues avec CSP^* sont proches à plus de 90% de celles-ci. Signalons que les solutions "$CAIDA$" ont été validées avec des informations réelles. Il semble que notre algorithme CSP^* fournit une vision proche de la réalité des interconnexions entre AS dans l'Internet [129, 130].

Pour le cas des solutions MIP (PLNE), les limitations de la modélisation du problème apparaissent : doit-on seulement maximiser le nombre de chemins ou de triplets valides ? Est-il pertinent de maximiser le nombre de liens de peering après une première optimisation qui valide les cheminements ? Finalement, on peut affirmer que les bons résultats de la méthode CSP^* sont sans doute dus à l'interdiction des cycles d'AS clients dans les solutions explorées par l'heuristique. L'interdiction de ces cycles est pour l'instant impossible à prendre en compte dans notre approche exacte. Nous n'avons en effet pas trouvé le moyen de choisir simplement les "bonnes" coupes à introduire. Il faudrait d'autres investigations pour améliorer l'intégration des coupes (C_{cc}) dans l'algorithme de résolution.

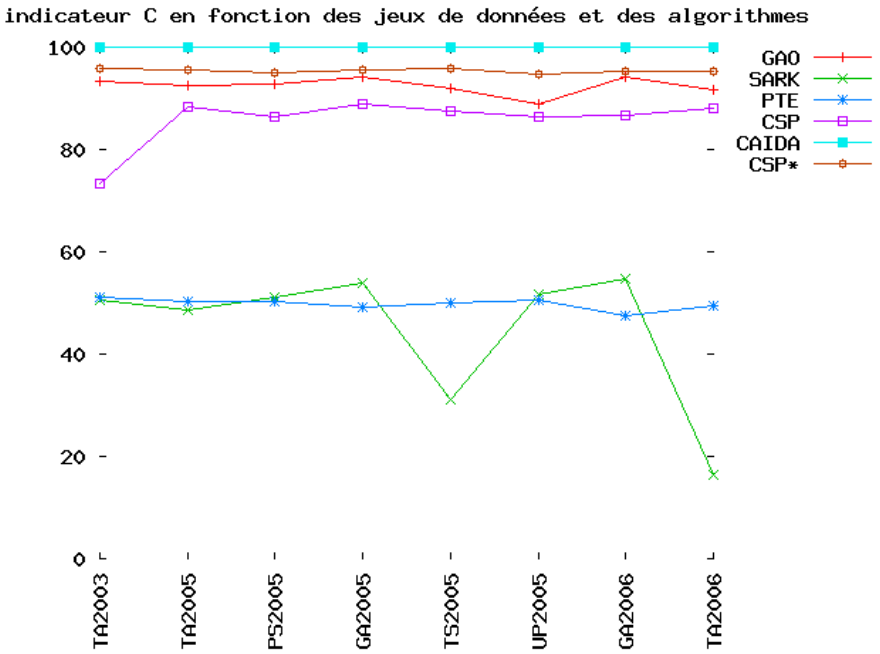


FIG. 3.21 – Valeurs de l'indicateur C pour les différentes solutions des algorithmes en fonction des jeux de test

Dans ce chapitre, on a étudié le problème d'inférence des accords d'interconnexion entre AS. Une modélisation du problème utilisant le nouveau concept des triplets d'AS est utilisée. De nouvelles contraintes pour garantir la connectivité d'un AS vers le reste de l'Internet ou pour interdire les cycles de clients ont été considérées. Une formulation MaxCSP du problème a été proposée. Celle-ci peut être résolue par une méta-heuristique de recherche tabou qui exploite les contraintes d'interdiction de cycles d'AS clients. D'après les différents indicateurs qui ont été mesurés, cette méta-heuristique fournit de meilleures solutions que celles obtenues par des algorithmes de la littérature. Le problème MaxTOR a ensuite été généralisé en un problème plus général appelé MaxVCAP. Cette généralisation est transformée en programmes linéaires en nombres entiers. La formulation PLNE obtenue pour le cas particulier du problème MaxTOR est finalement améliorée. Les améliorations ont permis de diviser par plus de 1000 le temps de résolution à l'optimum du problème. En fin de chapitre, les solutions fournies par nos algorithmes ont été confrontées à celles d'autres méthodes de la littérature. En particulier, les solutions ont été comparées avec les accords connus de l'AS 5511 exploité par le groupe France Telecom. Les calculs effectués sur de nombreux jeux de données différents, montrent également la pertinence de notre modélisation et l'efficacité de nos méthodes de résolution.

Chapitre 4

Inférence des chemins interdomaines

Dans le chapitre 2, on a montré comment « mesurer » le réseau Internet et en déduire une tomographie BGP. Chaque tomographie BGP est constituée des routes d'AS à destination de préfixes. Un processus de filtrage permet d'obtenir à partir des tomographies BGP une vision statique et idéale des routages Internet sur une période donnée de temps. La détermination de ces routes s'effectue au travers du processus suivant :

1. Mesure du réseau Internet en procédant à l'extraction d'un ensemble de routes P. Cet ensemble de routes est appelé tomographie BGP.
2. Application des filtres ap (AS publiques), pp (préfixes publiques), pt (préfixe-temps) à la tomographie P.
3. Application des filtres préfixe-erreur (pe), rm (réattribution de l'AS origine), pc (préfixes cédés d'un AS à un autre), poo (préfixes origine-oscillant) pour retirer des annonces du même préfixe par plusieurs AS. En particulier, on garde les préfixes multihomés (pm).
4. Sélection des chemins constitués des liens logiques AS-AS et des annonces stables avec les filtres lt (lien-temps) et aat (atome-annonce-temps).
5. Sélection des chemins primaires pour chaque AS source.

A partir des tomographies filtrées on extrait des matrices de politiques de routages. On note $\mathcal{M}_a$ les matrices de politique de routage d'annonce et $\mathcal{M}_t$ les matrices de politique de routage de transit. On extrait également le graphe des AS symétrique noté $G = (V, E)$. Dans le chapitre 3, les matrices de politiques de routage sont exploitées pour résoudre le problème d'inférence des accords d'interconnexion. Deux méthodes de résolution sont proposées pour déterminer l'accord d'interconnexion ($C2P$, $P2C$, SIB, $PEER$) pour

chaque arc entre deux AS du graphe G. La qualité des méthodes utilisées permet de valider beaucoup des contraintes économiques observées pour les chemins d'AS.
Dans ce chapitre, on propose une transformation de graphe permettant de s'affranchir des labels modélisant les accords d'interconnexion. Une structure algébrique est également définie pour modéliser les attributs BGP prépondérants d'une route d'AS. Ce graphe combiné avec la structure algébrique permet de calculer des routages économiquement pertinents entre deux AS quelconques, ainsi que les routages entre un AS et un NLRI BGP. Les routages calculés vérifient les deux premières étapes du processus simplifié de décision BGP. De plus, tous les chemins calculés sont économiquement valides vis-à-vis des accords d'interconnexion préalablement inférés.
Le processus de construction du modèle de graphe étendu et d'inférence de ses propriétés comporte trois grandes étapes :

1. Construction du graphe étendu en fonction des accords d'interconnexion.
2. Identification des groupes de préfixes annoncés par un même AS origine. Les groupes de préfixes sont appelés *agrégats*. Construction des noeuds préfixes et des noeuds agrégats dans le graphe étendu.
3. Validation du modèle (calcul des chemins dans le modèle et comparaison avec les chemins de la tomographie).

4.1 Un modèle de cheminement inter-AS

On propose de définir un nouveau modèle de graphe des AS pour calculer efficacement les routages BGP inter-AS. On utilise le formalisme développé dans [48] en définissant une structure algébrique E_1. A chaque arc du graphe est associé un poids issu de cette structure. Les poids des arcs peuvent être combinés pour donner le poids d'un chemin. Le poids d'un chemin composé de sous-chemins juxtaposés est égal à la combinaison des poids de chacun des sous-chemins. L'opérateur noté $\odot$ permet de concaténer les poids des chemins. L'opérateur $\oplus$ permet de comparer deux poids et de sélectionner le meilleur. Il est utilisé pour modéliser les préférences entre routes BGP. La structure algébrique $(E_1, \oplus, \odot)$ permet de calculer routages inter-AS. On propose de résoudre le calcul du meilleur chemin BGP grâce à une version modifiée de l'algorithme de Dijkstra généralisé. Cet algorithme permet d'obtenir de calculer les routages inter-AS en temps polynomial. La structure algébrique définie pour le graphe des AS est utilisée ensuite sur le graphe des AS étendu.

Cette transformation permet d'inférer des plus courts chemins BGP économiquement valides. Les contributions apportées dans cette partie sont les suivantes :

Un modèle de graphe étendu (cf 4.1.2) : il permet de modéliser les contraintes économiques entre AS.

Une structure algébrique de poids pour les arcs (cf 4.1.3) : elle caractérise la notion de préférence au sens des attributs BGP. On ne tient compte que des deux premières étapes du processus de décision BGP en accord avec les hypothèses dans 4.1.1.

Une preuve de convergence pour l'algorithme de Dijkstra généralisé et transposé qui autorise le calcul des routages inter-AS sous contraintes de préférences locales. On montre en particulier que l'algorithme fonctionne même si la structure algébrique n'est pas un dioïde[1]. L'algorithme permet de calculer les meilleurs chemins depuis "n sources" vers une destination.

4.1.1 Hypothèses et simplifications du processus de décision BGP

Un modèle topologique de graphe à l'échelle des AS ne permet de modéliser qu'un comportement global à tout l'AS. Ceci revient à agréger les sélections des routeurs BGP. Dans chaque AS, on ne peut ni distinguer les différents routeurs, ni différencier les sessions BGP multiples établies avec d'autres AS[2]. Le processus de décision simplifié d'un AS dans notre modèle correspond à une décision collective de tous les routeurs dans cet AS. Ce cas se produit en pratique, pour un NLRI destination, lorsque les valeurs *Local Pref* des routes reçues par les liaisons vers un même AS voisin sont égales et que le meilleur chemin BGP est décidé dans une des deux premières étapes du processus de décision BGP[3]. Les décisions de routage effectuées au delà de la deuxième étape du processus de décision BGP (*MED*, préférence eBGP par rapport à iBGP, métrique IGP vers le nexthop...) ne peuvent

[1]Les dioïdes sont des semi-anneaux particuliers. Les poids portés par les arcs d'un graphe et les poids des chemins dans ce même graphe peuvent former un tel type de semi-anneau lorsque les règles de comparaison entre deux poids et les règles de construction du poids d'un chemin vérifient certaines propriétés. Ces propriétés permettent d'appliquer de nombreux algorithmes de calcul de plus court chemin (Jacobi, Gauss-Seidel,...). Voir [48] pour plus de détails.

[2]Dans notre modèle de graphe, lorsque deux AS établissent plusieurs sessions eBGP entre eux, nous associons un unique lien logique entre les deux AS. On ne peut donc pas prendre en compte la préférence d'une session eBGP par rapport à une autre.

[3]Plus précisément, on doit aussi supposer qu'il n'y a pas de règle dans les politiques de routage qui modifient les imports et les exports ou qui changent la valeur de l'attribut *Local Pref* pour les routes en provenance d'un même AS avec des *AS_PATH* différents.

pas être distinguées à la granularité AS et doivent donc être considérées comme des choix de routage équivalents. En effet, certains attributs correspondent à des considérations intra-AS, comme par exemple les valeurs de *MED* pour distinguer les points de sortie vers un même AS, les valeurs *Local Pref* fixées à des valeurs différentes suivant la position du routeur... Il n'est pas possible de les prendre en compte dans notre modèle. De même, l'attribut *community* est problématique dans la mesure où sa signification peut être locale à chaque AS et correspond souvent à des marquages au niveau de chaque routeur, ou à des marquages de routes de backup (exemple pour fixer le *Local Pref* à une valeur faible). Finalement, les attributs *Local Pref* et *AS_PATH* sont les seuls que l'on peut prendre en compte dans un processus de décision BGP simplifié à l'échelle de détail des AS.

a) Simplification de l'attribut *Local Pref*

Pour chaque NLRI et pour chaque AS voisin, l'attribut *Local Pref* peut être différent. On suppose dans notre modèle qu'un AS n'attribue qu'une seule valeur de *Local Pref* par AS voisin[4] et commune à tous les NLRI. Cependant les valeurs des attributs *Local Pref* utilisées sont propres à chaque AS et donc inconnues à l'extérieur de celui-ci. On fera donc les hypothèses simplificatrices suivantes :

– une route BGP provenant d'un AS client a un attribut *Local Pref* plus élevé qu'un route provenant d'un AS en peering.
– une route BGP provenant d'un AS en peering a un attribut *Local Pref* plus élevé qu'une route provenant d'un AS fournisseur.

Cette simplification, couramment adoptée dans la littérature (voir par exemple [125]), semble bien correspondre à la réalité économique des politiques de routage de chaque AS. D'un point de vue économique, ces hypothèses sont cohérentes car un AS cherche à maximiser ses revenus. D'autre part, comme montré par [63] cette configuration mène en théorie à des routages BGP stables. Il est possible d'inférer les accords d'interconnexion et donc d'en déduire les Local Pref associés. On attribuera trois valeurs arbitraires de *Local Pref* (300 pour les liens AS-AS vers les clients, 200 pour les liens vers les peer et 100 les liens vers les fournisseurs). Mais ces valeurs ne permettent pas de différencier deux fournisseurs, deux clients ou deux peers.

[4]Il est d'usage de n'attribuer qu'une seule valeur de *Local Pref* pour les routes envoyées par un AS voisin (cf. [25] par exemple). Si certaines valeurs de *Local Pref* sont spécifiques pour certaines routes ou certains préfixes, il ne sera pas possible de les prendre en compte.

b) Simplification de l'attribut *AS_PATH*

Dans le protocole BGP, l'attribut AS_PATH permet d'évaluer le nombre d'AS à traverser nécessaires pour joindre un NLRI destination. En pratique, l'attribut *AS_PATH* n'est pas exactement le chemin d'AS parcouru par le message BGP puisqu'un AS est libre de répéter[5] son numéro d'AS autant de fois qu'il le souhaite dans les *AS_PATH* envoyés aux AS voisins[6]. Les cas d'*AS Set* (voir explication dans 1.2.1, page 18) sont difficiles à prendre en compte car la manière dont les agrégations sont décidées est difficile à caractériser.
Il est difficile de savoir dans quelle mesure un routeur va répéter son numéro d'AS ou non dans l'attribut *AS_PATH*. De plus, dans notre modèle un lien logique entre deux AS représente l'ensemble des sessions eBGP entre eux deux. Or, la même route peut être annoncée avec des motifs de répétition différents (*AS_PATH* différents) sur ces sessions eBGP. Ce cas sera négligé dans notre modèle d'autant plus qu'il correspond en général à des configurations de routes de secours.

Hypothèse : on suppose qu'un AS ne se répète que d'une seule façon pour chaque lien logique avec un autre AS. Cette valeur de répétition (par défaut 1) est appeleé poids de répétition du lien.

4.1.2 Règles économiques d'interconnexion

On supposera que les règles économiques explicitées dans le chapitre 1 sont toujours respectées en pratique. On néglige les erreurs de configuration d'un routeur et les situations de pannes conduisant à des routages particuliers. On ne tient pas compte non plus des accords d'interconnexion négociés par préfixe ou par plaque géographique (accords hybrides peering/transit). En effet, il n'est pas possible d'inférer les interconnexions entre AS à un tel niveau de détail.
Une première approche pour prendre en compte les règles économiques consiste à développer des algorithmes spécifiques pour chaque besoin de calcul de cheminement (plus courts chemins, k-plus courts chemins, chemins respectant des critères prédéfinis...). Cela nous semble maladroit car la résolution de chaque nouveau problème est fastidieuse. Une seconde approche, consiste à définir une structure algébrique de poids sur le graphe prenant en compte les accords supportés par les arcs. Mettre bout-à-bout deux chemins revient à

[5]Les répétitions d'AS, aussi appelées AS Path prepending, sont utilisées pour influencer les décisions BGP des autres AS.
[6]Il peut ainsi pénaliser certaines routes.

utiliser l'opérateur de concaténation sur les poids de ces deux chemins. Un chemin avec une vallée est alors représenté avec le poids maximum. Avec cette approche, la structure algébrique perd les propriétés de distributivité à droite et à gauche. Cette propriété est essentielle dans l'utilisation d'algorithmes existants (voir plus loin pour plus de détails concernant les opérateurs de l'espace des poids). Il faut alors calculer les chemins entre beaucoup de paires origine-destination pour calculer les chemins vers une seule destination ou depuis une seule source.

La notion de chemins sans vallée est traitée grâce à une transformation du graphe décrite dans c). Le calcul de chemins sans vallée dans le graphe des AS se ramène au calcul de chemins usuels dans le graphe étendu des AS. Ce graphe permet d'utiliser les algorithmes de calcul de cheminements utilisés dans les graphes classiques. Les chemins ainsi construits sont sans vallée par construction.

a) Le graphe des AS

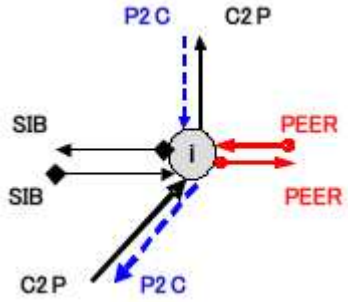
On définit le graphe des AS $G = (V, E)$ tel que chaque AS est représenté par un noeud. A chaque lien logique entre deux AS u et v, on associe les deux arcs symétriques $\overrightarrow{(u, v)}$ et $\overrightarrow{(v, u)}$ du graphe G. Chaque accord d'interconnexion entre deux AS u et v est représenté par les deux labels symétriques des arcs $\overrightarrow{(u, v)}$ et $\overrightarrow{(v, u)}$. Un exemple est représenté sur la figure 4.1(b).

b) Premier graphe étendu simple des AS

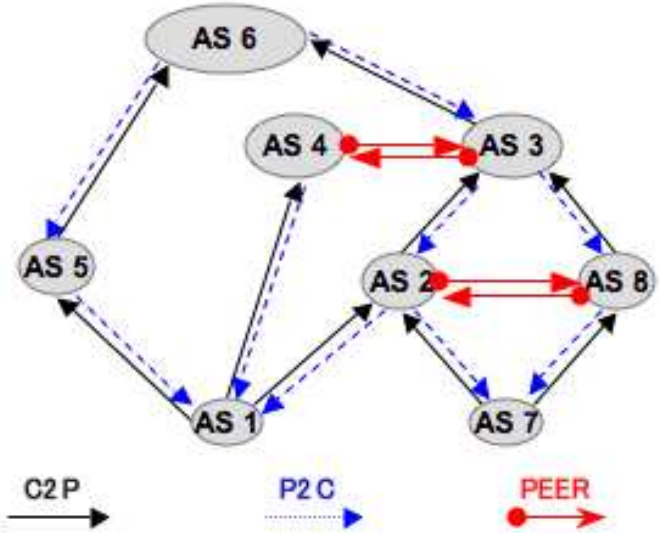
On note $G_e = (V_e, E_e)$ le graphe étendu simple des AS déduit du graphe G. A chaque noeud u du graphe G, on associe un noeud dans le graphe G_e pour chaque type d'accord d'interconnexion et chaque sens de la relation (entrée et sortie). Chaque noeud u du graphe G est transformé en un ensemble de noeuds $\{u_{e,1}(u), \ldots, u_{e,2*|\mathbb{L}|}(u)\}$. Chaque accord ($C2P$ par exemple) se traduit par deux noeuds ($C2P_{in}$ et $C2P_{out}$). On ajoute un arc entre un noeud d'accord $\mathbb{L}_i$ de l'entrée d'un AS avec un noeud d'accord $\mathbb{L}_j$ de la sortie du même AS si la succession de ces deux accords est économiquement valide[7]. Ensuite, on associe à chaque arc de G un arc dans G_e en reliant le noeud de sortie de l'AS source et le noeud d'entrée de l'AS destination qui correspondent tous les deux à l'accord sur l'arc.

[7]exemple : le noeud $PEER_{in}$ d'un AS est relié au noeud $P2C_{out}$ mais pas aux noeuds $C2P_{out}$ et $PEER_{out}$ en accord avec la matrice de compatibilité des accords successifs.

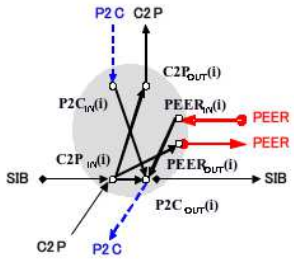
La figure 4.1 montre un exemple de transformation[8].



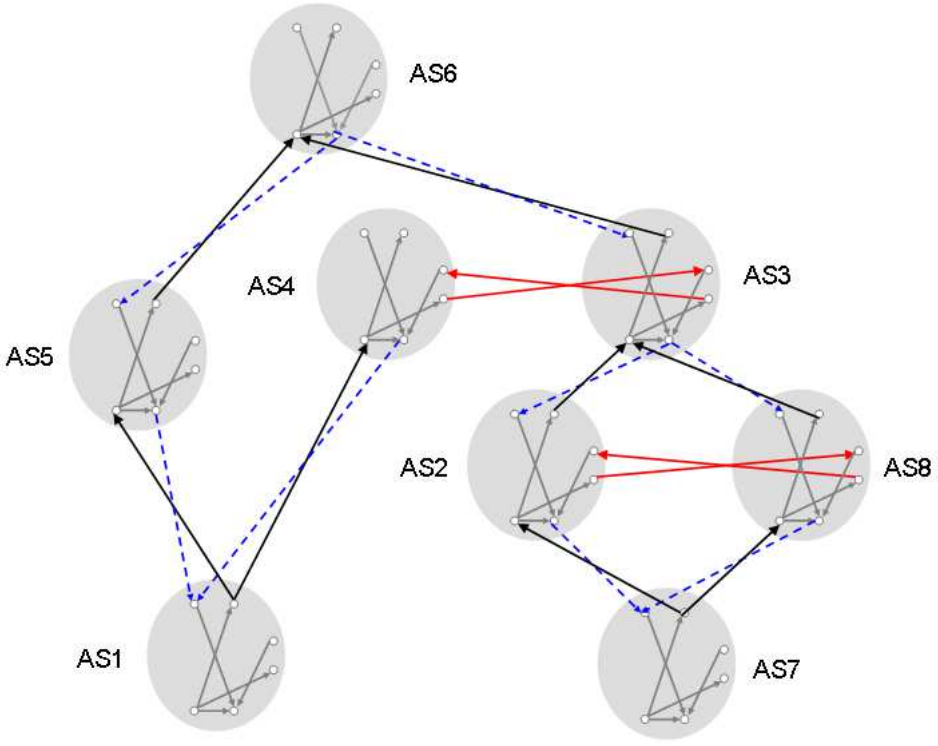
(a) un noeud du graphe des AS



(b) exemple de graphe des AS



(c) transformation d'un AS dans le graphe étendu simple



(d) graphe étendu simple des AS pour le graphe de la figure 4.1(b)

FIG. 4.1 – Exemple de transformation d'un graphe des AS en graphe étendu simple des AS

[8]L'accord de sibling, absent des figure 4.1(b) et 4.1(d), sera représenté par le raccordement des noeuds étendu $C2P_{out}$ au noeud étendu $C2P_{in}$.

Lemme 30. *il existe une bijection entre l'ensemble des chemins sans vallée dans un graphe des AS labelisé par les types d'accords d'interconnexion et l'ensemble des chemins topologiques du graphe étendu simple des AS entre les noeuds $C2P_{in}$ et les neuds $P2C_{out}$ de chaque AS.*

Preuve : soit $G = (V, E)$ un graphe des AS et l'ensemble des accords d'interconnexions associés. Soit $G_e = (V_e, E_e)$ le graphe étendu simple associé.

(sens $\Rightarrow$) *Soit un chemin du graphe étendu $(u_1, u_2, ..., u_n)$ avec $u_1 = C2P_{in}(AS_{source})$ et $u_n = P2C_{out}(AS_{dest})$. Montrons qu'il correspond à un unique chemin sans vallée du graphe des AS.*

Pour chaque arc $\overrightarrow{(u_i, u_{i+1})}, i = 1 \ldots n-1$ tels que u_i correspond dépend de AS_x et u_j dépend de AS_y, on associe un arc $e = \overrightarrow{(AS_x, AS_y)}$ du graphe des AS si $x \neq y$. Chaque sommet du graphe étendu labelisé "in" n'a que des arcs sortants vers des sommets "out" du même AS et réciproquement chaque sommet "out" n'a que des arcs sortants vers un sommet "in" d'un autre AS avec lequel un lien existe dans le graphe des AS. La séquence d'arc qui résulte est un chemin du graphe G par construction. Le chemin correspondant est sans vallée du fait de la topologie des noeuds internes à chaque AS.

(sens $\Leftarrow$) *Soit un chemin AS_1, ..., AS_n du graphe des AS. Ce chemin correspond à un unique chemin du graphe étendu entre le noeud $C2P_{in}(AS_1)$ et le noeud $P2C_{out}(AS_n)$ par construction du graphe étendu.*

□

Le graphe étendu comporte six fois plus de noeuds que le graphe des AS ($|V_e| = 6 * |V|$). De plus, le nombre d'arcs est tel que : $|E_e| = |E| + 6 \times |V|$. On constate que cette transformation est coûteuse en terme de taille et pour la complexité. Cela risque de se ressentir sur les temps (rappelons que le graphe des AS comporte plusieurs dizaines de milliers d'AS). L'intérêt de cette approche est qu'elle peut être reproduite pour des problèmes de cheminement avec d'autres contraintes de succession des labels sur les arcs d'un chemin (voir les problèmes VCAP dans 3.3.1) .

c) Le graphe étendu des AS (agrégation du graphe étendu simple des AS)

Considérons la matrice des contraintes dans la figure 4.2 correspondant aux contraintes de succession entre les accords d'interconnexion. On remarque des possibilités d'agréger des contraintes de succession entre groupes de labels. Une agrégation des noeuds du

graphe étendu est possible sans remettre en cause les propriétés du graphe étendu. La transformation d'un noeud peut aussi être interprétée au travers des règles suivantes :

1. les routes en provenance de fournisseurs et de peers ne sont transmises qu'à des clients,
2. seules les routes en provenance des clients peuvent être transmises aux peers et aux fournisseurs,
3. les routes en provenance de clients peuvent être transmises à d'autres clients.

Les contraintes valides pour des groupes des labels correspondent à des regroupements[9] de « 1 » dans la matrice des contraintes de succession. L'agrégation de la matrice permet de définir un graphe étendu avec seulement deux noeuds pour chaque AS définis tel que :

Noeud 0 Les arcs entrants pour ce noeud sont de type $C2P$ et les arcs sortants sont de type $PEER$ ou $C2P$ (correpond à deux lignes de la matrice de contraintes de succession d'arcs).

Noeud 1 Les arcs entrants pour ce noeud sont de type $P2C$ ou $PEER$ et les arcs sortants sont de type $P2C$ (correspond à deux colonnes de la matrice de contraintes de succession d'arcs).

De plus, chaque noeud 0 est relié au noeud 1 du même AS (voir figure 4.3). Le noeud 0 d'un AS regroupe les noeuds $C2P_{in}$, $C2P_{out}$ et $PEER_{out}$ du graphe étendu simple, et le noeud 1 regroupe les noeuds $P2C_{out}$, $P2C_{in}$ et $PEER_{in}$. On utilise la même notation $G_e = (V_e, E_e)$ pour ce graphe étendu que pour le graphe étendu simple car on ne référence plus ce dernier dans la suite.

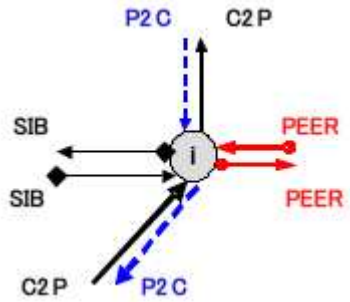
[9]Le lecteur peut garder à l'esprit que pour un autre problème de type VCAP (où une matrice de contraintes de succession d'arcs est définie) on pourrait aussi agréger des lignes et des colonnes de la matrice pour former un graphe étendu avec moins de noeuds et d'arcs que le graphe étendu simple.

Opérateur de Sortie → ↓Opérateur d'entrée	Fournisseur (C2P)	Client (P2C)	Peering (PEER)
Fournisseur (P2C)	0	1	0
Client (C2P)	1	1	1
Peering (PEER)	0	1	0

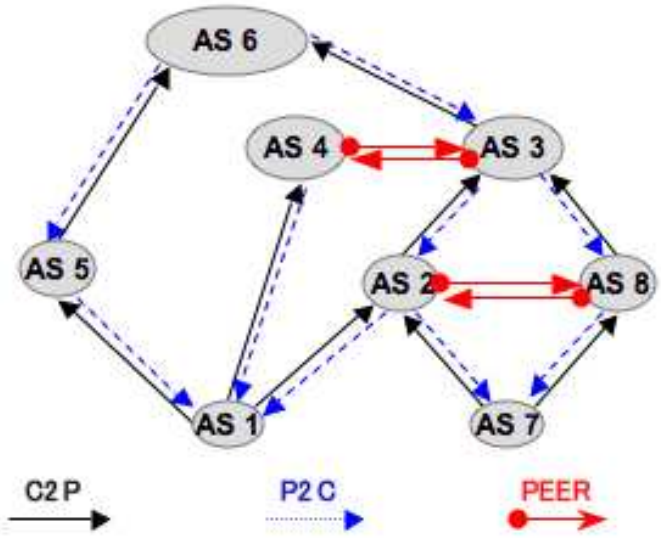
(a) La valeur 1 indique une paire d'accords dont la succession est valide. La valeur 0 indique une succession invalide de deux accords.

FIG. 4.2 – Matrice de compatibilité exprimant les contraintes économiques sur la succession des labels sur un chemin

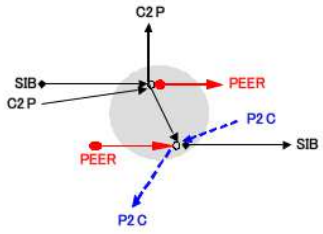
La complexité du graphe G_e est telle que : $|V_e| = 2 \times |V|$ et $|E_e| = |E| + |V|$.



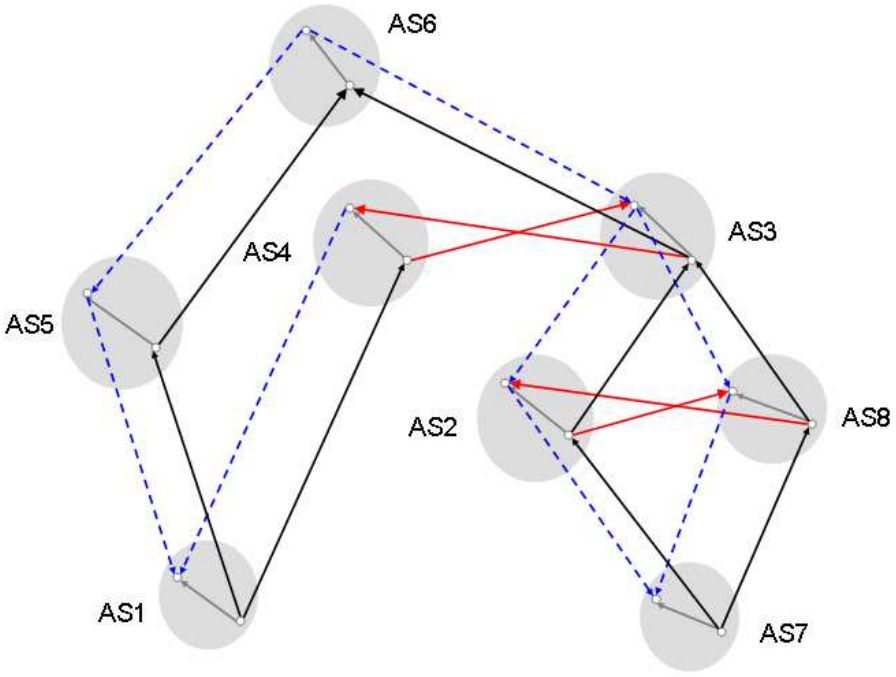
(a) un noeud du graphe des AS



(b) graphe des AS



(c) un noeud du graphe étendu des AS



(d) graphe étendu simple des AS

FIG. 4.3 – Transformation du graphe des AS en graphe étendu des AS

Lemme 31. *il existe une bijection entre l'ensemble des chemins sans vallée dans un graphe des AS marqué des accords d'interconnexion et l'ensemble des chemins du graphe étendu des AS entre tous les noeuds* 0 *et tous les noeuds* 1*.*

Preuve : La preuve est immédiate après la transformation topologique du graphe. □

On retient le dernier graphe présenté afin de calculer des chemins AS-AS valides.

4.1.3 Modèle de poids BGP simplifiés

On présente une structure algébrique permettant de modéliser les deux premières étapes simplifiées du processus de décision BGP.

a) Définitions et notations

Soit le graphe inter-AS orienté symétrique $G = (V, E)$. Pour deux noeuds du graphe AS_i et AS_j dans V, on note $P^k_{AS_i,AS_j}$ l'ensemble des chemins du graphe G avec $k+1$ sommets entre le noeud AS_i et le noeud AS_j. On note $P^k_{AS_i,AS_j}(0)$ l'ensemble des chemins élémentaires (n'empruntant pas un sommet plus d'une fois) du graphe G entre le noeud AS_i et le noeud AS_j. De plus $P_{AS_i,AS_j} = \bigcup_{k\in\mathbb{N}} P^k_{AS_i,AS_j}$ (respectivement $P_{AS_i,AS_j}(0) = \bigcup_{k\in\mathbb{N}} P^k_{AS_i,AS_j}(0)$) donne l'ensemble des chemins (respectivement chemins élémentaires) du graphe G liant le noeud AS_i au noeud AS_j. On appelera également $P^{(k)}_{AS_i,AS_j} = \bigcup_{n\in\mathbb{N}/n\leq k} P^n_{AS_i,AS_j}$ (respectivement $P^{(k)}_{AS_i,AS_j}(0) = \bigcup_{n\in\mathbb{N}/n\leq k} P^n_{AS_i,AS_j}(0)$) l'ensemble des chemins (respectivement chemins élémentaires) dans le graphe entre le noeud AS_i et le noeud AS_j.

On définit un espace des poids BGP simplifiés pour modéliser la sélection des routes par les routeurs de chaque AS. A chaque arc $\overrightarrow{(AS_i, AS_j)}$, on associe un poids $w(\overrightarrow{AS_i, AS_j})$ qui correspond à la fois au *Local Pref* de AS_i pour AS_j et à un nombre de répétitions. La métrique de répétition, correspond au nombre minimal de répétition d'AS_j dans les attributs AS_PATH des messages BGP transmis vers AS_i.

On note E_1 l'ensemble des poids BGP simplifiés. Chaque poids est formé de deux composantes. La première composante caractérise le Local Pref et la seconde composante caractérise le nombre de répétitions. On suppose qu'une route vide a un *Local Pref* infini noté ∞. On suppose également que le poids associé à l'absence d'une route a un Local Pref nul et une métrique de répétition ∞. On accepte la valeur ∞ sur la deuxième composante, seulement lorsque la première composante est nulle. Plus formellement, si note $LP = \mathbb{N}^+$ l'ensemble des valeurs possibles de *Local Pref* et $ASP = \mathbb{N}^+ \bigcup \{+\infty\}$ l'ensemble des valeurs possibles de répétition. Alors :

$$E_1 = LP \times ASP \backslash \{(x, \infty) / x \neq 0\}$$

On munit E_1 des deux lois internes $\oplus$ et $\otimes$ suivantes :

$$\begin{array}{lcccl} \oplus & : & E_1 \times E_1 & \longrightarrow & E_1 \\ & & (l_1,r_1),(l_2,r_2) & \mapsto & (l_3,r_3) = \begin{cases} (l_1,r_1)\, si\, (\infty \geq l_1 > l_2)\, ou\, (l_1 = l_2\, et\, r_1 \leq r_2) \\ ou \\ (l_2,r_2)\, sinon \end{cases} \end{array}$$

$$\begin{array}{lcccl} \odot & : & E_1 \times E_1 & \longrightarrow & E_1 \\ & & (l_1,r_1),(l_2,r_2) & \mapsto & (l_3,r_3) = \begin{cases} (0,\infty) \text{ si } r_2 = \infty \text{ ou } r_1 = \infty \\ \text{ ou} \\ (l_1, r_1 + r_2) \text{ sinon} \end{cases} \end{array}$$

Le choix de la loi $\oplus$ correspond au meilleur poids BGP simplifié parmi les deux arguments. La loi $\odot$ correspond à la propagation d'une route BGP. On pose $\varepsilon = (0,\infty)$ et $e = (\infty, 0)$. La valeur e correspond à un poids nul, et la valeur ε correspond à un poids infini. Soit la relation binaire $\leq$ définie sur E_1 comme :

$$\forall w_1, w_2 \in E_1,\ w_1 \leq w_2 \Leftrightarrow \exists w_3 \in E_1 / \ w_1 \oplus w_3 = w_2$$

La relation $\leq$ permet de traduire la préférence découlant du processus de décision BGP simplifié expliqué dans 4.1.1. Une route de poids w^* est préferée au sens de notre processus de décision BGP simplifié si et seulement si w^* est maximal au sens de la relation $\leq$.
On note $\preceq$ la relation binaire suivante :

$$\forall w_1, w_2 \in E_1,\ w_1 \preceq w_2 \Leftrightarrow \exists w_3 \in E_1 / \ w_1 \odot w_3 = w_2$$

b) Propriétés de la structure algébrique $(E_1, \oplus, \odot)$

Lemme 32. *propriétés des opérateurs $\oplus$ et $\odot$*

1. *la loi $\oplus$ est une loi interne sélective et idempotente.*
2. *la loi $\oplus$ est commutative*
3. *la loi $\oplus$ est associative*
4. *la relation $\leq$ est réflexive et transitive*
5. *la relation $\leq$ est antisymétrique (ordre commutatif)*
6. *ε est élément neutre de la loi interne $\oplus$*
7. *$\forall\, w \in E_1,\ e \oplus w = w \oplus e = e$*
8. *la relation $\preceq$ est associative*
9. *la relation $\preceq$ est réflexive et transitive*
10. *la loi $\odot$ est une loi interne non commutative*
11. *e est élement neutre à droite de la loi interne $\odot$*
12. *ε est absorbant pour la loi $\odot$*
13. *la loi $\odot$ est distributive à droite relativement à la loi $\oplus$*

Preuve : On sépare les éléments de preuve du lemme en plusieurs points ci-dessous :

Loi $\oplus$: la loi $\oplus$ sélectionne l'argument dont la première composante est maximum, ou en cas d'égalité, l'argument dont la deuxième composante est minimum. Ce choix lexicographique correspond à un ordre total. Les remarques précédentes prouvent les points 1, 2, 3, 4, 5. On vérifie aussi que l'élément $\varepsilon = (0, \infty)$ est neutre pour la loi $\oplus$ puisque sa première composante est minimum et sa deuxième composante est maximum. Ceci prouve le point 6. L'élément $e = (\infty, 0)$ est l'élément le plus grand au sens de la relation $\leq$ puisque sa première composante est maximum et sa deuxième composante est minimum. Ceci prouve le point 7.

Loi $\odot$: la loi $\odot$ sélectionne la première composante du poids en premier argument et ajoute les deuxièmes composantes des deux arguments. Elle est par construction non commutative. On montre le point 10 en ajoutant que la loi $\odot$ est interne. Le fait d'avoir la valeur 0 en première composante seulement si la seconde composante est ∞ n'est pas gênant. On remarquera que ε est absorbant par définition (point 12). On vérifie les points suivants :

– point 8 :

soient $x, y, z \in E_1$,

Alors $(x \odot y) \odot z = \begin{cases} \varepsilon = x \odot (y \odot z) \text{ si } \varepsilon \in \{x, y, z\} \\ (x_1, x_1 + y_1 + z_2) = x \odot (y \odot z) \text{ sinon} \end{cases}$

– point 9 :

soient $x, y, z \in E_1$ tels que $x \preceq y$ et $y \preceq z$.

Par définition de la relaion $\preceq$, il existe $a, b \in E_1$ tels que $x \odot a = y$ et $y \odot b = z$.

On a $z = y \odot b = x \odot a \odot b$. Donc $x \preceq z$.

– point 10 :

$\forall w \in E_1,\ w \odot e = w$

– point 11 :

$\forall w \in E_1,\ w \preceq w$

Non-distributivité à gauche de la loi $\odot$ par rapport à la loi $\oplus$:

Utilisons l'exemple[10] reporté dans la figure 4.4.

– Posons $w = (b \odot d) = (200, 2)$.

Il vient $\quad a \odot (c \oplus w) = (100, 5)$

$(a \odot c) \oplus (a \odot w) = (100, 4)$

Donc $a \odot (c \oplus w) \neq (a \odot c) \oplus (a \odot w)$

Distributivité à droite de la loi $\odot$ par rapport à la loi $\oplus$:

– Soient $a, b, c \in E_1$.

Montrons que $(a \oplus b) \odot c = (a \odot c) \oplus (b \odot c)$:

– *Si $c = \varepsilon$ alors* :

$(a \oplus b) \odot \varepsilon = \varepsilon = \varepsilon \oplus \varepsilon = (a \odot \varepsilon) \oplus (b \odot \varepsilon)$

– *Sinon si $c = e$ alors* :

$(a \oplus b) \odot e = (a \oplus b) = (a \odot e) \oplus (b \odot e)$

– *Sinon si $a = \varepsilon$* ($b = \varepsilon$ est un cas symétrique) *alors* :

$(\varepsilon \oplus b) \odot c = b \odot c = \varepsilon \oplus (b \odot c) = (\varepsilon \odot c) \oplus (b \odot c)$

– *Sinon si $a \oplus b = a$* d'après la première composante (cas symétrique avec b) *alors* :

[10]Intuitivement, si on tente de mettre à jour le plus court de chemin de l'AS 1 vers l'AS 7 en procédant à la concaténation de l'arc $\overrightarrow{(AS_1, AS_2)}$ de poids a avec le meilleur chemin sélectionné au préalable par l'AS 2 de poids $(c \oplus w)$, on aboutit au chemin 1,2,7 de poids $(100, 5)$. Alors que si on compare les poids des deux chemins 1,2,7 et 1,2,8,7 on obtient $(100, 4)$.

on a $(a \oplus b) \odot c = (a, c)$

la première composante du poids $(a \odot c)$ l'emporte sur la première composante du poids $(b \odot c)$

– *Sinon si* $a \oplus b = a$ d'après la deuxième composante (cas symétrique avec b) *alors* :

on a $(a \oplus b) \odot c = (a, c)$

la deuxième composante du poids $(a \odot c)$ l'emporte sur la première composante du poids $(b \odot c)$

□

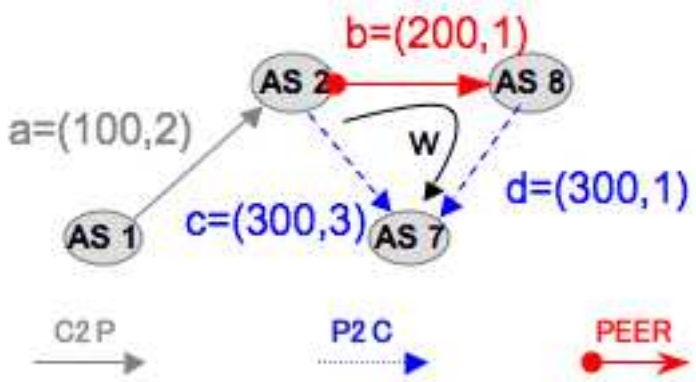


FIG. 4.4 – Non distributivité à gauche pour la structure algébrique des poids

Remarque : Signalons que l'opérateur $\odot$ n'admet pas d'élément neutre à gauche par construction. Cependant, cet opérateur est distributif à droite par rapport à la loi $\oplus$. Si on définit une loi $\odot'$ avec e l'élément neutre à gauche, alors en prenant $a = (100, 1)$, $b = e$ et $c = (200, 1)$, on obtiendrait $(a \oplus b) \odot' c = a \odot' c = (100, 2)$ alors que $(a \odot' c) \oplus (b \odot' c) = (a \odot' c) \oplus c = (100, 2) \oplus (200, 1) = (200, 1)$. Dès lors, la loi $\odot'$ ne serait pas distributive à droite par rapport à la loi $\oplus$.

D'après le lemme 32, la neutralité à gauche de l'élément e pour l'opérateur $\odot$ est la seule propriété manquante pour que la strucutre $(E_1, \oplus, \odot)$ soit un dioïde à droite. La structure $(E_1, \oplus, \odot)$ vérifie les points suivants :
– $(E_1, \oplus, \odot)$ est sélectif,
– L'élement e est le plus grand élément au sens de la relation d'ordre $\leq$,
– ε est le plus petit élément au sens de la relation d'ordre $\leq$.

c) Calcul de meilleurs chemins BGP AS-AS et le calcul de la pseudo-inverse de la matrice d'adjacence généralisée

On commence par montrer en quoi la recherche de meilleurs chemins BGP inter-AS se ramène à la résolution d'une équation de point fixe dans $(E_1, \oplus, \odot)$ et au calcul de la pseudo-inverse de la matrice d'adjacence généralisée du graphe des AS. La matrice d'adjacence généralisée du graphe G des AS, définie ci-dessous, est à valeurs dans l'ensemble E_1.
Comme supposé dans 4.1.1, rappelons que le processus de décision BGP pour chaque AS est simplifié. On considère que les décisons de chaque routeur de l'AS sont identiques, et qu'il existe, pour chaque AS, un ou plusieurs chemins préférés parmi tous les chemins possibles.
Soient AS_x, $AS_y \in G$ tels que $AS_x \neq AS_y$
Soient n chemins $\mu_1, \ldots, \mu_n$ dans le graphe des AS reliant les noeuds AS_x et AS_y. Le(s) chemin(s) μ^* préféré(s) par AS_X à destination de AS_Y vérifie(nt) la relation :

$$w(\mu^*) = \bigoplus_{i\in 1}^{n} w(\mu_i) = \bigoplus_{i=1}^{n} \left(w(\overrightarrow{AS_x,\, AS(\mu_i)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu_i)_{|\mu_i|-1},\, AS_y}) \right)$$

avec $\mu_i = \left(AS_x\,,\, AS(\mu_i)_2,\, \ldots,\, AS(\mu_i)_{|\mu_i|-1},\, AS_y \right)$, $i \in \{1 \ldots n\}$ un ensemble de chemins quelconques dans le graphe des AS.

En considérant tous les chemins possibles μ entre les noeuds AS_x et AS_y, et en notant $w_{x,y} = w(\mu^*)$, on obtient (par convention on a $\bigoplus_{\varnothing} = \varepsilon$) :

$$w_{x,y} = w(\mu^*) = \bigoplus_{\mu \in P_{AS_i, AS_j}} \left(w(\overrightarrow{AS_x,\, AS(\mu)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu)_{|\mu|-1},\, AS_y}) \right)$$

Puisque l'élément e est absorbant pour la loi $\oplus$, on remarque que le poids d'un circuit γ dans le graphe des AS est tel que :

$$\forall \gamma \in Cycles(G), \text{ on a } w(\gamma) \oplus e = e$$

On peut alors conclure que le graphe G est par définition sans cycle 0-absorbant[11]. On

[11] Voir [48] pour plus de détails.

peut s'abstraire des chemins avec cycles pour calculer le meilleur chemin de l'AS X vers l'AS Y.

Lemme 33. *"positivité" de $\odot$ pour la loi $\oplus$*

$(x \odot y) \oplus x = x$, $\forall x, y \in E_1$

Preuve :

Soient $x, y \in E_1$.

Montrons que : $(x \odot y) \oplus x = x$.

- *Si $x = \varepsilon$ alors*

  par absorption de ε pour la loi $\odot$, on a $\varepsilon \odot y = \varepsilon$

  par neutralité de ε pour la loi $\oplus$, on a $(x \odot y) \oplus x = \varepsilon \oplus \varepsilon = \varepsilon = x$
- *Si $y = \varepsilon$ alors*

  sachant que ε est l'élément neutre de $\oplus$ et aborbant pour $\odot$, on a $\varepsilon \oplus x = x$
- *Sinon* on a $x, y \in \mathbb{R}^{+*} \times \mathbb{N}^{+}$

  il vient $x \odot y = (x_1, x_2) \odot (y_1, y_2) = (x_1, x_2 + y_2)$

  donc $(x \odot y) \oplus x = (x_1, x_2 + y_2) \oplus (x_1, x_2) \underbrace{=}_{x_2+y_2 \geq x_2} (x_1, x_2) = x$.

□

Lemme 34.

Soit un chemin μ' du graphe des AS. On note μ le chemin obtenu en supprimant les cycles de μ'. Alors :

$w(\mu') \oplus w(\mu) = w(\mu)$

Preuve :

- *Soit μ'* un chemin du graphe des AS.

  Soit γ un cycle contenant $p + 1$ AS et inscrit dans μ'

  On note $\gamma = (AS_i, AS_{i_1}, \ldots, AS_{i_p}, AS_i)$

  On définit les quatres chemins suivants :
  - $\mu' = (\mu'_1, \gamma, \mu'_2)$
  - $\mu'' = (\mu'_1, AS_i, \mu'_2)$
  - $\mu_1 = (\mu'_1, AS_i)$
  - $\mu_2 = (AS_i, \mu'_2)$

  Il vient :

  $w(\mu') \oplus w(\mu'') \underbrace{=}_{\text{associativité de } \odot} w(\mu_1) \odot w(\gamma) \odot w(\mu_2) \oplus w(\mu_1) \odot w(\mu_2)$

$$w(\mu') \oplus w(\mu'') \underbrace{=}_{\text{distributivité à droite}} \left(\underbrace{w(\mu_1) \odot w(\gamma) \oplus w(\mu_1)}_{\underbrace{=\quad w(\mu_1)}_{\text{lemme de positivité}}} \right) \odot w(\mu_2) = w(\mu'')$$

On applique itérativement la procédure ci-dessus pour chaque cycle γ inscrit dans μ'.
On obtient à la dernière itération : $\mu'' = \mu$ et $w(\mu') \oplus w(\mu) = w(\mu)$.

□

Rappelons que :

$$w_{x,y} = w(\mu^*) = \bigoplus_{\mu \in P_{AS_x,AS_y}(0)} \left(w(\overrightarrow{AS_x,\ AS(\mu)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu)_{|\mu|-1},\ AS_y}) \right)$$
$$\oplus \bigoplus_{\mu \in P_{AS_x,AS_y} \setminus P_{AS_x,AS_y}(0)} \left(w(\overrightarrow{AS_x,\ AS(\mu)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu)_{|\mu|-1},\ AS_y}) \right)$$

Par définition, il existe un cycle inscrit dans chaque chemin μ' considéré dans la deuxième somme. Pour chacun de ces chemins, il existe un chemin μ duquel on a supprimé tous les cycles. D'après le lemme 34, le poids du chemin μ est absorbant pour le poids du chemin μ' .
Autrement dit :

$$\forall \mu' \in P_{AS_x,AS_y} \setminus P_{AS_x,AS_y}(0),\ \exists \mu \in P_{AS_x,AS_y}(0) \text{ tel que } w(\mu') \oplus w(\mu) = w(\mu)$$

On peut donc simplifier l'expression du poids $w_{x,y}$ de la manière suivante :

$$(i) \quad w_{x,y} = \bigoplus_{\mu \in P_{AS_x,AS_y}(0)} \left(w(\overrightarrow{AS_x,\ AS(\mu)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu)_{|\mu|-1},\ AS_y}) \right)$$

On partitionne l'ensemble des chemins élémentaires entre le noeuds AS_i et le noeud AS_j d'après les tailles des chemins. On obtient :

$$(ii) \quad w_{x,y} = \bigoplus_{k \in \mathbb{N}} \left(\bigoplus_{\mu \in P^k_{AS_x,AS_y}(0)} \left(w(\overrightarrow{AS_x,\ AS(\mu)_2}) \odot \ldots \odot w(\overrightarrow{AS(\mu)_{|\mu|-1},\ AS_y}) \right) \right)$$

On pose A la matrice carrée de taille $|V|$ à éléments dans E_1 dont les composantes vérifient :

$\forall AS_i, AS_j \in V,$

Si $\overrightarrow{(AS_i,\ AS_j)} \in E$ alors : $\quad (A)_{AS_i,AS_j} = w\overrightarrow{(AS_i,\ AS_j)}$

Sinon : $\quad (A)_{AS_i,AS_j} = \varepsilon$

La matrice A est la matrice d'adjacence généralisée associée au graphe G.

Par souci de clarté dans les expressions, on notera $(A)_{AS_i,AS_j} = A_{i,j}$, $\forall AS_i, AS_j \in V$.

Définition 35. *Soit $A \in \mathcal{M}_n(E_1)$ une matrice carrée.*

Par convention, $A^0 = \begin{pmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e \end{pmatrix}$.

En notant $A^n = \underbrace{A \odot A \odot \ldots \odot A}_{n\ fois}$, *on définit la matrice $A^{(n)}$ de la façon suivante :*

$$A^{(n)} = A^0 \oplus A \oplus A^2 \oplus \ldots A^n = \bigoplus_{0 \leq k \leq n} A^k$$

On définit également A^+ la matrice suivante :

$$A^+ = A^0 \oplus A \oplus A^2 \oplus \ldots = \bigoplus_{k \in \mathbb{N}} A^k$$

Sous réserve d'existence, la matrice A^+ est la pseudo-inverse de la matrice A.

Le terme $A^k_{x,y}$ de la matrice A^k correspond au plus grand (au sens de la relation $\leq$) chemin élémentaire de taille k exactement entre AS_x et AS_y . Le terme $A^{(k)}_{x,y}$ de la matrice $A^{(k)}$ correspond au plus grand (au sens de la relation $\leq$) chemin élémentaire de taille au plus k entre AS_x et AS_y .

On a :

$$\left(A^k\right)_{(x,y)} = \bigoplus_{\mu \in P^k_{AS_x,AS_y}(0)} \left(w\overrightarrow{(AS_x,\ (\mu)_2)} \odot \ldots \odot w\overrightarrow{((\mu)_{k-1},\ AS_y)} \right)$$

$$\left(A^{(k)}\right)_{(x,y)} = \bigoplus_{\mu \in P^{(k)}_{AS_x,AS_y}(0)} \left(w\overrightarrow{(AS_x,\ (\mu)_2)} \odot \ldots \odot w\overrightarrow{((\mu)_{|\mu|-1},\ AS_y)} \right)$$

L'équation (ii) s'écrit comme :

$$(iii) \quad w_{x,y} = \bigoplus_{k \in \mathbb{N}} \left(A^k\right)_{(x,y)}$$

En remarquant que la taille d'un chemin élémentaire du graphe G est bornée par le nombre de sommets moins 1, on obtient la preuve de l'existence de la pseudo-inverse de la matrice d'adjacence généralisée de G, puisque le calcul de la matrice A^+ devient une somme finie de matrices :

$$A^+ = \bigoplus_{k\in\mathbb{N}} A^k = \bigoplus_{k<|V|} A^k = A^{(|V|)}$$

L'équation (iii) se traduit par :

$$w_{x,y} = w(\mu^*) = \bigoplus_{k\in\mathcal{N}} \left(A^k\right)_{(x,y)} = \bigoplus_{k<|V|} \left(A^k\right)_{(x,y)} = \left(A^+\right)_{(x,y)}$$

L'équation ci-dessus montre finalement que le poids du plus court chemin entre le noeud AS_x et le noeud AS_y au sens du processus de décison BGP simplifié est égal au terme (x, y) de la pseudo-inverse de la matrice d'adjacence généralisée du graphe G.

En reprenant l'équation (i) et par distributivité à droite de la loi $\odot$ par rapport à la loi $\oplus$, on obtient :

$$w_{x,y} = w(\mu^*) = \bigoplus_{\overrightarrow{(AS_z,AS_y)}\in E} \left(\underbrace{\left(\bigoplus_{\mu'\in P_{AS_x,AS_z}(0)} w(\mu') \right)}_{=w_{y,z}} \odot w\overrightarrow{(AS_z,\, AS_y)} \right)$$

$$w_{x,y} = \bigoplus_{\overrightarrow{(AS_z,AS_y)}\in E} \left(w_{x,z} \odot w\overrightarrow{(AS_z,\, AS_y)} \right)$$

Cette dernière équation exprime les équations d'optimalité de Bellman.

Interprétation : Le meilleur chemin AS-AS du noeud AS_x vers le noeud AS_y est celui de poids maximal au sens de la relation $\leq$ parmi tous les chemins possibles partant du noeud AS_x et passant par un noeud AS_z quelconque avant d'arriver dans le noeud AS_y.

Par absorption de ε pour la loi $\odot$ et par neutralité de ε, on peut exprimer le poids $w_{x,y}$ en fonction de la matrice A et des poids des meilleurs chemins du sommet AS_x vers les autres sommets AS_z de l'ensemble V :

$$w_{x,y} = \bigoplus_{\overrightarrow{(AS_z,AS_y)}\in E} (w_{x,z} \odot A_{z,y})$$

$$w_{x,y} = \underbrace{\left(\bigoplus_{\overrightarrow{(AS_z,AS_y)}\notin E} \underbrace{w_{x,z} \odot A_{z,y}}_{=w_{x,z}\odot\varepsilon=\varepsilon} \right)}_{=\varepsilon} \oplus \bigoplus_{\overrightarrow{(AS_z,AS_y)}\in E} (w_{x,z} \odot A_{z,y})$$

$$w_{x,y} = \bigoplus_{AS_z\in V} (w_{x,z} \odot A_{z,y})$$

En considérant tous les couples de sommets AS_x et AS_y de l'ensemble V et en notant W la matrice des éléments $w_{x,y}$ indicée par les couples AS_x, AS_y, l'équation matricielle suivante apparait :

$(iiii) \quad W = W \odot A$

Cette équation de point fixe est une condition nécessaire à satisfaire pour W. Elle peut comporter des solutions incohérentes avec les poids du graphe G. Les meilleurs chemins sont en fait déterminés par la pseudo-inverse A^+ de la matrice A.

Lemme 36.

Lorsque la pseudo-inverse de A existe, alors la matrice A^+ est la solution maximale de l'équation $(iiii)$ au sens de la relation $\leq$.

Preuve :

Soit une solution W' de l'équation $(iiii)$.

W' vérifie l'équation ,donc :

$$W' \oplus A^+ \quad = \quad (W' \odot A) \oplus A^+$$

On remplace W' par $(W' \odot A)$. Il vient :

$$W' \oplus A^+ \quad = \quad ((W' \odot A) \odot A) \oplus A^+$$

$$W' \oplus A^+ \underbrace{=}_{\text{associativité}} (W' \odot A^2) \oplus A^+ = \ldots = \left(W' \odot \underbrace{A^{|V|}}_{=A^+} \right) \oplus A^+$$

On appliquant ce remplacement $|V|$ fois successivement, on obtient :

$$W' \oplus A^+ \quad = \quad \left(W' \odot \underbrace{A^{|V|}}_{=A^+} \right) \oplus A^+$$

$$W' \oplus A^+ \quad = \quad \underbrace{W' \odot A^+ \oplus A^+}_{\underbrace{=\ A^+}_{\text{lemme de positivité}}}$$

$$W' \oplus A^+ \quad = \quad A^+$$

Finalement A^+ est bien la solution maximale de l'équation $(iiii)$.

□

d) Calcul de la pseudo-inverse de la matrice d'adjacence

On peut calculer la pseudo-inverse de la matrice A avec la formule analytique par exemple ou avec d'autres algorithmes comme celui de Gauss-Seidel généralisé. Cependant, la taille du graphe des AS étant très grande, l'espace de stockage nécessaire pour calculer la pseudo-inverse est trop important. Les propriétés de la structure algébrique $(E_1, \oplus, \odot)$ permettent d'utiliser l'algorithme suivant pour calculer la matrice A^+ colonne par colonne. Une colonne de la matrice A^+ représente les poids des plus courts chemins de tous les sommets vers un seul.

Algorithme 8. *Algorithme de Dijkstra généralisé et transposé*

Entrée : le sommet destination *dst*

Pour tout i dans l'ensemble $V \backslash \{dst\}$ *faire*

$\pi(i) := \varepsilon$

$succ(i) := i$

fin pour

$\pi[dst] := e$

$succ[dst] := dst$

$S := \{dst\}$

Tant que $S \neq \varnothing$ f*aire*

Sélectionner un sommet $i \in S$ qui vérifie $\pi(i) = \bigoplus_{j \in S} \pi(j)$

$S := S \backslash \{i\}$

Pour tout (i, j) dans l'ensemble E *faire*

$\pi'(j) := \pi(j) \oplus (W(j, i) \otimes \pi(i))$

Si $\pi'(j) \neq \pi(j)$ *alors*

$\pi(j) := \pi'(j)$

$succ(j) := i$

$S := S \bigcup \{j\}$

fin si

fin pour

fin tant que

Sortie : le tableau de poids π et le tableau de successeurs *succ*

fin de l'algorithme

Lemme 37. *Convergence*
L'algorithme 8 converge.

Preuve :
Soit *dst* un noeud de V (*dst* est un AS du graphe G). Plaçons nous dans le cas du calcul des chemins vers le sommet destination *dst* avec l'algorithme 8. On commence la preuve par deux remarques. On parle d'une itération de l'algorithme lorsqu'un sommet i est choisit dans S et que l'on parcours ensuite les sommets adjacents à i. On dit qu'un sommet j est mis à jour dans une itération sur le sommet i lorsque le sommet j est adjacent au sommet i et que la condition $\pi'(j) \neq \pi(j)$ est vérifiée.

Sommet dst : $\pi(dst)$ est initialisé à la valeur e. La distance $\pi(dst)$ ne peut donc pas être mise à jour dans une itération sur un sommet i car e est absorbant pour la loi $\oplus$. On peut en déduire que le sommet *dst* n'est jamais réinséré dans S après la première itération.

Arcs de poids ε : Les arcs $\overrightarrow{(j,i)}$ de poids ε n'ont aucun impact dans la mise à jour d'un noeud j sélectionné dans une itération sur le noeud i. On donc peut supprimer les arcs de poids ε dans le graphe.

On propose de prouver la convergence de l'algorithme par contraposée.
Supposons que l'algorithme diverge. Montrons que l'on obtient une contradiction.
L'algorithme diverge, donc il existe au moins un sommet $u \neq dst$ qui est toujours mis à jour (sinon le nombre borné de sommets contredirait la divergence). Il existe donc une suite $(m_i)_{i\in\mathbb{N}^*}$ strictement croissante et non bornée d'itérations qui sélectionnent un même sommet u. A chaque itération m_i, pour i un entier, le sommet u est retiré de l'ensemble S. Il existe donc une suite $(m'_i)_{i\in\mathbb{N}^*}$ strictement croissante et non bornée d'itérations qui mettent à jour le sommet u et qui l'insèrent dans S. A chaque itération m'_i, le poids $\pi(u)$ du sommet u est mis à jour par la valeur $\pi_{m'_i}(u)$. La suite des valeurs $\left(\pi_{m'_i}(u)\right)_{i\in\mathbb{N}^*}$ est telle que :

(v) $\quad \pi_{m'_i}(u) \neq \pi_{m'_{i+1}}(u),\ \forall i \in \mathbb{N}$

(vi) $\quad \pi_{m'_{i+1}}(u) = \underbrace{\pi_{m'_i}(u) \oplus \overrightarrow{w(u, succ_{m'_i}(u))} \odot \pi_{m'_i}(succ(u))}_{\geq \pi_{m'_i}(u)},\ \forall i \in \mathbb{N}$

La relation $\leq$ étant une relation d'ordre, on obtient :

(vii) $\quad \pi_{m'_i}(u) \oplus \pi_{m'_n}(u) = \pi_{m'_n}(u),\ \forall n \in \mathbb{N}^*,\ \forall i < n$

$(viii)$ $\quad \pi_{m'_i}(u) \neq \pi_{m'_n}(u),\ \forall n \in \mathbb{N}^*,\ \forall i < n$

Pour n positif , on note $\pi_{m'_n}(u) = (w_1^{(n)}, w_2^{(n)})$. La suite $\left(\pi_{m'_i}(u)\right)_{i\in\mathbb{N}}$ est strictement croissante d'après (vii) et $(viii)$. Par définition de la loi $\oplus$, qui ordonne d'abord les poids BGP

par valeurs décroissantes de la première composante, la suite $\left(w_1^{(i)}\right)_{i\in\mathbb{N}}$ est une suite croissante. Or le nombre de valeurs admissibles pour la première composante est borné par le nombre d'arcs du graphe plus deux[12]. La valeur $w_1^{(i)}$ est donc soit le premier argument de la valeur initiale, soit le premier argument du poids d'un arc $\overrightarrow{(u,s)}$, s étant le sommet qui met à jour u.

La suite $\left(w_1^{(i)}\right)_{i\in\mathbb{N}}$ est finalement croissante et bornée. Elle converge donc vers une valeur w_1^*. Il existe un entier à partir duquel la suite est stationnaire. Autrement dit :

$\exists n_1 \in \mathbb{N}^*$ tel que $w_1^{(p)} = w_1^{(n_1)} = w_1^*$, $\forall p \geq n_1$

Considérons le poids $\pi_{m'_{n_1}}(u)$ correspondant à la n_1^{ime} mise à jour du sommet u. Ce poids est différent de ε car avant l'itération m'_1, $\pi(u) = \varepsilon$ (la valeur initiale).

D'après (vii) et $(viii)$, on a :

$(\underbrace{w_1^{(n)}}_{=w_1^*}, w_2^{(n)}) \oplus (\underbrace{w_1^{(n_1)}}_{=w_1^*}, w_2^{(n_1)}) = (w_1^*, w_2^{(n)})$, $\forall n > n_1$

$(\underbrace{w_1^{(n_1)}}_{=w_1^*}, w_2^{(n_1)}) \neq (w_1^*, w_2^{(n_1)})$, $\forall n > n_1$

Par définition de l'opérateur $\oplus$, on obtient :

$w_2^{(n)} < w_2^{(n_1)}$, $\forall n > n_1$

La suite $\left(w_2^{(i)}\right)_{i\in\mathbb{N}, i>n_1}$ est strictement décroissante. Or la suite $\left(w_2^{(i)}\right)_{i\in\mathbb{N}, i>n_1}$ est bornée inférieurement par la valeur 0. La suite converge donc vers une valeur w_2^*. Il existe un entier à partir duquel la suite est staionnaire. Autrement dit :

$\exists n_2 > n_1 \in \mathbb{N}^*$ tel que $w_2^{(n)} = w_2^{n_2} = w_2^*$, $\forall n > n_2$.

Finalement, il existe une itération m'_{n_2} telle que le poids du sommet u est stationnaire.

$\pi_m(u) = (w_1^*, w_2^*)$, $\forall m \geq m'_{n_2}$

Ceci contredit la sélection du sommet u à partir de l'itération m'_{n_2+1}, puisque $\pi_{m'_{n_2+1}}(u) = \pi_{m'_{n_2}}(u) = (w_1^*, w_2^*)$.

□

Après avoir prouvé la convergence de l'algorithme, on montre que l'algorithme calcule bien une colonne de A^+.

[12]En effet la loi $\oplus$ sélectionne la plus grande première composante entre celles des deux arguments. La loi $\odot$ conserve la première composante du premier argument.

Lemme 38.

L'algorithme de Dijkstra généralisé et transposé permet de calculer, colonne par colonne, la matrice A^+

Preuve :

Si le graphe privé des arcs de poids ε est connexe, alors on a :

$\forall x \in V,\ \pi(x) \neq \varepsilon$

On réduit donc le graphe G au sous-graphe induit par la composante connexe contenant dst.

Soit src un noeud du graphe. Soit $\pi(src)$ son poids pour le chemin vers dst. A chaque itération de l'algorithme, un des trois cas suivants est vrai.

- Si src n'a pas été traité lors des précédentes itérations, alors $\pi(src) = \varepsilon$.
- Si src est dans S, alors src a été mis à jour par un autre sommet, et ses arcs entrants n'ont pas encore été examinés.
- Si src n'est pas dans S, alors src a été mis à jour par le sommet $succ(src)$ dans une itération précédente, et aucun autre sommet déjà examiné et n'étant pas dans S n'a pu mettre à jour $\pi(src)$.

La dernière remarque montre que pour chaque itération :

$$\pi(src) = \bigoplus_{i \notin S / \overrightarrow{(src,i)} \in E} w\overrightarrow{(src,i)} \odot \pi(i)$$

Après la dernière itération, l'ensemble S est vide. Donc, l'égalité suivante est vérifiée :

$\pi(src) = \bigoplus_{i \in V / \overrightarrow{(src,i)} \in E} w\overrightarrow{(src,i)} \odot \pi(i),\ \forall src \in V$

Ceci montre que la solution calculée par l'algorithme vérifie l'équation de point fixe.

Maintenant, montrons par contraposée que cette solution est la solution maximale au sens de $\geq$ de l'équation de point fixe.

On peut d'abord remarquer que la pseudo-inverse de A vérifie :

$$A^+ = A^{(|V|)} = \bigoplus_{\mu \in P^{(|V|)}_{v,dst}(0)} \left(w\overrightarrow{(AS_x,\ AS\left(\mu_2\right))} \odot \ldots \odot w\overrightarrow{(AS\left(\mu_{|\mu|-1}\right),\ AS_y)} \right)$$

Grâce à la sélectivité de la loi $\oplus$, on a :

$(viiii) \quad \forall u \in V, \exists \mu \in P^{(|V|)}_{u,dst}(0) \,/\, w(\mu) = A^+_{u,dst}$

Supposons que la solution π calculée par l'algorithme ne soit pas la solution maximale de l'équation.

On note $(A^+)_{dst}$ la colonne d'indice dst de la matrice A^+

$(A^+)_{dst} = (\ldots, A^+_{i,dst}, \ldots)_{i \in V}$

Cette colonne est plus grande que π au sens de $\leq$. Donc qu'il existe un sommet v pour lequel le terme (v, dst) de la matrice A^+ est plus grand strictement que le poids $\pi(v)$. Autrement dit :

(x) $\exists v \in V$ tel que $(A^+)_{v,dst} \geq \pi(v)$ et $(A^+)_{v,dst} \neq \pi(v)$

La solution π est calculée par l'algorithme et elle vérifie l'équation suivante après la dernière itération :

$w(\overrightarrow{v,i}) \odot \pi(i) = \pi(v), \forall \overrightarrow{(v,i)} \in E$

Donc avec (x) :

(xi) $w(\overrightarrow{v,i}) \odot \pi(i) \leq A^+_{v,dst}$ et $w(\overrightarrow{v,i}) \odot \pi(i) \neq A^+_{v,dst}, \forall \overrightarrow{(v,i)} \in E$

En utilisant $(viiii)$, on introduit un chemin élémentaire μ entre le noeud v et le noeud dst dont le poids est $(A^+)_{v,dst}$. Autrement dit :

$\exists \mu \in P^{(|V|)}_{v,dst}(0) / \quad A^+_{v,dst} = w(\mu)$

Pour i égal à μ_2 (le deuxième sommet de μ) dans (xi), on obtient :

$w(\overrightarrow{v,\mu_2}) \odot \pi(\mu_2) < w(v, \mu_2, \ldots, \mu_{|\mu|})$

$w(\overrightarrow{v,\mu_2}) \odot \pi(\mu_2) < w(\overrightarrow{v,\mu_2}) \odot w(\mu_2, \ldots, \mu_{|\mu|})$

Sachant que les premières composantes des deux poids $w(\overrightarrow{v,\mu_2}) \odot \pi(\mu_2)$ et $w(\overrightarrow{v,\mu_2}) \odot w(\mu_2, \ldots, \mu_{|\mu|})$ sont égales, on a d'après la définition de $\oplus$:

(xii) $(\pi(\mu_2)_2 > (w(\mu_2, \ldots, \mu_{|\mu|}))_2$

Puisque A^+ est solution maximale (au sens de $\leq$) de l'équation de point fixe, on a :

$(A^+)_{\mu_2,dst} \geq \pi(\mu_2)$

Finalement en utilisant (xii) :

$(A^+)_{\mu_2,dst} \neq \pi(\mu_2)$.

On note $w = \mu_2$. Le raisonnement sur v a montré que le prochain sommet w sur le meilleur chemin BGP μ de v à dst vérifie $A^+_{w,dst} > \pi(w)$. On applique ce raisonnement successivement sur chaque sommet du chemin μ. Pour le dernier sommet, on peut prouver par récurrence que :

$$\underbrace{A^+_{\mu_{|\mu|},dst}}_{=A^+_{dst,dst}=e} \neq \underbrace{\pi(\mu_{|\mu|}}_{=\pi(dst)=e})$$

Contradiction : $e \neq e$.

□

Les deux lemmes 37 et 38 montrent que l'algorithme 8 permet de calculer chaque colonne de la pseudo-inverse de la matrice d'adjacence généralisée du graphe G. Les chemins

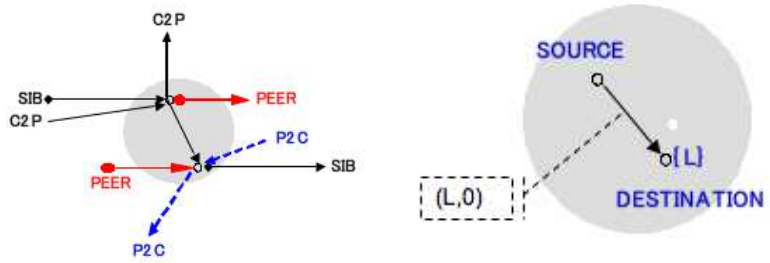
calculés sont les plus courts chemins au sens du processus de décision BGP simplifié.

4.1.4 Utilisation du graphe étendu avec la structure algébrique de poids

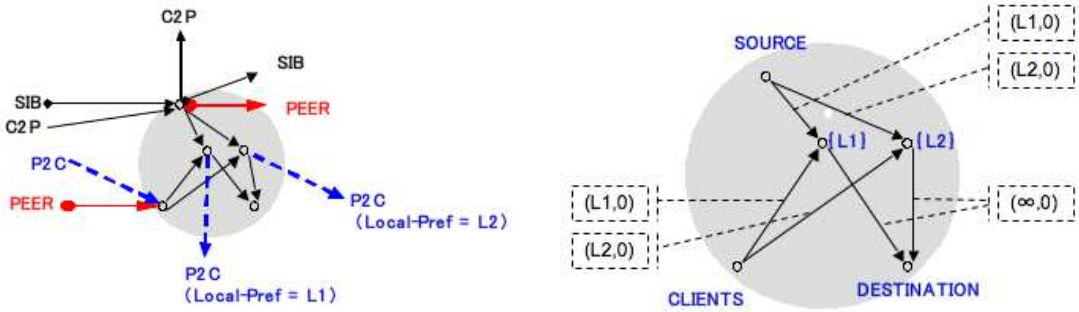
Les chemins calculés par l'algorithme 8 dans le graphe G peuvent comporter des vallées. Pour calculer des chemins sans vallée qui respectent le processus de décision BGP simplifié, on utilise le modéle de graphe étendu conjointement à la structure algébrique $(E_1, \oplus, \odot)$ pour manipuler les poids sur les arcs. On montre ci-dessous qu'il est possible de résoudre le problème du meilleur chemin inter-AS au sens BGP et sans vallée via l'algorithme 8 utilisé sur le graphe étendu des AS. Dans le graphe étendu, l'algorithme calcule directement des chemins qui correspondent à des chemins sans vallée dans le graphe des AS. On rappelle que pour obtenir l'équivalence entre les chemins du graphe étendu et les chemins sans vallée du graphe des AS, on doit sélectionner un ensemble restreint de noeuds source et destination pour l'algorithme :

– les noeuds 1 sont les seules destinations à considérer
– les noeuds 0 sont les seules sources à prendre en compte.

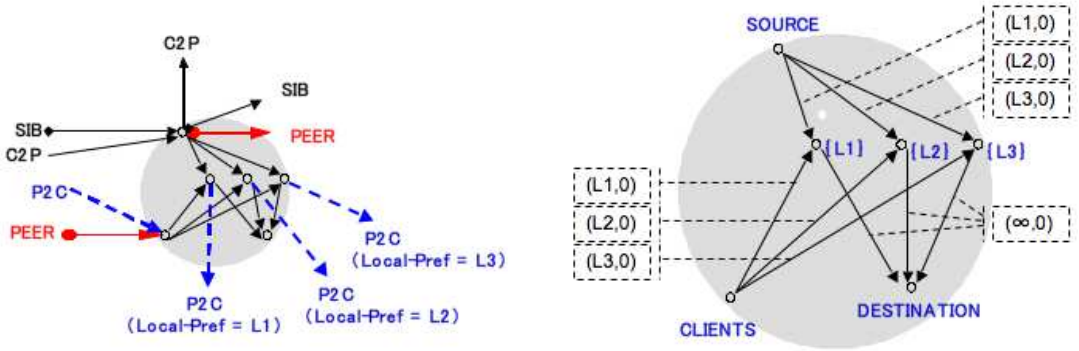
Dans l'algorithme 8, un problème se pose au niveau des calculs dans les noeuds 0. Une attention particulière doit être portée sur la définition de l'arc interne entre les noeuds 0 et les noeuds 1. On détaille deux cas. Un premier cas où seulement une seule valeur de *Local Pref* est utilisée pour les clients (voir la figure 4.5(a)). Un second cas où plusieurs valeurs de *Local Pref* peuvent être utilisées pour les différents clients d'un AS (voir les figures 4.5(b) et 4.5(c)). Dans le premier cas on fixe la première composante du poids de l'arc interne entre les noeuds 0 et 1 à la valeur *Local Pref* utilisée pour les clients. Dans le second cas, pour chaque AS on ajoute autant de noeuds qu'il y a de valeurs *Local Pref* différentes attribuées aux clients de l'AS. La structure topologique du graphe va donc dépendre à la fois des accords d'interconnexion et des différentes valeurs *Local Pref* attribuées aux AS clients.



(a) un noeud du graphe étendu des AS. Ce noeud est modifié pour utiliser la structure de poids BGP avec des *Local Pref* clients uniques. L désigne la valeur du *Local Pref* attribuée aux client de l'AS représenté par ces noeuds. Le noeud 0 est appelé noeud source, et le noeud 1 est appelé noeud destination.

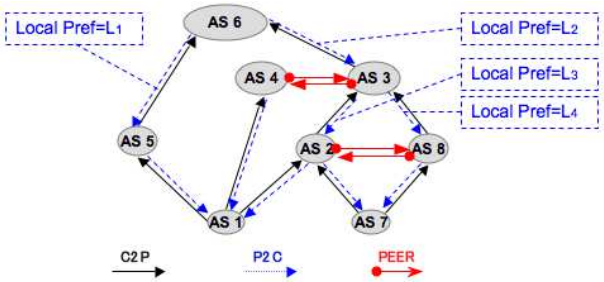


(b) un noeud du graphe étendu des AS. Ce noeud est modifié pour utiliser la structure de poids BGP avec deux *Local Pref* clients. L_1 et L_2 désignent ces valeurs de *Local Pref* attribuées aux client d'un AS.

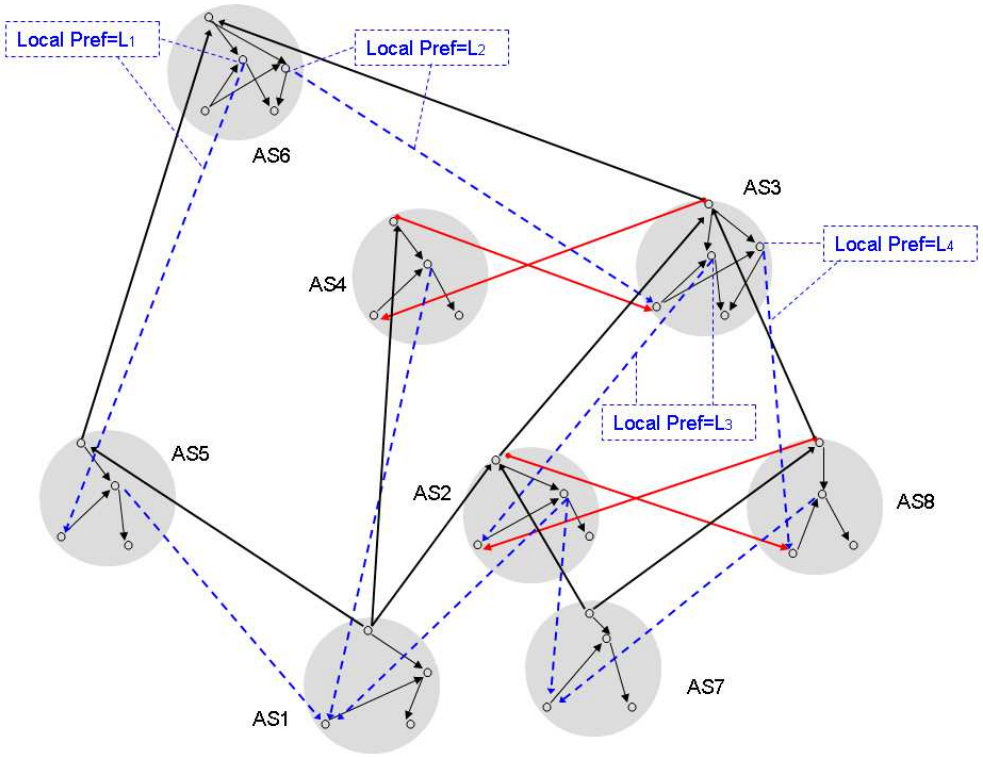


(c) un noeud du graphe étendu des AS. Ce noeud est modifié pour utiliser la structure de poids BGP avec trois *Local Pref* clients. L_1, L_2 et L_3 désignent ces valeurs de *Local Pref* attribuées aux clients d'un AS.

FIG. 4.5 – Prise en compte de plusieurs valeurs de *Local Pref* pour la transformation du graphe des AS en graphe étendu des AS



(a) graphe des AS



(b) graphe étendu des AS avec prise en compte de différentes valeurs de *Local Pref* pour les clients d'un même AS. Les valeurs de *Local Pref* indiquées avec des cadres en pointillés désignent des sommets internes ou des poids sur les liens.

FIG. 4.6 – Exemple de transformation d'un graphe des AS avec plusieurs valeurs de Local Pref pour les clients en un graphe étendu des AS

La structure topologique du graphe étendu permet d'obtenir une bijection entre les chemins du graphe des AS sans vallée et les chemins topologiques entre noeuds $SOURCE$ et noeuds $DESTINATION$ du graphe étendu. Cette bijection conserve le poids BGP d'après le lemme 39 ci-dessous. On notera respectivement S_i, D_i, C_i les noeuds de type $SOURCE$, $DESTINATION$ et $CLIENTS$ dans le graphe étendu pour un AS i quelconque.

Lemme 39.

Pour chaque chemin élémentaire sans vallée μ entre deux sommets i et j du graphe non étendu des AS, on sait qu'il existe un unique chemin du graphe étendu μ_e entre les deux noeuds S_i et D_j du graphe étendu.
On a l'égalité suivante : $w(\mu) = w(\mu_e)$

Preuve :

Montrons par récurrence sur la taille n ($n \geq 1$) du chemin μ, que :

$w(\mu) = w(\mu_e)$, $\forall n \geq 1$, $\forall i, j, \forall \mu \in P^n_{i,j}(0)$.

On rappele que μ désigne un chemin du graphe des AS, et μ_e désigne l'unique chemin du graphe étendu associé. Le chemin μ_e commence un par noeud de type $SOURCE$ et se termine par un noeud de type $DESTINATION$.

n=1 : Si μ est de taille 1, alors il est trivial de vérifier que $w(\mu) = w(\mu_e)$ en considérant les différents cas pour chaque label de l'arc $\overrightarrow{(\mu_1, \mu_2)}$.

passage n⟶n+1 : Soit $n \geq 1$. *Supposons que* $w(\mu) = w(\mu_e)$, $\forall i, j, \forall \mu \in P^n_{i,j}(0)$.

Montrons que cette propriété est vraie pour le rang $n + 1$.

Soient i, j et $\mu \in P^n_{i,j}(0)$.

On note μ_e l'unique chemin du graphe étendu qui correspond au chemin μ dans le graphe non étendu des AS.

Tout d'abord, remarquons que $\mu_e = (S_i, (\mu_e)_2, \ldots, D_j)$. Suivant le label du premier arc du chemin μ, le sommet $(\mu_e)_2$ est soit le sommet S_{μ_2}, soit le sommet C_{μ_2}, soit un somment LP_i puis le sommet C_{μ_2}. Dans tous les cas, le poids du sous-chemin dans le graphe étendu du sommet S_i vers le premier sommet de l'AS μ_2 a la même première composante égale à la première composante du poids de l'arc $\overrightarrow{(i, \mu_2)}$ dans le graphe non étendu. C'est la transformation conforme à la figure 4.5 qui offre cette propriété. La seconde composante du poids est aussi égale à la seconde composante du poids de l'arc $\overrightarrow{(i, \mu_2)}$ car les arcs internes entre les noeuds représentant le même AS ont une seconde composante nulle.
Donc :

$$\begin{cases} w\overrightarrow{(S_i, S_{\mu_2})} = w\overrightarrow{(i, \mu_2)} \\ \text{ou} \\ w\overrightarrow{(S_i, \mu_2)} = w\overrightarrow{(i, \mu_2)} \\ \text{ou} \\ w(S_i, LP(\rightarrow (\mu)_2)_i, C_{\mu_2}) = w\overrightarrow{(i, \mu_2)} \end{cases}$$

D'après l'hypothèse de récurrence, on peut en déduire que :

$w(S_{\mu_2}, \ldots, D_j) = w(\mu_2, \ldots, \mu_{|\mu|})$

Si le premier sommet correspondant à l'AS μ_2 dans le chemin μ_e est le sommet S_{μ_2} *alors* :

$w(\mu_e) = w(\overrightarrow{S_i, S_{\mu_2}}) \odot w(S_{\mu_2}, \ldots, D_j) = w(\overrightarrow{i, \mu_2}) \odot w(\mu_2, \ldots, \mu_{|\mu|}) = w(\mu)$

Si le premier sommet correspondant à l'AS μ_2 dans le chemin μ_e est le sommet C_{μ_2} *alors* :

Sachant que le chemin d'AS μ est sans vallée, le chemin $w(S_{\mu_2}, \ldots, D_j)$ passe par un noeud LP_{μ_2} . Sinon il n'existerait pas de chemin dans le graphe Pe. On peut en déduire le poids en partant du sommet C_{μ_2} vers D_j :

$w(S_{\mu_2}, \ldots, D_j) = w(S_{\mu_2}, LP(\rightarrow \mu_3)_{\mu_2}, C_{\mu_3}, \ldots) = w(LP(\rightarrow \mu_3)_{\mu_2}, C_{\mu_3}, \ldots)$

$w(S_{\mu_2}, \ldots, D_j) = w(C_{\mu_2}, LP(\rightarrow \mu_3)_{\mu_2}, C_{\mu_3}, \ldots)$

D'où :

$w(\mu_e) = w(S_i, \ldots, C_{\mu_2}, LP(\rightarrow \mu_3)_{\mu_2}, C_{\mu_3}, \ldots)$

$w(\mu_e) = w(S_i, \ldots, C_{\mu_2}) \odot w(C_{\mu_2}, LP(\rightarrow \mu_3)_{\mu_2}, C_{\mu_3}, \ldots) = w(\overrightarrow{i, \mu_2}) \odot w(S_{\mu_2}, \ldots, D_j)$

en utilisant l'hypothèse de récurrence, il vient :

$w(\mu_e) = w(\overrightarrow{i, \mu_2}) \odot w(\mu_2, \ldots, \mu_{|\mu|}) = w(\mu)$

□

Lemme 40.

Pour chaque sommet destination dans le graphe, l'algorithme 8 converge et calcule dans le graphe étendu le meilleur chemin sans vallée.

Preuve :

La preuve de convergence pour l'algorithme 8 reste valable pour le graphe étendu (cf. lemme 37). Il reste à prouver que les chemins calculés dans l'algorithme avec le graphe étendu, sont les mêmes que les meilleurs chemins BGP sans vallée.

Soit $j \in V$ un sommet destination et i un sommet source.

*S'*il n'existe pas de chemin sans vallée entre i et j dans le graphe des AS, *alors* il n'existe pas de chemin topologique du noeud S_i vers le noeud D_j dans le graphe étendu. L'algorithme ne met donc jamais à jour le poids du sommet i. Ainsi, le poids $\pi(i)$ calculé dans le graphe étendu vérifie bien :

$$\pi(i) = \varepsilon = \bigoplus_{\varnothing} = \bigoplus_{\mu \in P_{i,j}^{|V|}(0) \, et \, \mu \text{ est sans vallée}} w(\mu).$$

Sinon au moins un chemin sans vallée existe entre i et j, *alors* le chemin recherché est :

$$\mu^* = ArgMin \left(\bigoplus_{\mu \in P_{i,j}^{|V|}(0) \, et \, \mu \text{ est sans vallée}} w(\mu) \right)$$

On note $Pe_{u,v}(0)$ l'ensemble des chemins élémentaires entre deux noeuds du graphe étendu.

L'algorithme 8 dans le graphe étendu, calcule le chemin :

$$\gamma_e^* = ArgMin\left(\bigoplus_{\gamma_e \in Pe_{S_i,D_j}(0)} w(\gamma_e)\right)$$

On note γ le chemin d'AS, sans vallée dans le graphe des AS, associé au chemin γ_e^* du graphe étendu. Par définition de μ^*, on a :

$(xiii)$ $\quad w(\mu^*) \oplus w(\gamma) = w(\mu^*)$

Comme le chemin μ^*est sans vallée, il correspond à un chemin μ_e du graphe étendu. Par définition de γ_e^*, on a :

$(xiiii)$ $\quad w(\mu_e) \oplus w(\gamma_e^*) = w(\gamma_e^*)$

En utilisant le résultat du lemme 39, on obtient :

$w(\mu^*) = w(\mu_e)$ et $w(\gamma_e^*) = w(\gamma)$

Donc en utilisant $(xiiii)$ on obient :

$w(\mu^*) \oplus w(\gamma) = w(\gamma)$

On peut donc conclure sur la validité de l'égalité :

$w(\mu^*) = w(\gamma_e^*)$

□

4.2 Un modèle de cheminement AS-Préfixe

Dans ce paragraphe, on s'intéresse à raffiner le modèle de graphe étendu des AS, en introduisant les NLRI BGP dans le graphe. Dans la pratique, un AS sélectionne les différents préfixes qu'il va exporter vers chaque AS voisin. De plus, un AS peut se répéter de plusieurs façons différentes dans les *AS_PATH* de routes envoyées pour le même préfixe à des AS voisins différents.

On s'intéresse à traiter la sélectivité des annonces de préfixe faite par chaque AS origine. On montre à cet effet comment transformer un graphe étendu des AS en un graphe AS-préfixe étendu en utilisant les matrices de politique d'annonce.

4.2.1 Analyse des matrices de politique d'annonce de préfixe

Un AS origine peut, pour chaque préfixe qu'il annonce, définir des règles dans sa politique de routage qui tiennent compte de plusieurs attributs BGP. Ces règles tiennent compte en particulier de l'AS voisin et de la session BGP destination de chaque route. On a vu dans

2.3.1 qu'un AS traite généralement ses préfixes par groupe. On conserve cette notion de groupe de préfixes annoncé par chaque AS à chacun de ses AS voisins. Ces groupes de préfixes, appelés aggrégats, vont être ajoutés au graphe étendu. Ils permettent de prendre en compte la sélectivité des annonces des AS tout en limitant le nombre de noeuds et le nombre d'arcs ajoutés dans le graphe.

4.2.2 Ajout des noeuds préfixes au graphe étendu

Pour ajouter des matrices de politique d'annonce à un graphe étendu des AS, nous allons ajouter deux nouveaux types de noeuds, et trois nouveaux types de liens.

Préfixes : nous ajoutons un noeud pour chaque préfixe.

Agrégats de préfixes : pour chaque AS, nous ajoutons un noeud pour chaque groupe de préfixes annoncé par l'AS à chaque AS voisin.

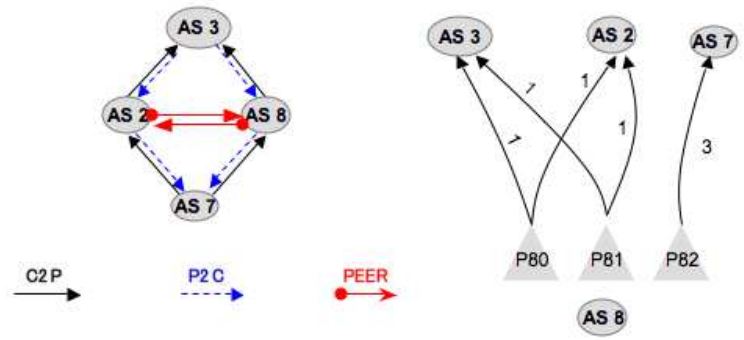
Arcs d'inclusion : pour chaque AS origine et pour chacun de ses groupes de préfixes, nous ajoutons un arc entre le noeud agrégat et chacun des préfixes du groupe.

Arcs d'AS origine : pour chaque AS, nous ajoutons un arc entre le noeud interne source de l'AS du graphe préfixe étendu (qui n'est pas le même que le noeud interne source dans le graphe étendu des AS). Cet arc explicite qu'un AS puisse joindre ses propres préfixes.

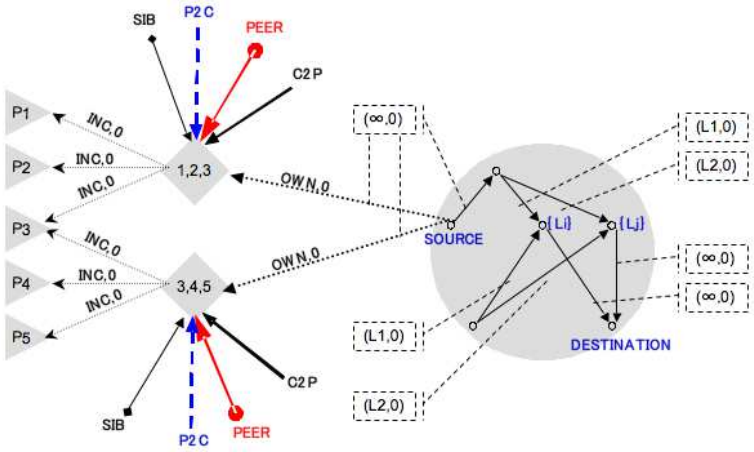
Arcs d'accès à un préfixe : pour chaque AS et pour chaque atome de politique d'annonce de l'AS, on relie l'AS voisin dans l'atome au noeud agrégat associé de l'AS qui annonce le préfixe.

FIG. 4.7 – Détail des noeuds d'un AS origine relié aux noeuds préfixes et aux noeuds aggrégats dans le graphe étendu AS-préfixe

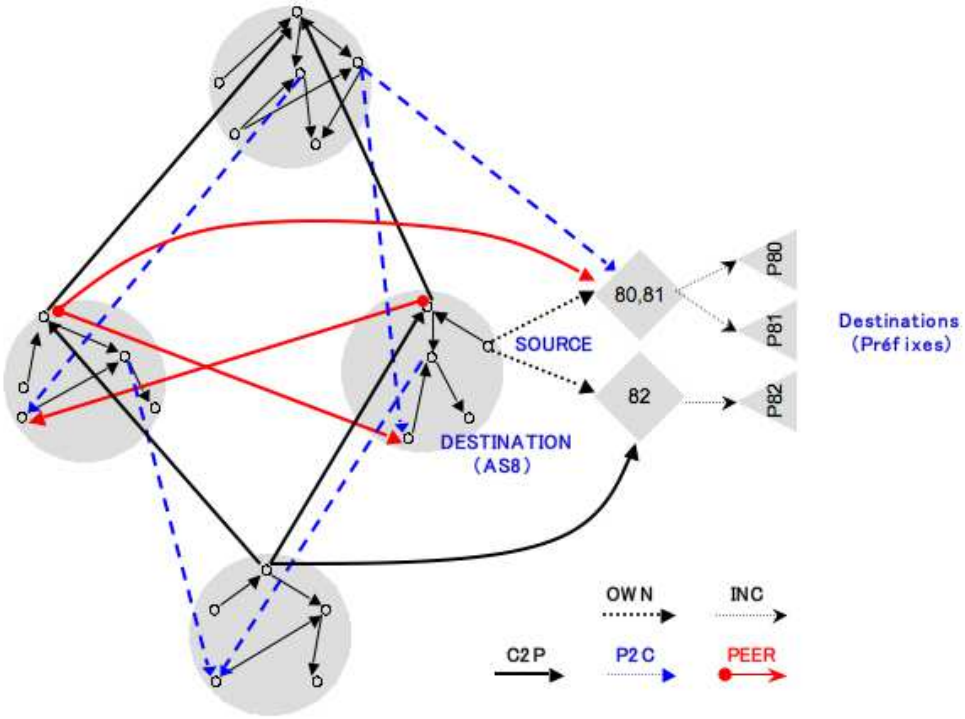
Les figures 4.7 et 4.8 montrent des exemples de transformation.



(a) exmple de graphe des AS avec l'illustration de la matrice de politique d'annonce de l'AS 8.



(b) un noeud du graphe étendu des AS-préfixe. D'une part le nombre de *Local Pref* clients est propre à chaque AS. D'autre part, les différents aggrégats de préfixes sont aussi propres à chaque AS. Le poids des arcs d'inclusion et des arcs d'accès est égal à e.



(c) transformation du graphe de la figure 4.8(a) en graphe étendu AS-préfixe.

FIG. 4.8 – Exemple de transformation d'un graphe étendu des AS et d'une matrice de politique d'annonce en un graphe étendu AS-préfixe

4.3 Résultats numériques

Pour évaluer l'intérêt et la pertinence de notre modélisation, on va à partir des mêmes tomographies pour construire le modèle et pour comparer les chemins réels aux chemins calculés par l'algorithme 8. Dans la littérature, certains auteurs utilisent une partie des données d'entrée pour construire le modèle et la partie restante pour valider les chemins calculés par le modèle [133]. On ne procède pas de cette manière.

4.3.1 Données d'entrée

a) Les tomographies BGP

On utilise deux tomographies de référence reportées dans le tableau 4.1. Pour ces tomogaphies, on calcule pour chaque modèle de graphe étendu, les chemins d'AS entre toutes les paires origine-destination pour lesquelles on dispose d'au moins un chemin réel. Lorsque l'on a plusieurs chemins réels pour un même couple origine-destination, on sélectionne seulement le chemin d'AS le plus court en nombre de bonds. Pour la tomographie P_{2003}, on construit le modèle AS-préfixe en utilisant les matrices de politique d'annonce. On compare une route de cette tomographie pour chaque AS source vers chaque préfixe avec le chemin d'AS vers le préfixe calculé avec le modèle.

	Période	Chemins	Couples origine-destination (O-D)
Tomographies pour les chemins inter-AS			
P_{2003}	15 Nov. 2003	621 343	460 566
P_{2005}	$1 \rightarrow 31$ Janv. 2005	1 840 077	1 402 058
Tomographies pour les chemins entre AS et préfixes			
P_{2003}	15 Nov. 2003	3 762 213	3 597 724

TAB. 4.1 – Tomographies utilisées pour l'inférence de chemins interdomaines

b) Les modèles de graphe

On utilise les deux modélisations suivantes pour le routage entre AS :

G-AS : modèle de graphe étendu des AS construit à partir d'une solution du problème d'inférence des accords d'interconnexion. L'algorithme 8 calcule des plus courts chemins en nombre de bonds sans vallée.

G-LP : modèle de graphe étendu des AS construit à partir d'une solution du problème d'inférence des accords d'interconnexion où l'on fixe la première composante

du poids de chaque arc en fonction de l'accord d'interconnexion[13]. L'algorithme 8 calcule des plus courts chemins en nombre de bonds sans vallée avec la préférence respective des routes vers les clients, puis vers les peer puis vers les fournisseurs.

Ces deux modèles sont utilisés avec les deux tomographies dans les résultats.
On utilise la modélisation suivante pour le calcul des chemins d'AS vers des préfixes :

G-PREFIX : modèle de graphe AS-préfixe étendu construit à partir d'une solution du problème d'inférence des accords d'interconnexion. L'algorithme 8 calcule des plus courts chemins en nombre de bonds sans vallée des AS vers les préfixes.

Ce modèle est utilisé avec une seule tomographie.

c) Indicateurs de comparaison

Pour comparer les chemins calculés avec les modèles et les chemins réels des tomographies, on utilise les indicateurs suivants :

EGAL : égalité entre le chemin réel et le chemin calculé avec le modèle sans tenir compte des répétitions d'AS.

PREMIER_AS : le premier AS traversé est le même dans le chemin réel et dans le chemin calculé avec le modèle.

DERNIER_AS : le dernier AS traversé est le même dans le chemin réel et dans le chemin calculé avec le modèle.

TAILLE : le chemin réel et le chemin calculé avec le modèle ont la même taille en nombre de bonds (on ne tient pas compte des répétitions).

TROP-COURT : le chemin calculé avec le modèle est plus court que le chemin réel.

TROP-LONG : le chemin calculé avec le modèle est plus long que le chemin réel.

TROP-PREF : le chemin calculé avec le modèle a un *Local Pref* strictement plus grand que le *Local Pref* du chemin réel.

[13]On préfère les clients d'abord, puis les peers, puis les fournisseurs.

4.3.2 Comparaisons

a) Comparaison des chemins inférés avec les tomographies

On reporte dans le tableau 4.2 les valeurs des indicateurs pour les deux tomographies et les différents modèles de graphe.

Tomographie	*Indicateurs*						
	EGAL	*PERMIER-AS*	*DERNIER-AS*	*TAILLE*	*TROP-COURT*	*TROP-LONG*	*TROP-PREF*
Modèle de graphe G-AS							
P_{2003}	*44.6%*	*51.8%*	*76.1%*	*81.1%*	*18.9%*	*0.02%*	X
P_{2005}	*33.5%*	*40.2%*	*73.5%*	*74.4%*	*25.5%*	*0.1%*	X
Modèle de graphe G-LP							
P_{2003}	*10.9%*	*17.1%*	*71%*	*28.2%*	*3.9%*	*67.9%*	*80.6%*
P_{2005}	*6.1%*	*7.8%*	*68.1%*	*31.7%*	*6.9%*	*61.4%*	*87.9%*
Modèle de graphe G-PREFIX							
P_{2003}	61.7%	68.7%	84.9%	85%	14.9%	0.1%	X

TAB. 4.2 – Evaluation de chaque modèle de graphe sur les tomographies

Le modèle *G-PREFIX* permet d'inférer 61% des chemins réels des AS vers les préfixes. Ce modèle fournit les meilleurs résultats. Le modèle *G-AS* qui ne prend pas en compte les préfixes, permet d'inférer de 33% à 44% des chemins réels. Cette valeur est d'autant plus faible que l'on dispose de plusieurs chemins vers un même AS (pour des préfixes différents ou non). En effet pour un même couple origine-destination, on sélectionne un seul chemin réel pour le confronter au chemin calculé avec chaque modèle.
Le modèle *G-LP* permet d'inférer seulement de 6% à 10% des chemins réels. L'introduction des préférences de chaque AS en fonction des accords d'interconnexion ne permet donc pas d'améliorer les résultats, bien au contraire. Ceci s'explique par deux raisons :

1. Dans le modèle *G-LP* on préfère les clients d'un AS à ses peers et ses peers à ses fournisseurs. Ces deux règles ne sont pas forcément justes. D'autre part, des valeurs différentes peuvent avoir été indiquées pour deux voisins avec le même accord économique.
2. Les erreurs dans les solutions du problème d'inférence des accords d'interconnexion introduisent des chemins sans vallées invalides et des *Local Pref* erronés.

Le dernier AS à traverser avant d'atteindre un AS destination est inféré pour 70% à 80% des couples origine-destination (indicateur *DERNIER-AS*), même pour le modèle *G-LP*. Ce bon résultat s'explique par la présence de nombreux AS stub ou terminaux pour lesquels il existe un faible nombre d'AS voisins.

D'après l'indicateur *PREMIER-AS*, il semble que le phénomène qui limite la justesse de notre inférence est le choix du premier AS à traverser pour joindre une destination. Lorsque l'on fixe des préférences dans le modèle *G-LP*, l'indicateur *PREMIER-AS* chute de 52% à 17% pour la tomographie P_{2003} et de 40% à 8% pour la tomographie P_{2005}. L'introduction des valeurs de *Local Pref* produit les effect suivants sur les indicateurs :

- un impact très négatif sur le pourcentage de chemins correctement inférés *EGAL*,
- un faible impact sur l'indicateur *DERNIER-AS,*
- un impact très négatif sur l'indicateur *PREMIER-AS,*
- des chemins inférés de taille trop importante (indicateur *TROP-LONG*) pour plus de 60% des couples origine-destination alors que très peu de chemins sont dans ce cas lorsqu'il sont calculés avec le modèle *G-AS* (sans *Local Pref*),
- des chemins avec un *Local Pref* strictement meilleurs sont calculés pour au moins 80% des couples origine-destination (indicateur *TROP-PREF*).

On peut donc expliquer les résultats médiocres du modèle *G-LP* par l'existence de détours sans vallée dans le graphe. Ces détours sont des chemins plus longs mais avec un meilleur *Local Pref* que le chemin réel. Autrement dit, chaque accord d'interconnexion fixé à une valeur incorrecte introduit une valeur potentiellement erronée pour le *Local Pref* de l'arc dans le graphe. Si ce *Local Pref* est plus grand, alors toute route commençant par cet arc sera préférrée à une autre route commençant par un arc avec un *Local Pref* plus faible, et ce même si elle plus longue en nombre de bonds. Réciproquement, si ce *Local Pref* est plus petit, alors tout autre route commençant par un arc qui porte un Local Pref plus grand sera toujours préférée aux routes commençant par l'arc avec l'accord mal inféré. Quelques fautes suffisent dans l'algorithme d'inférence pour que l'algorithme calcule des chemins très souvent différents du chemin réel. Ces chemins sont en général plus longs (60%) que le chemin réel et avec un *Local Pref* strictement plus grand (80%).

b) Comparaison des chemins inférés pour chaque AS source

On s'intéresse aux valeurs des indicateurs détaillées pour les chemins de chaque AS source. Sur la figure 4.9 on montre les résultats pour le modèle *G-AS.* Pour la tomographie P_{2005}, il y a 3 fois plus d'AS sources que pour la tomographie P_{2003}. Un plus large panel d'AS sources est donc couvert par la tomographie P_{2005}. Sur la figure 4.9(b) les indicateurs prennent des valeurs sensiblement différentes en fonction de l'AS source considéré. Le modèle *G-AS* est une transformation globale du graphe. Les résultats de chaque AS source sont donc normalement inter-dépendants. Ceci indique que la proportion de che-

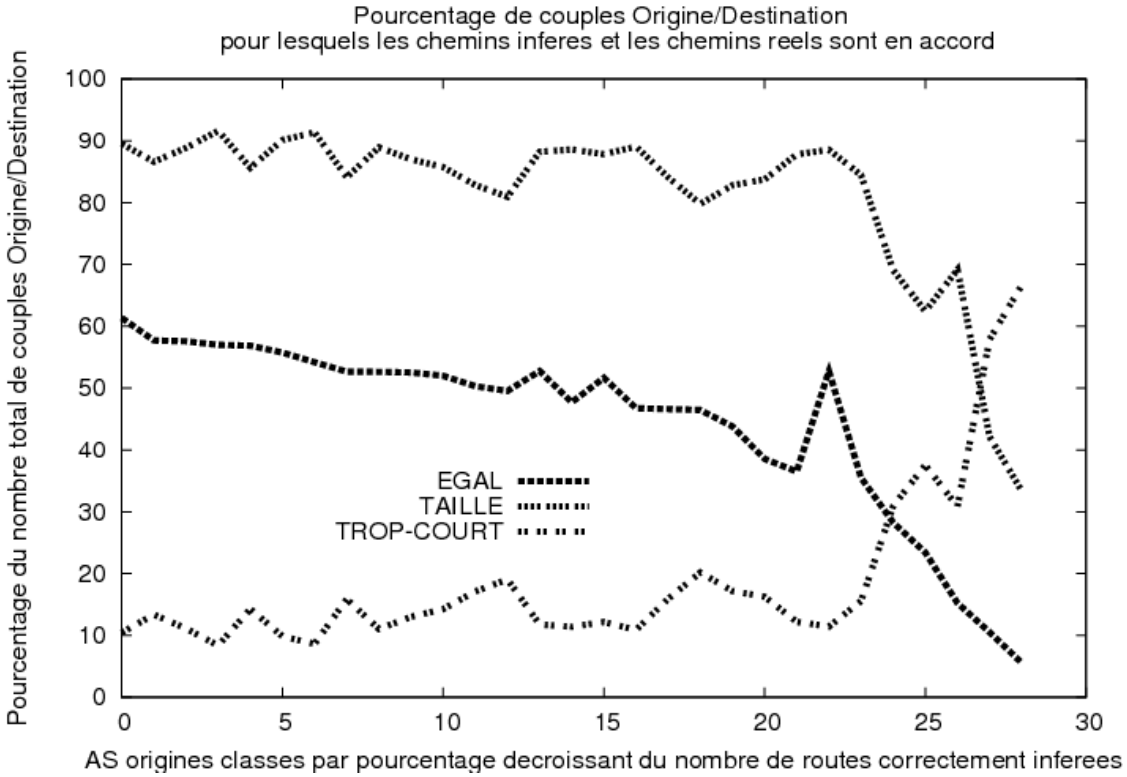
mins correctement inférés pour un AS source dépend de la justesse globale de l'inférence des accords d'interconnexion et des accords aux alentours de l'AS. Il y a plus de 98% de chemins réels valides pour les tomographies grâce aux algorithmes du chapitre 3. Mais le problème d'inférence des accords d'interconnexion est sous-déterminé et il existe un nombre exponentiel de solutions qui valident autant de chemins. La sélection du chemin réel est possible avec le modèle *G-AS* , sauf si un autre chemin plus court existe. On remarque que de 70% à 90% des chemins calculés pour une majorité d' AS source sont de taille identique aux chemins réels. Seulement 40% à 50% des chemins calculés ont le même premier AS. Les chemins calculés avec le modèle *G-AS* semblent donc être des chemins alternatifs. En résumé, pour environ 3 AS sources sur 4, le modèle *G-AS* est capable d'inférer 1 chemin réel sur 3 et de déterminer la taille en nombre d'AS pour 3 chemins sur 4.

Les valeurs des indicateurs détaillées par AS source sont indiqués pour le modèle *G-PREFIX* sur la figure 4.10 . Pour 3 AS sources sur 4, il est possible d'inférer 70 des chemins vers chaque préfixe. Pour la majorité des AS sources, les routes calculées sont de taille identique au chemin réel pour plus de 4 préfixes sur 5. La prise en compte des matrices de politique d'annonce dans le modèle de graphe permet donc d'obtenir une meilleure inférence des chemins entre AS et préfixes que ne le permet le modèle *G-AS* entre les AS.

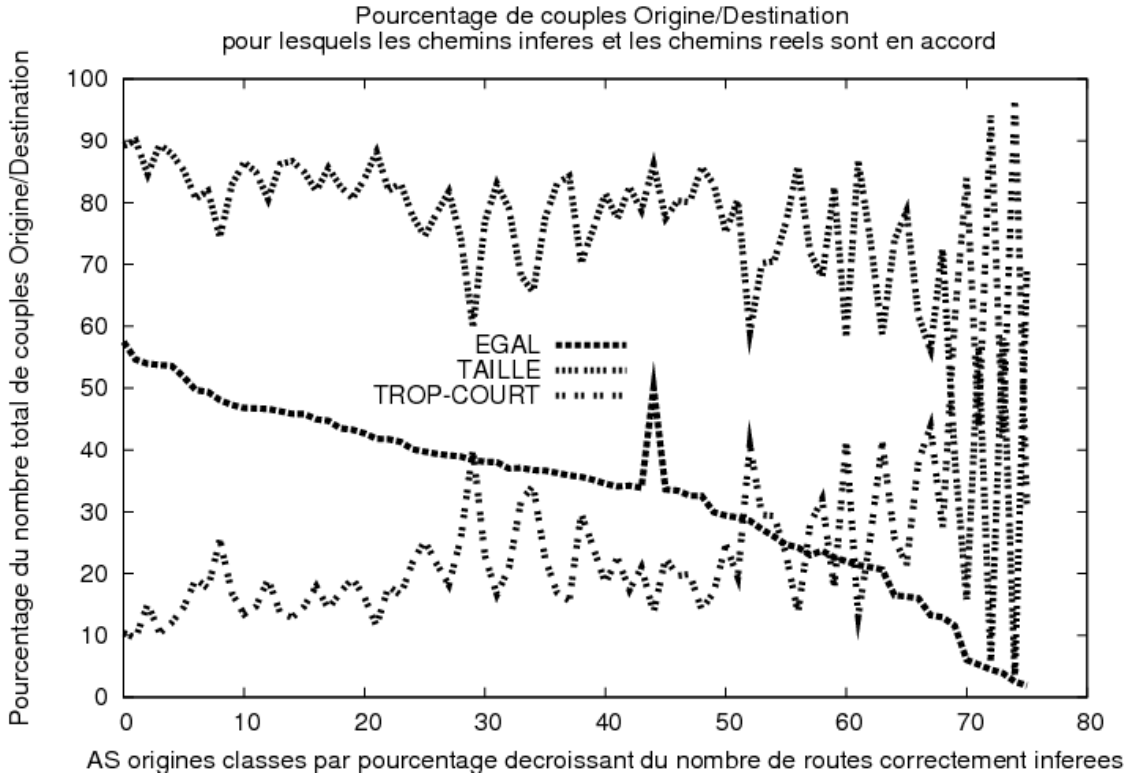
L'effet de l'introduction des valeurs de *Local Pref* pour le calcul des chemins inter-AS est reporté sur la figure 4.11. Pour savoir si la valeur du *Local Pref* d'un chemin calculé avec le modèle *G-LP* est la même que valeur du *Local Pref* du chemin réel, on introduit l'indicateur suivant :

MEME-ACCORD : l'accord d'interconnexion entre l'AS source et le premier AS du chemin reel est le même que l'accord d'interconnexion entre l'AS source et le premier AS du chemin calculé avec le modèle. Le chemin calculé avec le modèle a donc dans ce cas un *Local Pref* égal à celui du chemin réel.

L'indicateur *MEME-ACCORD* indique qu'une majorité d'AS sélectionnent des routes avec un Local Pref différent de celui du chemin réel. Les faibles valeurs de cet indicateur et les grandes valeurs des indicateurs *TROP-PREF* et *TROP-LONG* montrent que des détours sont souvent calculés par l'algorithme. Ceci explique donc bien les résultats médiocres introduits par les mauvaises préférences et les chemins non valides en réalité mais sans vallée pour notre solution des accords d'interconnexion.

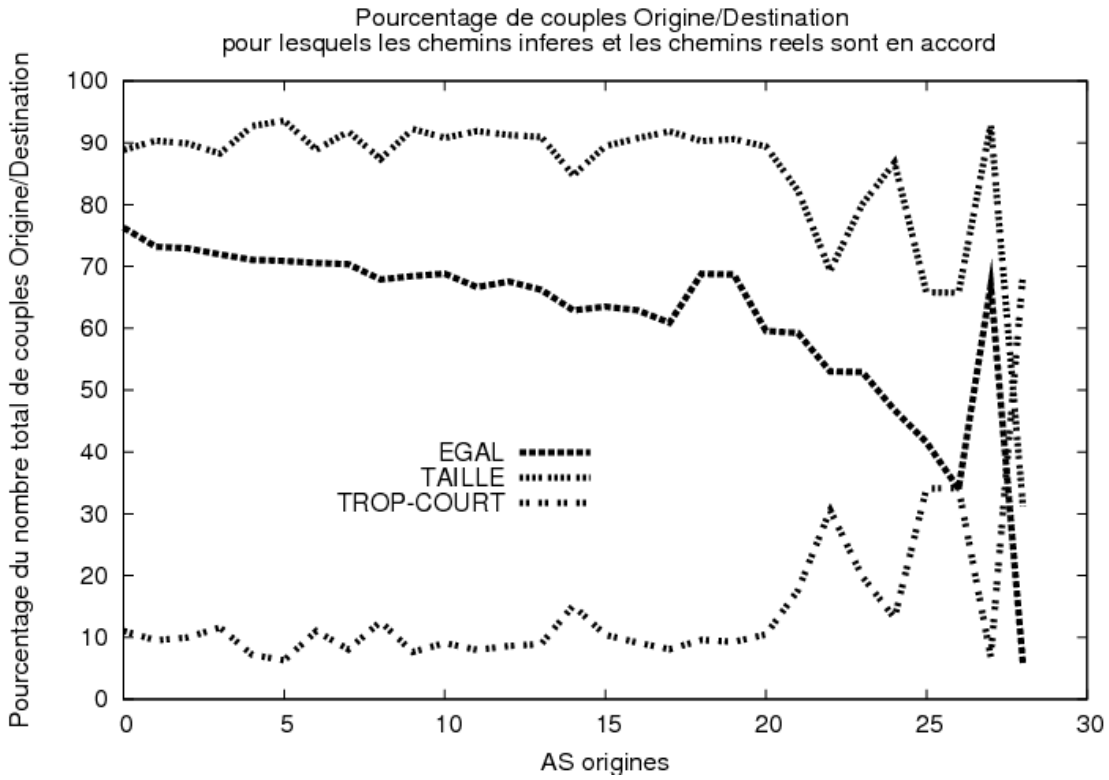


(a) modèle $G - AS$ utilisé avec la tomographie P_{2003}



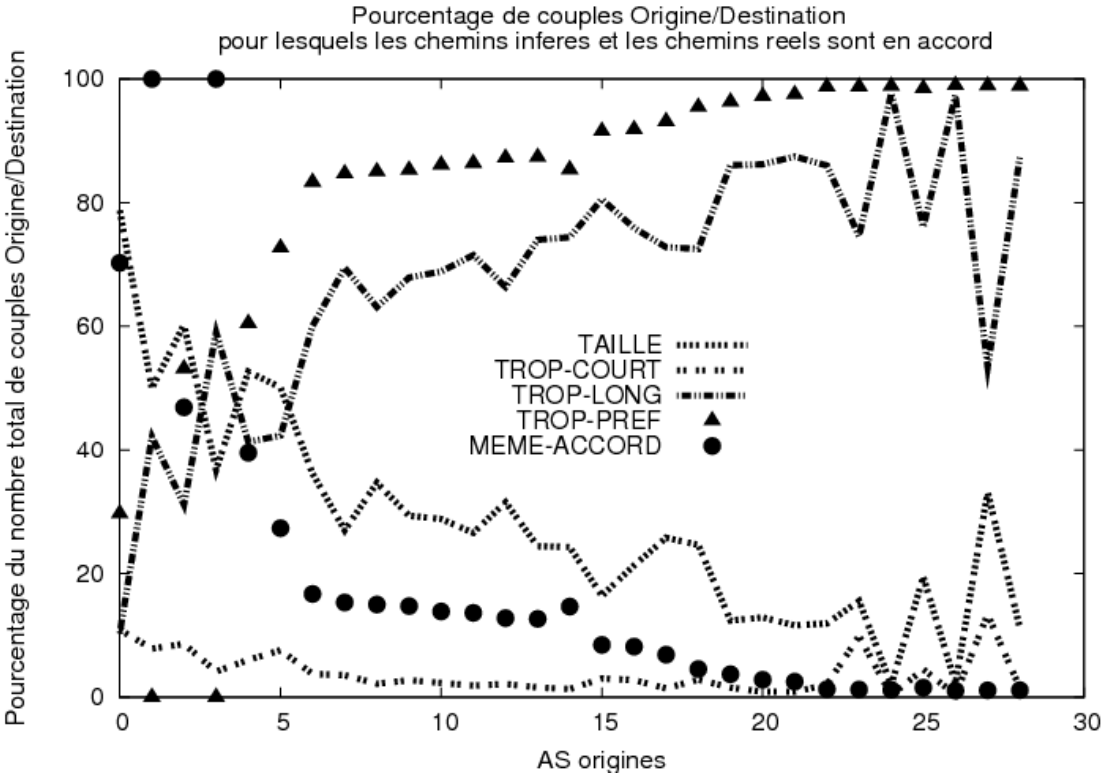
(b) modèle $G - AS$ utilisé avec la tomographie P_{2005}

FIG. 4.9 – Evaluation du modèle *G-AS* pour les chemins de chaque AS source

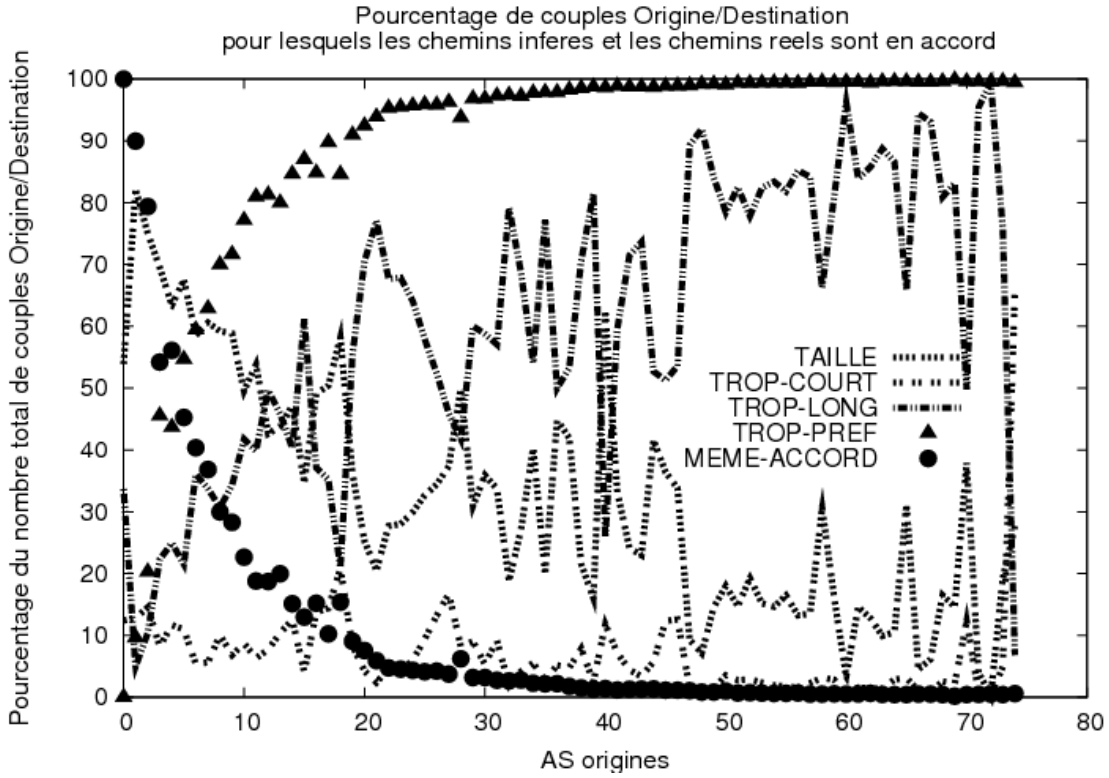


(a) modèle $G - PREFIX$ utilisé avec la tomographie P_{2003}

FIG. 4.10 – Evaluation du modèle *G-PREFIX* pour les chemins de chaque AS source vers les préfixes



(a) modèle $G - LP$ utilisé avec la tomographie P_{2003}

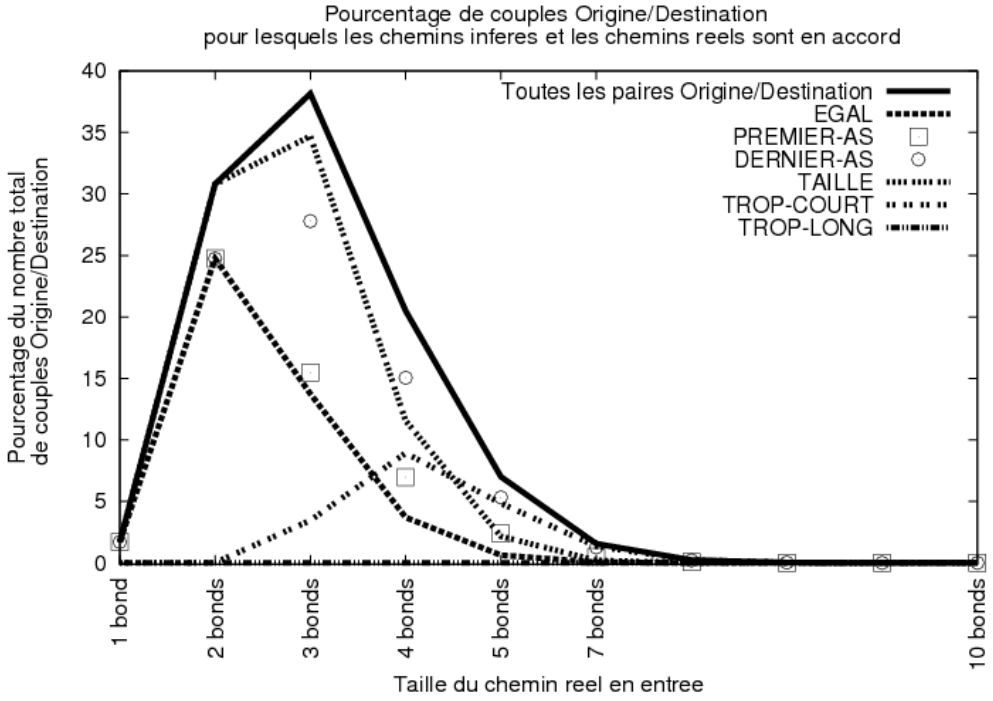


(b) modèle $G - LP$ utilisé avec la tomographie P_{2005}

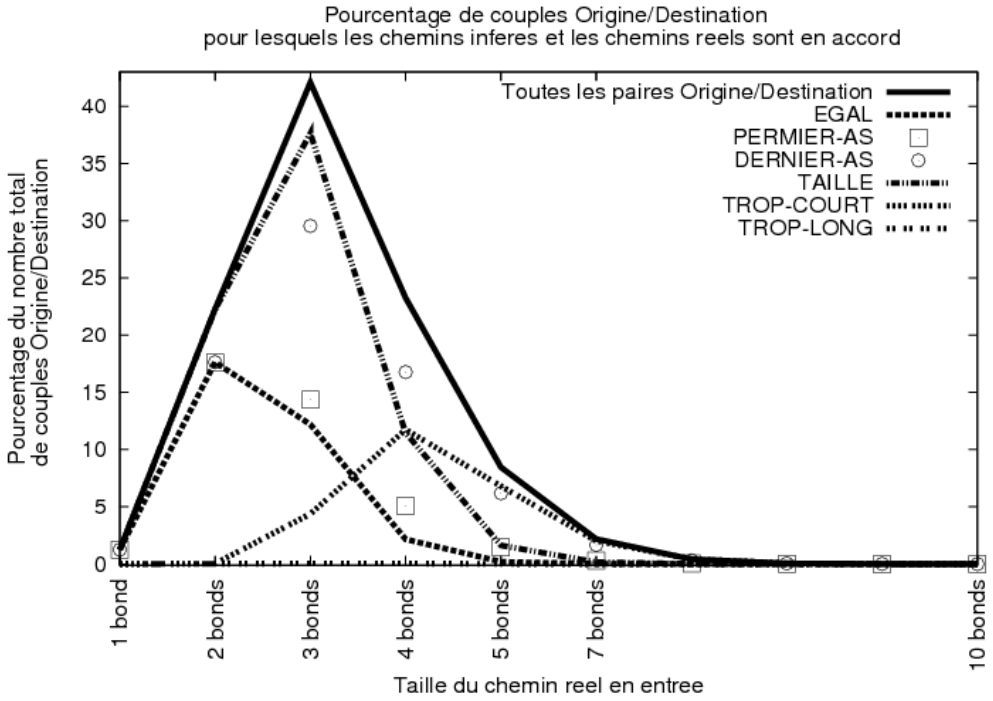
FIG. 4.11 – Evaluation du modèle *G-LP* pour les chemins de chaque AS source

c) Inférence en fonction de la taille des chemins

Dans le modèle *G-AS*, on prend seulement en compte le nombre de bonds et les vallées. La figure 4.12 montre que ses performances pour l'inférence diminuent considérablement avec la taille des chemins en entrée. Ceci indique que la modélisation sans *Local Pref* est insuffisante pour reconnaitre le chemin réel le plus court. Ce modèle permet d'inférer toutefois des chemins de même taille que les chemins réels pour environ 80% des paires origine-destination et ce quelque soit la taille des chemins en entrée. Les chemins calculés par l'algorithme avec le modèle *G-AS* sont donc bien des chemins alternatifs.
D'après la figure 4.13, le modèle *G-LP* permet d'inférer des chemins de même taille que les chemins réels pour un maximum de 31.7% des couples origine-destination. Les performances de l'inférence avec ce modèle sont très médiocres lorsque la taille des chemins augmente. Cela montre que l'algorithme utilisé avec le modèle *G-LP* calcule des chemins interdomaines qui sont des détours.
Le modèle *G-PREFIX* correspond au modèle *G-AS* avec la prise en compte des matrices de politique d'annonce. Il permet d'inférer le chemins réel sélectionné pour les comparaisons pour 60% des couples origine-destination (voir figure 4.14). De plus, les performances de l'inférence ne diminue pas autant significativement avec la taille des chemins que pour les autres modèles. Les chemins calculés sont de même taille que les chemins réels pour 85% des couples AS et préfixe. De plus, le modèle donne un compromis entre le calcul de chemins plus courts et de chemins plus longs.

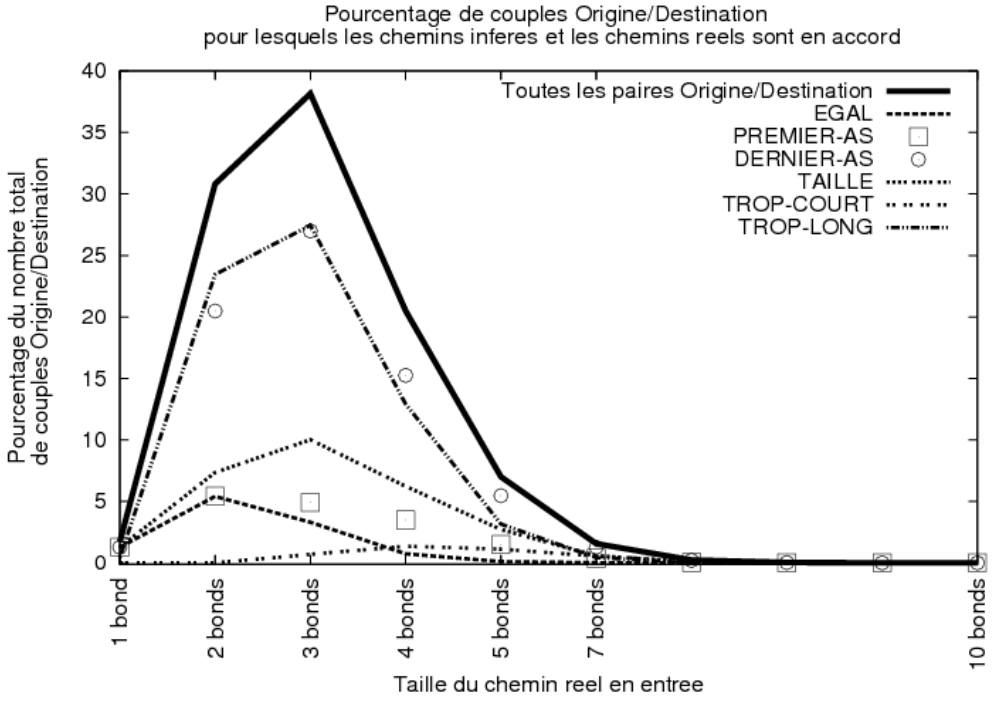


(a) modèle $G - AS$ utilisé avec la tomographie P_{2003}

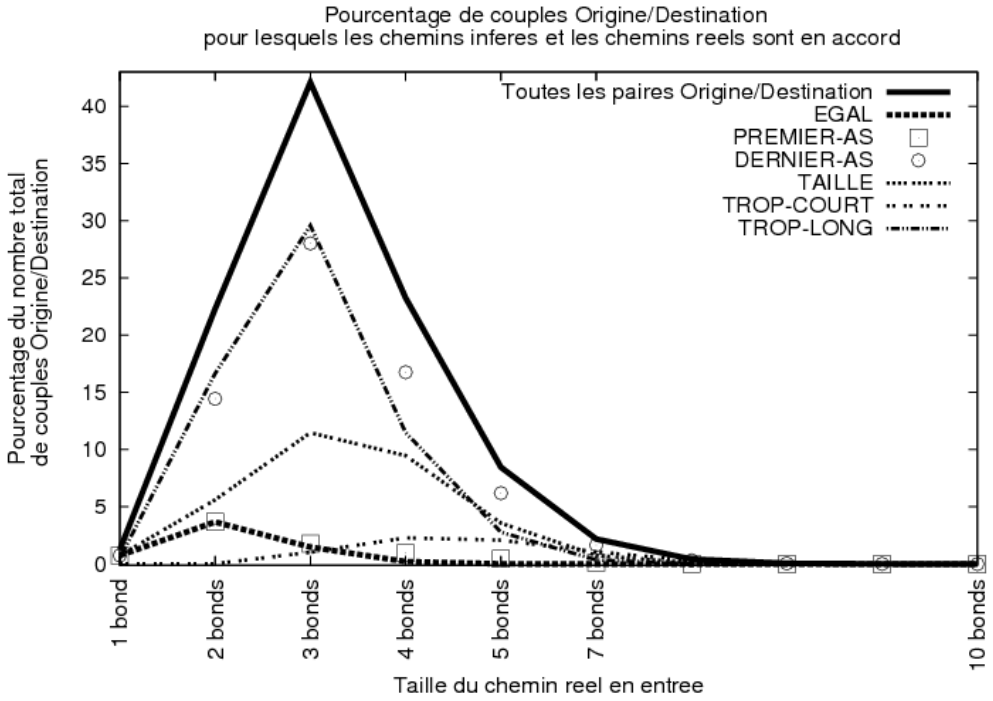


(b) modèle $G - AS$ utilisé avec la tomographie P_{2005}

FIG. 4.12 – Inférence des chemins inter-AS en fonction de la taille des chemins réels.

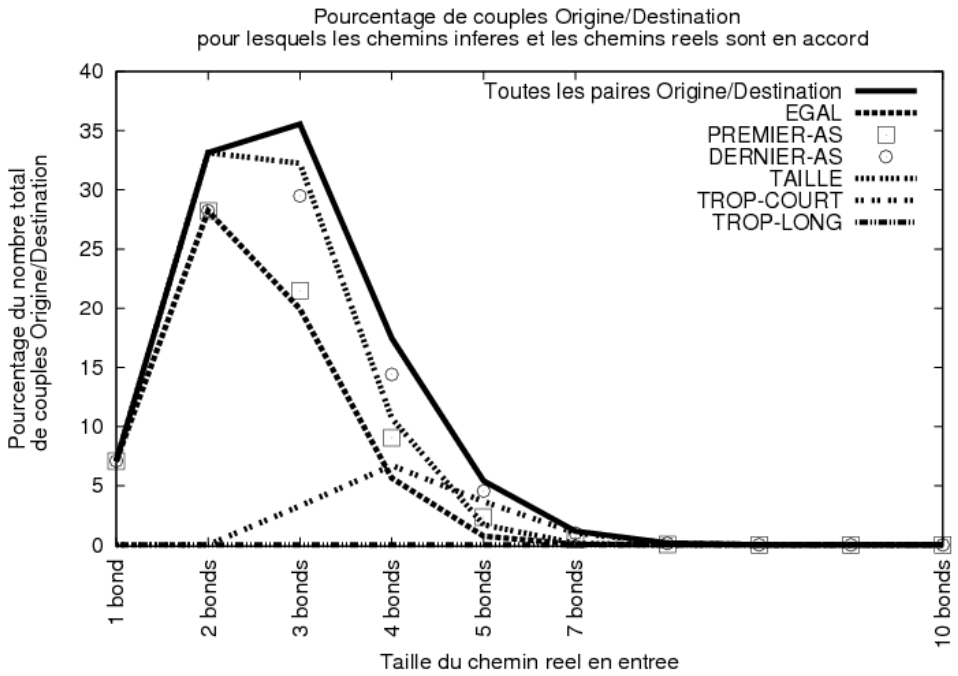


(a) modèle $G - LP$ utilisé avec la tomographie P_{2003}



(b) modèle $G - LP$ utilisé avec la tomographie P_{2005}

FIG. 4.13 – Inférence des chemins d'AS en fonction de la taille des chemins réels.



(a) modèle $G - PREFIX$ utilisé avec la tomographie P_{2003}

FIG. 4.14 – Inférence des chemins d'AS vers les préfixes en fonction de la taille des chemins réels.

La taille des chemins pour joindre un AS ou un préfixe est donc l'information la plus correctement inférés par nos modèle du routage interdomaine. Cela permet par exemple d'évaluer la distribution en nombre de bonds d'un AS vers le reste de l'Internet. L'inférence des chemins exact souffre de notre modélisation à la granularité AS et de la sous-détermination du problème d'inférence des accords d'interconnexion. Dans notre modélisation, tout se passe comme si chaque AS est formé d'un seul routeur. Ceci qui est insuffisant pour reconstituer la diversité de tous les routages BGP dans l'Internet. Notre approche algébrique pourrait servir à exploiter un algorithme de K-plus courts chemins modélisant alors la présence de plusieurs routes dans un même AS. Cela ne correspond pas exactement à un modèle avec plusieurs routeurs par AS car le nombre de routes transmises par un AS à un autre AS dépend du nombre de sessions eBGP entre les deux AS et des choix concurrents des routeurs.

Notre modèle a l'avantage de pouvoir prendre en compte des valeurs de Local Pref arbitraires grâce à l'utilisation combinée du graphe étendu des AS et de la strucutre algébrique des poids de chemins. La détermination de valeurs *Local Pref* prometteuses pour améliorer les résultats de l'inférence est un problème difficile. D'autres investigations sont nécessaire pour inférer cette information afin de l'exploiter correctement.

Conclusion

Dans ces travaux de thèse, on s'est intéressé à déterminer les politiques de routage interdomaines dans l'Internet. Après avoir rappelé les modes d'interconnexion économiques entre opérateurs et discuté de quelques-uns des enjeux de l'interconnexion dans l'Internet, on a introduit les principes de fonctionnement du protocole BGP ainsi que la notion de politique de routage interdomaine. L'inférence de ces politiques est au coeur de nombreuses problématiques comme la simulation du routage BGP, la régulation des marchés d'interconnexion ou encore l'aide à la décision pour un opérateur dans ses négociations commerciales. Un état de l'art concernant la connaissance et la mesure de la topologie de l'Internet a été réalisé. Dans l'inférence des politiques de routage économiques, on a identifié trois problématiques importantes : la mesure des politiques de routage, l'inférence des accords d'interconnexion économiques entre AS et la modélisation des routages BGP. Dans le chapitre 2, une nouvelle méthode de mesure de la topologie interdomaine a été introduite. Le procédé exploite les tables de routage BGP de routeurs sondes pour former un ensemble de routes appelé tomographie BGP. Les tomographies, construites avec plusieurs tables successives pour les mêmes sondes, permettent de sélectionner un sous-ensemble de chemins de routage homogènes et stables dans le temps. Les routages BGP obtenus par ce procédé sont analysés grâce à de nouveaux objets appelés matrices de politique de routage. Ces matrices sont des éléments topologiques élementaires. Elles représentent un compromis entre une tomographie (vision chemin) et la topologie inter-AS (vision graphe). Elles constituent un point clé des résultats de cette thèse car elles offrent la possibilité d'étudier le problème d'inférence des accords économiques dans le chapitre 3 et d'aborder le problème de modélisation du routage inter-AS dans le chapitre 4. L'analyse des matrices de politique de routage montre qu'elles sont majoritairement fixées de sorte à obtenir des routages par groupe d'AS. Ces routages par groupe témoignent de l'influence du caractère économique des interconnexions entre AS.

Dans le chapitre 3, on a déterminé les accords économiques entre systèmes autonomes

en étudiant le problème MaxTOR. Ce dernier est un problème inverse d'optimisation qui cherche une affectation des contrats économiques sur les liaisons inter-AS pour qu'un maximum de chemins observés respectent les règles de routage économique usuelles. On montre qu'il est a priori plus juste de résoudre le problème d'inférence des accords en traitant le problème MaxTOR3. Ce problème utilise en entrée les sous-chemins élémentaires de taille 3 (cette formulation avait été exploitée de façon plus simple dans la littérature). Nos contributions dans cette thèse pour le problème d'inférence des accords portent sur sa modélisation (au travers de nouvelles contraintes), sa formulation ainsi que sur les méthodes permettant de le résoudre. Une formulation basée sur le modèle de satisfaction de contraintes (CSP) a d'abord été introduite. Cette formulation MaxCSP est résolue avec une métaheuristique Tabou qui exploite un voisinage restreint des solutions réalisables. Le problème MaxTOR a été par la suite généralisé en une nouvelle classe de problèmes appelée MaxVCAP. Un procédé est introduit pour obtenir une formulation en programme linéaire en nombres entiers. Ce procédé a été amélioré pour le cas particulier de MaxTOR3. Le problème a pu être résolu pour des instances réelles de très grande taille. On remarquera que le procédé de formulation linéaire ainsi que les nouvelles coupes polyédrales introduites pour la résolution peuvent être exploités dans certains autres problèmes MaxVCAP. A la fin du chapitre, on démontre la pertinence de nos solutions en les comparant à celles des algorithmes de la littérature et en vérifiant les résultats pour plusieurs ensembles de routes BGP. Les solutions, confrontées avec les accords connus de l'AS 5511, sont partiellement validées. Celles produites par la métaheuristique Tabou vérifient la quasi-totalité des contraintes et concordent à plus de 90% avec les accords connus. La méthode exacte permet d'obtenir des solutions optimales pour MaxTOR3. Ces solutions ne semblent pas aussi bonnes que les solutions de la métaheuristique car elles contiennent des cycles d'AS clients. Mais grâce à la formulation exacte, il est possible d'ajouter de nouvelles contraintes au problème pour augmenter la fidélité des solutions. Notre méthodologie et nos formulations peuvent être exploités pour résoudre d'autres problèmes comme par exemple le problème de conception de topologie iBGP dans un système autonome.

Dans le chapitre 4, un procédé de transformation du graphe des AS a été proposé. Le graphe étendu des AS obtenu permet de calculer directement des chemins sans vallée. Le routage BGP a été modélisé à l'échelle de détail des AS grâce à une algèbre de poids. Cette algèbre permet de modéliser le processus de décision BGP en prenant en compte les décisions locales de chaque AS (avec un attribut "Local Preference") et le nombre

de bonds dans un chemin. La structure algébrique ne respecte pas certaines propriétés usuelles comme l'existence d'un élément neutre à droite ou la distributivité à gauche. On montre néamoins que le problème des plus courts chemins BGP sans vallées peut se résoudre en temps polynomial (on prouve la convergence l'algorithme de Dijkstra généralisé). L'algorithme qui calcule les chemins inter-AS est utilisé dans le sens "n sources" vers une destination. On montre enfin comment ajouter des préfixes dans le modèle de graphe pour tenir compte des sélectivités des annonces de chaque AS. Notre modélisation permet dans certains cas l'inférence de plus de 60% des chemins interdomaines vers les AS ou vers les préfixes. De plus, les chemins calculés avec l'algorithme sont de la même taille que les chemins des tomographies pour plus de 80% des couples origine-destination. La prise en compte des *Local Pref* de chaque AS, possible grâce à notre modélisation, ne permet pas d'améliorer les résultats. De plus amples investigations sont nécessaires pour déterminer de meilleures valeurs pour ces attributs *Local Pref*.

Bibliographie

[1] *On the complexity of vertex and facet enumeration for convex polytopes*. PhD thesis, School of Computer Science, McGill University, 1997.

[2] draft-ietf-grow-mrt-03.txt : Mrt routing information export format, 2006.

[3] E. Davies A. Doria. Analysis of idr requirements and history. In *draft-irtf-routing-history-02.txt*, 10 2005.

[4] C. Alaettinoglu, T. Bates, E. Gerich, D. Karrenberg, D. Meyer, M. Terpstra, and C. Villamizar. RFC 2280 : Routing Policy Specification Language (RPSL), January 1998. Status : PROPOSED STANDARD.

[5] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). Internet Engineering Task Force : RFC 2622, June 1999.

[6] Lisa Amini and Henning Schulzrinne. Observations from router-level internet traces. In *DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling*, Piscataway, New Jersey, February 2002.

[7] Lisa Amini, Anees Shaikh, and Henning Schulzrinne. Issues with inferring internet topological attributes. *Computer Communications*, 27(6) :557–567, 2004.

[8] Kapil Bajaj. Cooperative topology discovery within an autonomous system, 2001.

[9] Kapil Bajaj and D. Manjunath. Intranet topology discovery using untwine, 2003.

[10] J. M. Barcelo, J. I. Nieto-Hipolito, and Jorge Garcia-Vidal. Study of tnternet autonomous system interconnectivity from bgp routing tables. *Comput. Networks*, 45(3) :333–344, 2004.

[11] Paul Barford, Azer Bestavros, John Byers, and Mark Crovella. On the marginal utility of network topology measurements. In *IMW '01 : Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, pages 5–17, New York, NY, USA, 2001. ACM Press.

[12] Barricelli and Nils Aall. Esempi numerici di processi di evoluzione. *Methodos*, pages 45–68, 1954.

[13] Anindya Basu, Chih-Hao Luke Ong, April Rasala, F. Bruce Shepherd, and Gordon Wilfong. Route oscillations in i-bgp with route reflection. In *SIGCOMM '02 : Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 235–247, New York, NY, USA, 2002. ACM Press.

[14] G. Di Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems, 2002.

[15] Giuseppe Di Battista, Thomas Erlebach, Alexander Hall, Maurizio Patrignani, Maurizio Pizzonia, , and Thomas Schank. Computing the types of the relationships between autonomous systems.

[16] Giuseppe Di Battista, Tiziana Refice, and Massimo Rimondini. How to extract bgp peering information from the internet routing registry. In *MineNet 06 : Proceedings of the 2006 SIGCOMM workshop on Mining network data*, pages 317–322, New York, NY, USA, 2006. ACM Press.

[17] Rudolf Bayer. Symmetric binary b-trees : Data structure and maintenance algorithms. *Acta Inf.*, 1 :290–306, 1972.

[18] Ginestra Bianconi, Guido Caldarelli, and Andrea Capocci. Number of h-cycles in the internet at the autonomous systems level, 2003.

[19] Lumeta Corp. Bill Cheswick. Internet mapping project.

[20] Scott Bradner. A short history of the internet. North American Network Operators' Group (NANOG) presentation, February 2004.

[21] Yuri Breitbart, Minos N. Garofalakis, Cliff Martin, Rajeev Rastogi, S. Seshadri, and Abraham Silberschatz. Topology discovery in heterogeneous IP networks. In *INFOCOM (1)*, pages 265–274, 2000.

[22] Tian Bu, Lixin Gao, and Don Towsley. On characterizing bgp routing table growth. *Comput. Networks*, 45(1) :45–54, 2004.

[23] Hal Burch. Eliminating machine duplicity in traceroute-based eliminating machine duplicity in traceroute-based internet topology measurement. Technical report, School of Computer Science, Carnegie Mellon University, 2002.

[24] Hal Burch and Bill Cheswick. Mapping the internet. *Computer*, 32(4) :97–98,102, 1999.

[25] Matthew Caesar and Jennifer Rexford. Bgp routing policies in isp networks. Technical Report UCB/CSD-05-1377, EECS Department, University of California, Berkeley, 2005.

[26] H. Chang, S. Jamin, and W. Willinger. Inferring as-level internet topology from router-level path traces, 2001.

[27] H. Chang, S. Jamin, and W. Willinger. What causal forces shape internet connectivity at the as-level, 2003.

[28] Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, and Walter Willinger. Towards capturing representative as-level internet topologies. In *SIGMETRICS '02 : Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 280–281, New York, NY, USA, 2002. ACM Press.

[29] Hyunseok Chang, Ramesh Govindan, Sugih Jamin, Scott J. Shenker, and Walter Willinger. Towards capturing representative as-level internet topologies. *SIGMETRICS Perform. Eval. Rev.*, 30(1) :280–281, 2002.

[30] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. The origin of power laws in internet topologies revisited. In *INFOCOM*, 2002.

[31] Monk T. E. Claffy K. and McRobb D. *Internet tomography.* Nature web matters, January 1999.

[32] D.D. Clark. Policy routing in internet protocols. RFC 1102, May 1989.

[33] Aaron Clauset and Cristopher Moore. Traceroute sampling makes random graphs appear to have power law degree distributions, 2003.

[34] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71 : Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM Press.

[35] Luca Dall'Asta, Ignacio Alvarez-Hamelin, Alain Barrat, Alexei Vazquez, and Alessandro Vespignani. A statistical approach to the traceroute-like exploration of networks : theory and simulations. *LNCS*, 3405 :140, 2005.

[36] Peter Jeavons David Cohen, Martin Cooper and Andrei Krokhin. Identifying efficiently solvable cases of max csp. In *STACS*, pages 152–163. Lecture Notes in Computer Science 2996, 2004.

[37] Peter Jeavons David Cohen, Martin Cooper and Andrei Krokhin. Supermodular functions and the complexity of max csp. *Discrete Applied Mathematics 149*, pages 53–72, 2005.

[38] C. de Launois, B. Quoitin, and O. Bonaventure. Leveraging network performances with ipv6 multihoming and multiple provider-dependent aggregatable prefixes, 2005.

[39] Stefan Dehm and Steven Sinha. Incompleteness of the internet topology : A structural analysis. Technical report, 2001.

[40] X. Dimitropoulos, D. Krioukov, B. Huffaker, kc claffy, and G. Riley. Inferring as relationships : Dead end or lively beginning ? In *Submitted to HotNets-III*, 2004.

[41] Xenofontas Dimitropoulos, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Young Hyun, kc claffy, and George Riley. As relationships : Inference and validation, 2006.

[42] Xenofontas Dimitropoulos, Dmitri Krioukov, and George Riley. Revisiting internet as-level topology discovery. *LNCS*, 3431 :177, 2005.

[43] Benoit Donnet, Timur Friedman, and Mark Crovella. Improved algorithms for network topology discovery. In *PAM*, pages 149–162, 2005.

[44] David Eppstein. Finding the k shortest paths. In *Proc. 35th Symp. Foundations of Computer Science*, pages 154–165. IEEE, 1994.

[45] David Eppstein. Finding the k shortest paths. *SIAM J. Comput.*, 28(2) :652–673, 1999.

[46] T. Erlebach, A. Hall, and T. Schank. Classifying customer-provider relationships in the internet, 2002.

[47] Thomas Erlebach. Autonomous systems in the internet : A potential subject for studying self-* aspects. In *SELF-STAR : International Workshop on Self-* Properties in Complex Information Systems*, University of Bologna Residential Center Bertinoro (Forli), Italy, 2004.

[48] Michel Gondran et Michel Minoux. *Graphes, dioïdes et semi-anneaux. Nouveaux modèles et algorithmes.* Paris : Éditions TEC & DOC, 2002.

[49] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99 : Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262. ACM Press, 1999.

[50] N. Feamster, H. Balakrishnan, and J. Rexford. Some foundational problems in interdomain routing. In *ACM SIGCOMM Workshop on Hot Topics in Networking (HotNets-III)*, November 2004.

[51] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. In *ACM SIGCOMM Computer Communication Review*, pages 19–30, October 2003.

[52] Nick Feamster, David G. Andersen, Hari Balakrishnan, and M. Frans Kaashoek. Measuring the effects of internet path faults on reactive routing. In *SIGMETRICS '03 : Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 126–137, New York, NY, USA, 2003. ACM Press.

[53] Nick Feamster, Jared Winick, and Jennifer Rexford. A model of bgp routing for network engineering. In Edward G. Coffman Jr., Zhen Liu, and Arif Merchant, editors, *SIGMETRICS*, pages 331–342. ACM, 2004.

[54] L. C. Freeman. Centrality in social networks : Conceptual clarification. *Social Networks*, 1 :215–239, 1979.

[55] ftp ://ftp.apnic.org/public/whois data/. Apnic ftp repository.

[56] ftp ://ftp.ripe.net/ripe/stats. Rir statistics exchange.

[57] Komei Fukuda. Frequently asked questions in polyhedral computation, June 2004.

[58] Komei Fukuda, Thomas M. Liebling, and F. Margot. Analysis of backtrack algorithms for listing all vertices and all faces of a convex polyhedron. *Comput. Geom. Theory Appl.*, 8(1) :1–12, 1997.

[59] Komei Fukuda and Vera Rosta. Combinatorial face enumeration in convex polytopes. *Comput. Geom. Theory Appl.*, 4(4) :191–198, 1994.

[60] L. Gao. On inferring autonomous system relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, November 2000.

[61] L. Gao and F. Wang. The extent of as path inflation by routing policies. 2002.

[62] Lixin Gao, Timothy Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *INFOCOM*, pages 547–556, 2001.

[63] Lixin Gao and Jennifer Rexford. Stable internet routing without global coordination. *IEEE/ACM Trans. Netw.*, 9(6) :681–692, 2001.

[64] Jim Gast and Paul Barford. Representing the internet as a succinct forest. *Comput. Networks*, 45(1) :35–44, 2004.

[65] Ewgenij Gawrilow and Michael Joswig. Polymake : an approach to modular software design in computational geometry. In *SCG '01 : Proceedings of the seventeenth annual symposium on Computational geometry*, pages 222–231, New York, NY, USA, 2001. ACM Press.

[66] Z. Ge, D. Figueiredo, S. Jaiwal, and L. Gao. On the hierarchical structure of the logical Internet graph. SPIE ITCOM, August 2001.

[67] Fred Glover and M. Laguna. Tabu search. In C. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, Oxford, England, 1993. Blackwell Scientific Publishing.

[68] Daniel Golding. Tutorial abstract : Routing policy implementation guide. NANOG24 Meeting, Feruary 2002.

[69] Ramesh Govindan. The state of internet topology research, 2001.

[70] Ramesh Govindan and Anoop Reddy. An analysis of internet inter-domain topology and route stability. In *INFOCOM (2)*, pages 850–857, 1997.

[71] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *IEEE INFOCOM 2000*, pages 1371–1380, Tel Aviv, Israel, March 2000. IEEE.

[72] T. Griffin and G. Huston. BGP Wedgies. RFC 4264 (Informational), November 2005.

[73] T. Griffin, A. Jaggard, and V. Ramachandran. Design principles of policy languages for path vector protocols, 2003.

[74] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10.2 :232–243, April 2002.

[75] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. Policy disputes in path-vector protocols. In *ICNP '99 : Proceedings of the Seventh Annual International Conference on Network Protocols*, page 21, Washington, DC, USA, 1999. IEEE Computer Society.

[76] Matthias Grossglauser and Balachander Krishnamurthy. Looking for science in the art of network measurement. In *IWDC '01 : Proceedings of the Thyrrhenian*

International Workshop on Digital Communications, pages 524–535, London, UK, 2001. Springer-Verlag.

[77] C. Gunduz and B. Yener. Accuracy and sampling trade-offs for inferring internet router graph. Technical report, Rensselaer Polytechnic Institute, Department of Computer Science, July 2003.

[78] Anwar M. Haneef and Santhosh R. Thampuran. Sibre inferring sibling relationships on the internet, 2000.

[79] Fang Hao and Pramod Koppol. An internet scale simulation setup for bgp. *SIGCOMM Comput. Commun. Rev.*, 33(3) :43–57, 2003.

[80] Sue Hares. Bgp attack trees : Real world examples. In *NANOG 28*, 2003.

[81] Susan Hares. 15 years of policy routing. In *NANOG 30*, february 2004.

[82] Juan Ivan Nieto Hipolito. *Analysis of the Internet Topology and generation models.* PhD thesis, UNIVERSIDAD POLITECNICA DE CATALUNYA, July 2005.

[83] Joseph D. Horton and Alejandro Lopez-Ortiz. On the number of distributed measurement points for network tomography. In *IMC '03 : Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 204–209, New York, NY, USA, 2003. ACM Press.

[84] http ://bgpview.6test.edu.cn/bgp view/index.shtml. Cernet bgp view global internet.

[85] http ://puck.nether.net/netops/nocs.cgi. Network operations centers list.

[86] http ://sk aslinks.caida.org/data/. Caida's macroscopic topology as adjacencies.

[87] http ://tocai.dia.uniroma3.it/ irr_analysis/analysis_data/.

[88] http ://topology.eecs.umich.edu/data.html. The origin of power-laws in internet topologies revisited.

[89] http ://www-net.cs.umass.edu/ ratton/AS/. Hierarchical as classification program.

[90] http ://www.caida.org/analysis/topology/as_topo_comparisons/. The data page for comparative analysis of the internet as-level topologies.

[91] http ://www.caida.org/data/active/as relationships/. The caida as relationships dataset.

[92] http ://www.caida.org/tools/measurement/skitter/as_adjacencies.xml. Macroscopic topology as adjacencies.

[93] http ://www.cs.berkeley.edu/sagarwal/research/BGP hierarchy. Sharad agarwal homepage.

[94] http ://www.cymru.com/Bogons/. The team cymru bogon reference page.

[95] http ://www.ece.gatech.edu/research/labs/MANIACS/as_taxonomy/. Autonomous system taxonomy repository.

[96] http ://www.irr.net/docs/list.html. List of routing registries.

[97] http ://www.pch.net/documents/data/routing tables/archive/. Packet clearing house routing tables.

[98] http ://www.research.att.com/ jiawang/as_traceroute. Scalable and accurate as-level traceroute tool.

[99] B. HUFFAKER, D. PLUMMER, D. MOORE, and K CLAFFY. Topology discovery by active probing. 2002.

[100] G. Huston. Interconnection, peering, and settlements. In *Internet Protocol Journal*, March 1999.

[101] Geoff Huston. http ://bgp.potaroo.net/.

[102] Geoff Huston. As number exhaustion and the 4byte transition plan, 11 2005.

[103] Geoff Huston. Where's the money ? - internet interconnection and financial settlements, January 2005.

[104] Y. Hyun, A. Broido, and k claffy. Traceroute and bgp as path incongruities, 2003.

[105] Young Hyun, Andre Broido, and kc claffy. On third-party addresses in traceroute paths. In *Passive and Active Measurement Workshop 2003*, La Jolla, CA, Apr 2003.

[106] Alain Barrat Ignacio Alvarez-Hamelin, Luca Dall'Asta and Alessandro Vespignani. DELIS-TR-0166 - k-core decomposition : a tool for the visualization of large scale networks. techreport 0166, DELIS, 2004.

[107] S. Jaiswal, A. Rosenberg, and D. Towsley. Comparing the structure of power-law graphs and the internet as graph, 2004.

[108] Z. M. Mao D. Johnson J. Rexford J. Wang R. H. Katz. Scalable and accurate identification of as-level forwarding paths. In *INFOCOM*, 2004.

[109] Inas Khalifa. Characterization of the internet at the as level. Technical report, 2002.

[110] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, Number 4598, 13 May 1983*, 220, 4598 :671–680, 1983.

[111] Mark W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36(3) :490–509, 1988.

[112] Ritesh Kumar and Jasleen Kaur. Efficient beacon placement for network tomography. In *IMC '04 : Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 181–186, New York, NY, USA, 2004. ACM Press.

[113] Craig Labovitz, G. R. Malan, and Farnam Jahanian. Origins of internet routing instability. Technical Report CSE-TR-368-98, 17, 1998.

[114] Anukool Lakhina, John Byers, Mark Crovella, and Peng Xie. Sampling biases in IP topology measurements. In *Proceedings of IEEE Infocom*, April 2003.

[115] Lun Li, David Alderson, Walter Willinger, and John Doyle. A first-principles approach to understanding the internet's router-level topology. In *SIGCOMM '04 : Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–14, New York, NY, USA, 2004. ACM Press.

[116] Damien Magoni and Jean Jacques Pansiot. Analysis of the autonomous system network topology. *SIGCOMM Comput. Commun. Rev.*, 31(3) :26–37, 2001.

[117] Daniel C.H. Mah. Explaining internet connectivity : Voluntary interconnection among commercial internet service providers. In *TPRC 2003*, september 2003.

[118] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. A basis for systematic analysis of network topologies, 2006.

[119] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Xenofontas Dimitropoulos, k c claffy, and Amin Vahdat. The internet as-level topology : three data sources and one definitive metric. *SIGCOMM Comput. Commun. Rev.*, 36(1) :17–26, 2006.

[120] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. The internet as-level topology : Three data sources and one definitive metric. *ACM SIGCOMM COMPUTER COMMUNICATION REVIEW*, 36 :2006, 2005.

[121] Priya Mahadevan, Dmitri Krioukov, Marina Fomenkov, Bradley Huffaker, Xenofontas Dimitropoulos, kc claffy, and Amin Vahdat. Lessons from three views of the internet topology, 2005.

[122] Ratul Mahajan. *Practical and Efficient Internet Routing with Competing Interests*. PhD thesis, University of Washington, 2005.

[123] Ratul Mahajan, David Wetherall, and Tom Anderson. Understanding BGP misconfiguration. In *ACM SIGCOMM*, August 2002.

[124] Ali Ridha MAHJOUB. Approches polyédrales, December 2004.

[125] Z. Morley Mao, Lili Qiu, Jia Wang, and Yin Zhang. On as-level path inference. In *SIGMETRICS '05 : Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 339–349, New York, NY, USA, 2005. ACM Press.

[126] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. Towards an accurate as-level traceroute tool. In *SIGCOMM '03 : Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 365–378, New York, NY, USA, 2003. ACM Press.

[127] Quang Nguyen Marc-Olivier Buob, Mickael Meulle. Inférence des accords d'interconnexion de l'internet : Modélisation et résolution. Technical report, France Télécom Division R&D, 2005.

[128] Alberto Medina, Ibrahim Matta, and John Byers. On the origin of power laws in internet topologies. *SIGCOMM Comput. Commun. Rev.*, 30(2) :18–28, 2000.

[129] Jin-Kao Hao Mickael Meulle, Quang Nguyen. Formulation csp et approches heuristiques pour l'inférence des accords d'interconnexion dans l'internet. In *ROADEF'06*, Lille, France, 2006.

[130] Philippe Mahey Mickael Meulle, Jean-luc Lutton. Résolution exacte du problme d'inférence des accords d'interconnexion entre as. In *Algotel'06*, France, 2006.

[131] Quang Nugyen Mickael Meulle, Michel Decima. Découverte de la topologie de l'internet pour l'optimisation des politiques de routage entre opérateurs. Technical report, France Télécom Division R&D, 2003.

[132] M. Mihail, C. Gkantsidis, and E. Zegura. Spectral analysis of internet topologies, 2003.

[133] Wolfgang Muehlbauer, Anja Feldmann, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Building an as-topology model that captures route diversity. In *To appear in ACM SIGCOMM*, 09 2006.

[134] Kengo Nagahashi, Hiroshi Esaki, and Jun Murai. Bgp integrity check for the conflict origin as prefix in the inter-domain routing. In *SAINT*, pages 276–282, 2003.

[135] RIPE NCC. Routing information service raw data, http ://data.ris.ripe.net/.

[136] Ola Nordstrom and Constantinos Dovrolis. Beware of bgp attacks. *SIGCOMM Comput. Commun. Rev.*, 34(2) :1–8, 2004.

[137] W. B. Norton. The evolution of the u.s. internet peering ecosystem. 2004.

[138] W.B. Norton. Internet service providers and peering. In *Proceedings of NANOG 19, Albuquerque (New Mexico)*, June 2000.

[139] W.B. Norton. The art of peering : The peering playbook. Technical report, Equinix.com, 2001.

[140] University of Oregon. Route views project, http ://archive.routeviews.org/.

[141] M. Minoux P. Bonami. Exact max-2sat solution via lift-and-project closure. *Operations Research Letters*, 2005.

[142] M. Minoux P. Bonami. Using rank-1 lift-and-project closures to generate cuts for 0-1 mips, a computational investigation. *Discrete Optimization*, 2 :288–307, 2006.

[143] A. Vahdat P. Mahadevan, D. Krioukov Huffaker X. Dimitropoulos kc claffy. Comparative analysis of the internet as-level topologies extracted from different data sources, June 2004.

[144] S. Park, D. M. Pennock, and C. Lee Giles. Comparing static and dynamic measurements and models of the internet's as topology, 2004.

[145] Seung-Taek Park, David M. Pennock, and C. Lee Giles. Comparing static and dynamic measurements and models of the internet's as topology. In *INFOCOM*, 2004.

[146] Thomas Petermann and Paolo De Los Rios. Exploration of scale-free networks. *The European Physical Journal B*, 38 :201, 2004.

[147] Itamar Pitowsky. Correlation polytopes : their geometry and complexity. *Math. Program.*, 50(3) :395–414, 1991.

[148] Metropolis project. http ://www.laas.fr/ owe/metropolis/metropolis_eng.html.

[149] B. Quoitin and O. Bonaventure. A cooperative approach to interdomain traffic engineering. Submitted to 1st Conference on Next Generation Internet Networks Traffic Engineering (NGI 2005), 2005.

[150] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 2003.

[151] P. Reichl, S. Leinen, and B. Stiller. A practical review of pricing and cost recovery for internet services, 1999.

[152] J. Rekhter. EGP and policy based routing in the new NSFNET backbone. RFC 1092, February 1989.

[153] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006.

[154] Jennifer Rexford. Measuring the autonomous system path through the internet. Intel Research Cambridge Seminar, 2004.

[155] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang. Bgp routing stability of popular destinations. In *Internet Measurement Workshop*, pages 197–202. ACM, 2002.

[156] M. Rimondini, M. Pizzonia, G. Di Battista, and M. Patrignani. Algorithms for the inference of the commercial relationships between autonomous systems : Results analysis and model validation. In *IPS 2004, International Workshop on Inter-domain Performance and Simulation*, pages 33–45, March 2004.

[157] Paolo De Los Rios. Exploration bias of complex networks. In *AIP Conference Proceedings*, volume 661, Issue 1, pages 33–36, April 2003.

[158] Thomas J. Schaefer. The complexity of satisfiability problems. In *STOC '78 : Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, New York, NY, USA, 1978. ACM Press.

[159] Colleen Shannon, David Moore, Ken Keys, Marina Fomenkov, Bradley Huffaker, and k claffy. The internet measurement data catalog. *SIGCOMM Comput. Commun. Rev.*, 35(5) :97–100, 2005.

[160] Yuval Shavitt and Eran Shir. Dimes : Let the internet measure itself, 2005.

[161] R. Siamwalla, R. Sharma, and S. Keshav. Discovering internet topology, 1999.

[162] G. Siganos and M. Faloutsos. Analyzing bgp policies : Methodology and tool, 2004.

[163] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos. Power laws and the as-level internet topology. *IEEE/ACM Trans. Netw.*, 11(4) :514–524, 2003.

[164] N. Spring, R. Mahajan, and T. Anderson. The causes of path inflation. In *SIGCOMM '03 : Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 113–124, New York, NY, USA, 2003. ACM Press.

[165] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, August 2002.

[166] N. Spring, D. Wetherall, and T. Anderson. Scriptroute : A public internet measurement facility, 2002.

[167] N. Spring, D. Wetherall, and T. Anderson. Reverse-engineering the internet, 2003.

[168] Neil Spring. *Efficient discovery of network topology and routing policy in the Internet*. PhD thesis, october 2004.

[169] Neil Spring, Mira Dontcheva, Maya Rodrig, and David Wetherall. How to resolve ip aliases. Technical report, may 2004.

[170] Neil Spring, Larry Peterson, Andy Bavier, and Vivek Pai. Using planetlab for network research : myths, realities, and best practices. *SIGOPS Oper. Syst. Rev.*, 40(1) :17–24, 2006.

[171] Aditya Akella Srinivasan. Toward representative internet measurements. In *NYMAN*, 2003.

[172] J. Stoer and C. Witzgall. *Convexity and Optimization in Finite Dimensions I.* Springer-Verlag, Berlin, 1970.

[173] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *UC Berkeley Technical Report CSD-01-1151, August 2001*, 2001.

[174] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topologies, power laws, and hierarchy, 2001.

[175] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. In *INFOCOM*, pages 736–742, 2001.

[176] Hongsuda Tangmunarunkit, John Doyle, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Does as size determine degree in as topology ?, 2002.

[177] Renata Teixeira, Keith Marzullo, Stefan Savage, and Geoffrey M. Voelker. In search of path diversity in isp networks. In *Proceedings of the conference on Internet measurement conference*, pages 313–318. ACM Press, 2003.

[178] Traceroute.org. http ://www.traceroute.org/.

[179] Y. Vardi. Metrics useful in network tomography studies. *Signal Processing Letters, IEEE*, 11 :353–355, March 2004.

[180] A. Vazquez, R. Pastor-Satorras, and A. Vespignani. Internet topology at the router and autonomous system level, 2002.

[181] Fabien Viger, Alain Barrat, Luca Dall'Asta, Cun-Hui Zhang, and Eric D. Kolaczyk. Network inference from traceroute measurements : Internet topology 'species', 2005.

[182] D. Walton, D. Cook, A. Retana, and J. Scudder. Advertisement of multiple paths in bgp, November 2002.

[183] Feng Wang and Lixin Gao. On inferring and characterizing internet routing policies. In *Proceedings of the conference on Internet measurement conference*, pages 15–26. ACM Press, 2003.

[184] Feng Wang and Lixin Gao. On inferring and characterizing internet routing policies. In *IMC '03 : Proceedings of the 3rd ACM SIGCOMM conference on Internetmeasurement*, pages 15–26, New York, NY, USA, 2003. ACM Press.

[185] Martin B. Weiss and Seung Jae Shin. Internet interconnection economic model and its analysis : Peering and settlement. *Netnomics*, 6(1) :43–57, 04 2004. available at http ://ideas.repec.org/a/kap/netnom/v6y2004i1p43-57.html.

[186] Rene Wilhelm. Ttm as-level traceroutes : Matching ips to ases, september 2003.

[187] www.cambridge.intel research.net/monitoring/dante. Intel-dante monitoring project.

[188] www.iana.org/assignments/as numbers. Asn allocations (iana).

[189] Dimitropoulos X., Krioukov D., Riley G., and klaffy k. Revealing the autonomous system taxonomy : The machine learning approach. In *Passive and Active Measurement Conference Passive and Active Measurement Conference Passive and Active Measurement Conference PAM, Passive and Active Measurement Conference*, 2006.

[190] Jianhong Xia and Lixin Gao. On the evaluation of as relationship inferences. In *IEEE Global Communications Conference (GLOBECOM), Dallas, TX*, November 2004.

[191] Jianhong Xia and Lixin Gao. On the evaluation of as relationship inferences (technical report). Technical report, November 2004.

[192] Kuai Xu, Zhenhai Duan, Zhi-Li Zhang, and Jaideep Chandrashekar. On properties of internet exchange points and their impact on as topology and relationship. In *NETWORKING*, pages 284–295, 2004.

[193] Bin Yao, Ramesh Viswanathan, Fangzhe Chang, and Dan G. Waddington. Topology inference in the presence of anonymous routers. In *INFOCOM*, 2003.

[194] Eric W. Yocam. Autonomous system architecture tutorial : Protocols, design and implementation, August 2002.

[195] Ellen W. Zegura, Milena Mihail, Christos Gkantsidis, and Amin Saberi. On the semantics of internet topologies. Technical report, Georgia Institute of Technology, 2002.

[196] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1) :53–61, 2005.

[197] Hongwei Zhang, Anish Arora, and Zhijun Liu. A stability-oriented approach to improving bgp convergence. *srds*, 00 :90–99, 2004.

[198] Xiaoliang Zhao, Dan Pei, Lan Wang, Dan Massey, Allison Mankin, S. Felix Wu, and Lixia Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *Proceedings of the First ACM SIGCOMM Workshop on Internet Measurement*, pages 31–35. ACM Press, 2001.

[199] Shi Zhou. Understanding the internet topology evolution dynamics, 2005.

[200] Shi Zhou and Raul J. Mondragon. The missing links in the bgp-based as connectivity maps, 2002.

[201] Shi Zhou and Raul J. Mondragon. The rich-club phenomenon in the internet topology, 2003.

[202] Shi Zhou and Raul J. Mondragon. Towards modelling the internet topology - the interactive growth model, 2003.

[203] Shi Zhou and Raul J. Mondragon. Accurately modeling the internet topology, 2004.

[204] Shi Zhou, Guoqiang Zhang, and Guoqing Zhang. The chinese internet as-level topology, 2005.

[205] Gunter M. Ziegler. Lectures on polytopes : updates, corrections, and more (updated version), July 2000.

Annexe A

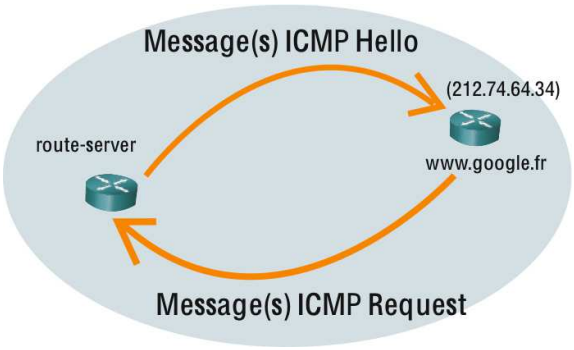
Outils de tomographie IP

Cette annexe présente plusieurs variantes des deux principaux outils de tomographie IP : ping et traceroute. Il existe d'autres outils qui exploitent des fonctionnalités similaires des protocoles IP, ICMP, TCP et UDP. Mais nous ne les détaillons pas ici car ils n'apportent pas d'informations supplémentaires directes concernant la topologie ou le routage IP.

A.1 L'outil ping (ICMP)

L'outil ping permet de savoir si une adresse IP correspond à une machine du réseau. Par défaut, une machine répond au requêtes ICMP hello avec une de ses adresses IP. Notons que sous environnement coopératif, les réponses sont fréquentes mais pas toujours assurées. Des routeurs désactivent les réponses aux requêtes ICMP lorsque la configuration de sécurité de la machine est spécifique, ou lorsque le réseau comporte des routeurs filtrants (switches[1] évolués, pare-feux[2] matériels et logiciels...).
La figure A.1 illustre l'opération effectuée par ping : une requête ICMP (couche 3) de type Hello est envoyée en destination d'une adresse donnée @. Le routeur correspondant à l'adresse @ répond par un message protocolaire ICMP de type Reply. Le routeur originaire du ping récupère le message ICMP où une adresse IP différente peut avoir été indiquée par la machine d'adresse @.

[1]Les switches sont des équipements de routage qui fonctionnement à la couche 2 du modèle OSI.
[2]Les pare-feux sont des équipements de routage du réseau capables de filtrer des paquets IP en fonction de nombreux paramètres.



(a) Illustration d'une commande ping du routeur "route-server" vers l'adresse 212.74.64.34

```
route-server>ping www.google.fr
Translating "www.google.fr"...domain server (212.74.64.34) [OK]

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 216.239.59.147, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 20/26/36 ms
```

(b) Trace de la commande ping exécutée du routeur "route-server" vers l'adresse 212.74.64.34

FIG. A.1 – Ping ICMP

A.2 L'outil "ping TCP"

L'outil ping TCP permet de savoir si une adresse IP correspond à une machine du réseau en lui envoyant une requête de type TCP avec le flag[3] ACK activé. Par défaut, une machine répond un paquet TCP si le port contacté correspond à un service réseau installé (port 80 pour les serveurs web, 21 pour FTP[4]...). Sous environnement coopératif, les réponses des machines sont fréquentes. Des routeurs filtrants (switches évolués, firewall matériels et logiciels...) peuvent toutefois filtrer ce genre de requête si les flux TCP sont analysés dans le dispositif de sécurité.

La figure A.1 illustre l'opération effectuée par ping TCP : une requête TCP (couche 4) avec le flag ACK[5] est envoyée en destination d'une adresse donnée @ et d'un port (les ports usuels 80 pour HTTP[6], 21 pour FTP, 22 pour SSH[7], 23 pour telnet donnent de bons résultats). Le routeur correspondant à l'adresse @ répond par message protocolaire TCP

[3]Les flags sont des indicateurs dans les en-têtes des paquets de données (un en-tête par protocole). Ils sont en rapport avec le fonctionnement du protocole concerné.

[4]File Transfer Protocol

[5]Le flag ACK sert à acquitter des paquets TCP comme s'il avaient été préalablement reçus.

[6]Hypertext Transfer Protocol

[7]Secure Shell

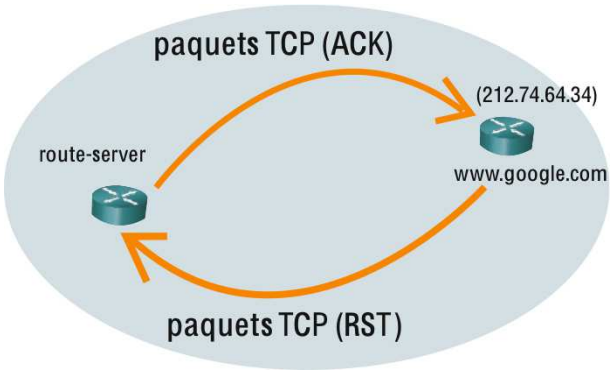


FIG. A.2 – Exemple de ping TCP

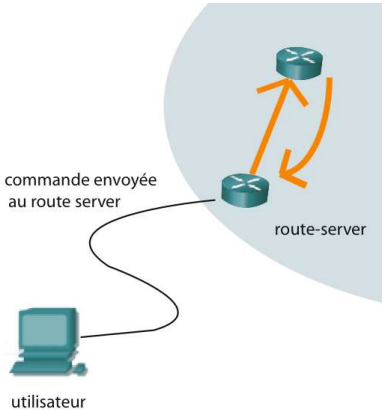
avec le flag RST[8].

A.3 L'outil traceroute (ICMP, UDP, TCP)

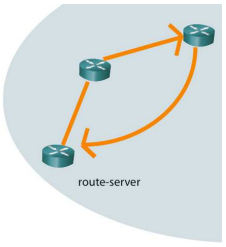
L'outil traceroute permet la récupération d'un chemin de signalisation formé d'adresses IP via l'envoi de requêtes ICMP, TCP ou UDP dont la durée de vie (la durée de vie est le champ TTL[9] des paquets IP) est incrémentée de 1 en 1 à partir de 1 jusqu'à joindre l'adresse IP destination demandée. Une machine exécutant un traceroute va faire un nombre de requêtes fixe par nombre de bonds. Nous prenons 1 requête par bond dans notre explication. La machine commence par envoyer une requête ICMP hello (ou UDP sur un port injoignable ou TCP ACK), avec une durée de vie de 1 et à destination d'une adresse @. La machine recevant le message, qui est la prochaine machine en direction de l'adresse @, décrémente la durée de vie du paquet à sa réception. Le paquet ne peut alors atteindre l'hôte destination car son TTL est nul. Dans la plupart des cas, un paquet ICMP est renvoyé à la source indiquant la non distribution du ou des paquets envoyés (type "Destination Unreachable"). Une fois la réponse reçue de la machine au premier bond, la machine source continue la trace en envoyant une nouvelle requête vers @ mais avec une durée de vie de 2. Le paquet retour sera donc envoyé par la deuxième machine. Et ainsi de suite jusqu'à la dernière machine d'adresse @. La machine d'adresse @ ne renvoie pas cette fois une requête "Destination Unreachable" puisqu'elle a reçu le paquet. En fonction du type de message envoyé par la source (ICMP "Hello" ou TCP ou UDP), la machine d'adresse @ décidera de répondre on non un message de type ICMP "Echo

[8]remise à zéro de la session TCP
[9]Time To Leave.

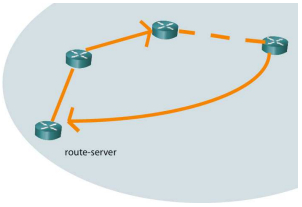
Reply" (respectivement TCP (RST) et ICMP "Port Unreachable"). Si les routages n'ont pas changé pendant le sondage et si toutes les machines ont répondu aux requêtes envoyées, on obtient alors un chemin d'adresses IP. L'adresse IP indiquée par chaque routeur est en fait soit l'adresse de l'interface d'arrivée, soit l'adresse de l'interface de sortie, soit une adresse fixe, soit l'adresse d'un firewall (cas de routeurs NAT invisibles).
On ajoutera enfin que la commande traceroute ne permet de connaître que le chemin de la source vers la destination et pas le chemin retour.



(a) Vers le permier bond (IP TTL=1)



(b) Vers le deuxième bond (IP TTL=2)



(c) Vers le k-ième bond (IP TTL=k)

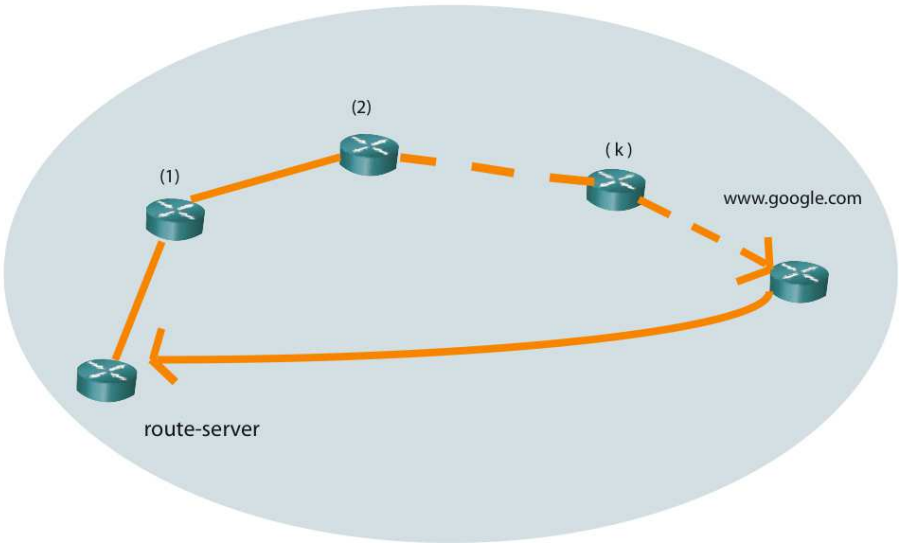
```
doodoo.opentransit.net>traceroute www.google.com
Translating "www.google.com"...domain server (193.251.246.3) (193.251.151.57)

Translating "www.google.com"...domain server (193.251.246.3) [OK]

Type escape sequence to abort.
Tracing the route to www.l.google.com (66.249.85.104)

 1 grumpy.opentransit.net (204.59.3.37) 0 msec 0 msec 0 msec
 2 po6-0.ashcr2.Ashburn.opentransit.net (193.251.240.237) 0 msec 0 msec 0 msec
 3 gi0-0-0.ashcr1.Ashburn.opentransit.net (193.251.241.41) 0 msec 0 msec 0 msec
 4 po6-0.loncr3.London.opentransit.net (193.251.240.181) 72 msec 72 msec 72 msec
 5 so-1-1-0-0.loncr4.London.opentransit.net (193.251.242.218) 72 msec 80 msec 76 msec
 6 so-0-0-0-0.fftcr4.Frankfurt.opentransit.net (193.251.154.90) 88 msec 88 msec
   so-0-1-0-0.fftcr4.Frankfurt.opentransit.net (193.251.242.137) 92 msec
 7 te10-1.fftse1.FrankfurtAmMain.opentransit.net (193.251.241.202) 88 msec 88 msec 88 msec
 8 google.GW.opentransit.net (193.251.249.62) 164 msec 92 msec 88 msec
 9 www.l.google.com (66.249.85.104) 92 msec 88 msec 92 msec
doodoo.opentransit.net>
```

(d) Commande exécutée en direction de www.google.fr (66.249.85.104)



(e) Vers la destination (dernier bond)

FIG. A.3 – Traceroute bond par bond

Annexe B

Filtrage des tomographies BGP

Cette annexe présente en détail les différents filtres appliqués sur les tomographies.

B.1 Les tomographies utilisées

Nous réalisons nos expériences en partant de Tomographies passives : des tables de routage BGP $P_{j,s}$ capturées un jour j sur une sonde s. On collecte des tables de routage jour par jour pendant 30 jours. Les tables de routage sont en provenance des projets Route-Views [140], RIS [135]. Nous avons également ajouté les tables en provenance des réseaux Cernet [84], GEANT [187] et Packet clearing House [97], et en provenance de routeurs looking glass [178]. Les routeurs sondes disponibles sont reportés tableau B.1. Les différentes périodes de temps étudiées sont reportées dans le tableau B.2.

B.2 Gestion des dates

B.2.1 Observation d'une donnée dans le temps

Soit un observable quelconque (une route, un préfixe, l'annonce d'un préfixe par un AS...) qui peut se calculer pour une tomographie élémentaire ou pour une union de tomographies élémentaires du même jour. En considérant une tomographie P durant plusieurs jours, on peut étudier les apparitions dans le temps de l'information recherchée. On définit les quantités suivantes :

Définition 41. *Notations sur les densités*

T temps discret de la tomographie (en nombre de dates). Exemple : trente jours.

Sondes	sondes BGP	next hop BGP	AS sources
Route-Views	Univ of Oregon, Eugene Oregon, USA	39	33
Route-Views	ISC (PAIX), Palo Alto CA, USA	80	12
Route-Views	LINX, London, GB	270	10
Route-Views	Equinix, Ashburn, VA	4	4
Route-Views	DIXIE (NSPIXP), Tokyo, Japan	60	4
Route-Views	Toutes	453	50
R.I.S.	RRC00, RIPE NCC, Amsterdam	11	11
R.I.S.	RRC01, LINX, London	135	43
R.I.S.	RRC02, SFINX, Paris	20	19
R.I.S.	RRC03, AMS-IX, NL-IX, GN-IX Amsterdam	234	78
R.I.S.	RRC04, CIXP, Geneva	23	9
R.I.S.	RRC05, VIX, Vienna	66	46
R.I.S.	RRC06, NSPIXP2, Otemachi, Japan	29	6
R.I.S.	RRC07, Netnod, Stockholm, Sweden	31	13
R.I.S.	RRC10, MIX, Milan, Italy	59	10
R.I.S.	RRC11, NYIIX, New York (NY), USA	14	14
R.I.S.	RRC12, DE-CIX, Frankfurt, Germany	150	24
R.I.S.	RRC14, PAIX, Palo Alto, USA	2	1
R.I.S.	Toutes	772	211
Intel-Dante	geneva, CH	48	1
Toutes	Toutes	1116	241

TAB. B.1 – Sondes BGP utilisées

Tomographie	date début	date fin	Routes	Préfixes	Route canoniques
$T_{2005-01}$	01/01/2005	31/01/2005	28 267 547	192 339	2 803 869
$T_{2006-08}$	01/08/2006	31/08/2006	?	?	?

TAB. B.2 – Tomographies utilisées

T_e *date de première observation de l'information. Exemple : cinquième jour.*

T_s *date de dernière observation de l'information. Exemple : vingt-neuvième jour.*

$d = T_s - T_e + 1$ *durée d'observation ou longévité de l'information. Exemple : vingt-cinq jours.*

N *nombre de périodes d'observations consécutives de l'information. Exemple : trois périodes d'observation avec stabilité.*

$J_i, i \in 1 \dots N$ *i-ème période consécutive d'observation de l'information (les ensembles J_i sont disjoints). Exemple : du quatorzième jour jusqu'au vingtième jour.*

$j_i = |J_i|, i \in 1 \dots N$ *durée en nombre de jours de la i-ème période consécutive. Exemple : 6 jours.*

$J = \bigcup_{i=1}^{N} J_i$ *dates d'observations de l'information. Exemple : les jours 5,...,12,14,..,20,24,..,29.*

$j = \sum_{i=1}^{N} j_i$ *nombre de jours d'observations de l'information. Exemple : 21 jours.*

$D_a = J/T$ *densité absolue de l'information. Exemple : $\frac{21}{30} = 0.7$ (70 %).*

$D_r = d/J$ *densité relative de l'information. Exemple : $\frac{21}{25} = 0.84$(84%).*

$D = d/T = \frac{D_r}{D_a}$ *densité, ou fidélité d'existence de l'information. Exemple : $\frac{25}{30} = 0.83$(83%)*

Les observables n'ont pas toujours un sens suivant ce qui est observé. Grâce à ce formalisme, on peut étudier par exemple chaque route d'un next hop d'une sonde pour chaque préfixe. Pour chaque couple $(s, nexthop, p, path, repet)$ (sonde,next hop, chemin canonique, motif de répétitions), on peut connaître les dates correspondantes d'observation. Ces dates permettent de connaître les périodes J_i correspondantes.
Exemple pour l'observation de la route 7575 1239 5511 2200 vers le préfixe 193.54.79/24, pour le next hop BGP 198.32.176.177 issu de l'AS 7575 connecté à la sonde $route - views - palo - alto(usa)$. Voir Figure B.1.

B.2.2 Observation des valeurs d'un observable dans le temps : multiplicité

Soit un observable donné, on peut définir les notions suivantes :

Définition 42. *multiplicité de valeurs d'un observable et densités comparées*

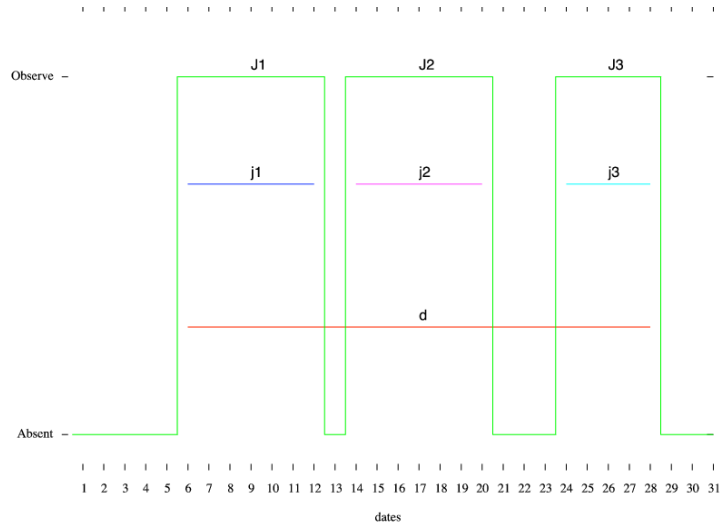


FIG. B.1 – Observation de la présence d'une route sur 31 dates.

Soit un observable α d'une tomographie P. Soient $(\alpha_1), \ldots, (\alpha_n)$. les valeurs de α observées pour P. Chaque valeur α_i est aussi un observable. Conformément à la définition 41, on peut associer à chaque α_i, des intervalles d'observation consécutifs $J_1(i), \ldots, J_{N_i}(i)$. On peut alors re-écrire la densité absolue et la densité relative de l'observable α en fonction des différentes valeurs α_i :

$$D_r(\alpha) = \frac{|\bigcup_{i=1}^{n}\bigcup_{j=1}^{N_i} J_j(i)|}{Max_{1\leq i\leq n} d_s(i) - Min_{1\leq i\leq n} d_e(i)} \text{ et } D_a(\alpha) = \frac{|\bigcup_{i=1}^{n}\bigcup_{j=1}^{n_i} J_j(i)|}{T}$$

Afin de comparer les différentes valeurs α_i, on définit les grandeurs suivantes :

Multiplicité des valeurs de l'observable α : $M(\alpha) = n$

Densité relative de la valeur α_{i_0} par rapport à α : $D_r(\alpha_{i_0}/\alpha) = \dfrac{j(i_0)}{|\bigcup_{i=1}^{n} J(i)|}$

Densité de conflit d'une valeur α_{i_0} : $C(\alpha_{i_0}/\alpha) = \dfrac{|\bigcup_{i=1}^{n}(J(i_0) \bigcap J(i))|}{j(i_0)}$

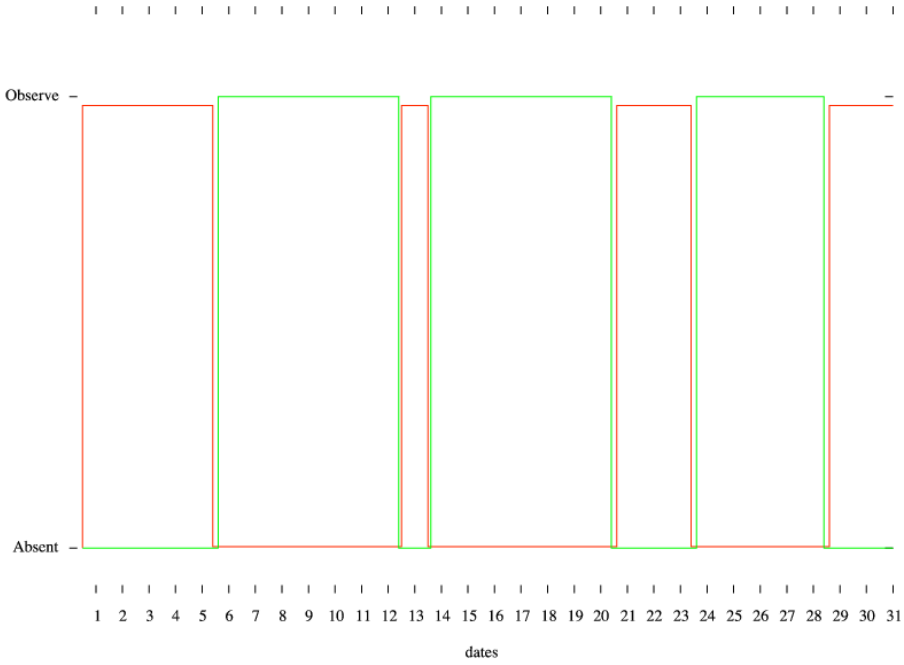
Densité comparée d'une valeur α_{i_0} :

$$D_c(\alpha_{i_0}) = \frac{|\bigcup_{i=1}^{n}(J(i_0) \bigcap J(i))|}{|\bigcup_{i=1}^{n} J(i)|} = C(\alpha_{i_0}/\alpha) \times D_r(\alpha_{i_0}/\alpha)$$

Densité comparée des différentes valeurs de α :

$$D_c(\alpha) = \frac{\bigcup_{i=1}^{n} \bigcup_{j=1}^{j<i} (J(i) \bigcap J(j))}{\bigcup_{i=1}^{n} J(i)}$$

Coefficient de persistance des valeurs de α : $U(\alpha) = \dfrac{d(\alpha)}{\sum_i d(\alpha_i)}$



(a) Observation des deux routes du next hop BGP 198.32.176.177 de la sonde $route - views - palo - alto(usa)$ pour le NLRI 193.54.79/24 pour 31 dates. Dans cet exemple, l'observable α est la route d'un collecteur vers un préfixe. Les valeurs de l'observable sont les deux routes observées : $M(\alpha) = 2$. Pour la route 1 : $N(1) = 2$, $d_1 = 30$, $D_a(1) = \frac{22+4}{31} = D_r(1/\alpha)$, $D_r(1) = \frac{22+4}{30}$. Pour la route 2 : $N(1) = 2$, $d_1 = 8$, $D_a(1) = \frac{1+4}{31} = D_r(2/\alpha)$, $D_r(1) = \frac{1+4}{8}$. On a $U = \frac{8+30}{31}$. Ici, il n'y jamais deux routes différentes pour le même collecteur vers le même préfixe. Donc $D_c(1/\alpha) = D_c(2/\alpha) = D_c(\alpha) = 0$.

FIG. B.2 – Observations de deux routes BGP dans le temps

Exemples d'application : attribution de l'AS origine d'un préfixe (voir B.3.2).

Un autre exemple consiste à observer les différents chemins d'un routeur collecteur vers chaque préfixe.

Exemple : Soient, pour le préfixe 193.54.79/24, pour la sonde $route - views - palo - alto(usa)$ et pour le next hop BGP 198.32.176.177 issu de l'AS 7575, deux routes : 7575 1239 5511 2200 (route 1) et 7575 2914 5511 2200 (route 2). Voir Figure B.1.

B.3 Filtrage des tomographie BGP

On décrit une série de filtres destinés à extraire les cheminements stables et à filtrer les erreurs possibles dans les routes BGP observées. Dans l'annexe B, on utilise une première tomographie $T_{2005-01}$ pour présenter une étude complète de l'effet des filtres. Pour la seconde tomographie $T_{2006-08}$, la volumétrie des données est telle qu'il n'a pas été possible de stocker toutes les données en même temps. Les méthodes de compression et d'indexation des données permettent d'obtenir de bonnes performances, mais l'espace mémoire disponible sur un ordinateur (plus de $3Go$) ne permet pas le chargement de toutes les routes avec les collecteurs et les dates d'observation correspondants. Par contre il est possible en deux phases de lecture de pouvoir extraire une tomographie stable. Une première phase de lecture des données BGP ne conserve que les informations nécessaires pour initialiser les filtres. La deuxième phase de lecture des donnée construit la tomographie finale en ne retenant que les routes non filtrées.

B.3.1 Filtrage élémentaire

Les filtres élémentaires décident si des préfixes doivent être conserver en fonction de leur nature ou en fonction de leurs apparitions dans le temps. Pour la première phase de lecture des données et de création du filtre, on peut conserver uniquement une association entre les préfixes et les dates d'apparition correspondantes.

a) Filtrage des destinations invalides

Les plages d'adresses IP publiques du réseau Internet ne couvrent pas l'ensemble du spectre d'adresses. En dehors des adresses publiques, des plages sont réservées pour d'autres usages (réseaux privés locaux, plages multicast, plages réservées...). Dans une tomographie, il est pertinent de rejeter les routes à destination de préfixes non publiques [94]. Voir tableau B.3.

Définition 43. *Filtre préfixes-publiques (pp)*
On appelle filtre préfixe-publique ou (pp), le filtre qui supprime les routes à destination de préfixes non publiques d'une tomographie.

Tomographie	préfixes filtrés
$T_{2005-01}$	1 670
$T_{2006-08}$	24 482

TAB. B.3 – Filtrage (pp) appliqué aux tomographies

b) Filtrage des AS privés

Il existe des cas où les routes BGP simplifiées d'une tomographie ont des routes canoniques contenant des AS privés. Dans une tomographie BGP, pour construire un graphe des AS par exemple, il est pertinent de filtrer les routes contenant des AS privés[1].

Définition 44. *Filtre AS-publique (ap)*
On appelle filtre AS-publique ou (ap), le filtre qui supprime les routes d'une tomographie contenant des AS non publiques.

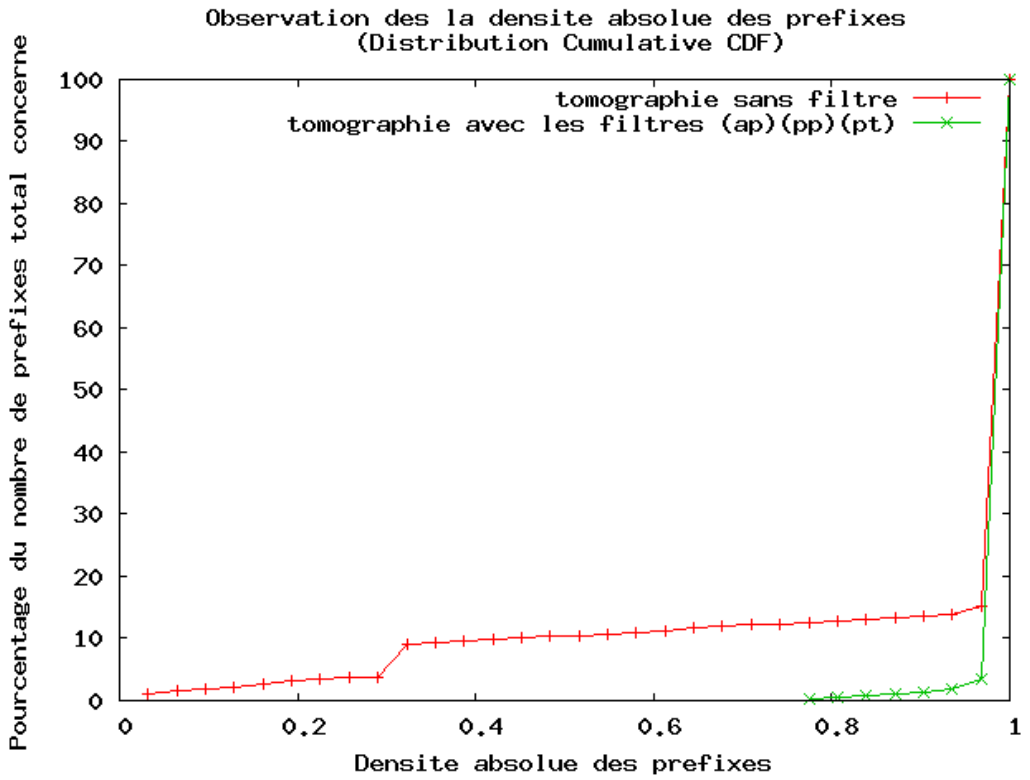
c) Filtrage des préfixes rares

Si certains préfixes n'apparaissent que très peu de temps par rapport au temps d'observation T de la tomographie (toutes sondes confondues), on peut considérer que ces derniers ne sont pas à prendre en compte. Ces préfixes observés peuvent correspondre à des manipulations ponctuelles, à des erreurs de configuration, à des disparitions ou des apparitions de préfixes en cours de tomographie. Voir tableau B.4.

Définition 45. *Filtre préfixe-temps (pt)*
On appelle filtre préfixe-temps ou (pt) le filtre qui supprime les routes tomographie à destination de préfixes dont la densité absolue est inférieur à une valeur de seuil (la densité absolue est calculée toutes sondes confondues). La valeur de coupure sera déterminée expérimentalement en fonction de chaque tomographie.

Les préfixes sont en théorie observés toutes les dates à cause de la contrainte de connectivité totale de chaque réseau. Nous avons opté pour la valeur 0.75 permettant de prendre en compte les cas où il manque des données pour certains jours.

[1] on pourrait imaginer toutefois d'agréger les routes contenant des AS privés pour conserver la partie publique de la route.



(a) Apparitions temporelles des différents préfixes avant et après application du filtre préfixe-temps (pt) et des filtres (pp)(ap) sur la tomographie $T_{2005-01}$

FIG. B.3 – Filtrage préfixe-temps

Tomographie	valeur coupure	préfixes filtrés
$T_{2005-01}$	0.75	23 935
$T_{2006-08}$	0.75	42 294

TAB. B.4 – Filtrage préfixe-temps

B.3.2 Attribution du ou des AS origines pour un préfixe

Pour une tomographie BGP P, il est fréquent d'observer des préfixes annoncés par plusieurs AS origines. Lorsque pour une tomographie P, un préfixe est observé avec au moins deux AS origines différents, on dit que le préfixe est *litigieux*. Les différents cas suivants peuvent se produire :

- Changement d'AS origine (inclu la fusion et la cession d'AS). Pendant la période de transfert, les deux AS peuvent annoncer le même préfixe pour assurer la connectivité.
- Un réseau qui ne définit pas d'AS publique, mais qui dispose de plages publiques, va s'interconnecter avec des AS (routage statique, OSPF, BGP avec AS privé...) qui annonceront ses plages. Les plages peuvent être constamment annoncées par plusieurs AS ou alors le réseau à qui appartient le préfixe utilise un seul fournisseur pour annoncer ses plages, et d'autres fournisseurs en cas de pannes.
- Les fournisseurs d'un AS peuvent quelquefois annoncer certains des préfixes de l'AS pour des raisons pratiques.
- Des utilisations spécifiques des annonces de préfixes par plusieurs AS existent (exemple : les préfixes de points d'échanges, le service "Connexion by boeing").
- Les erreurs des administrateurs.

On se propose de détecter les changements d'AS origines, les erreurs et les préfixes de clients multihomés[2].

Remarque : pour la première phase de lecture des données et de création du filtre, on peut conserver uniquement une association entre les couples (AS origine, préfixe) et les dates d'apparition correspondantes.

Définition 46. *préfixe MOAS*
On appelle préfixe MOAS d'une tomographie P, un préfixe pour lequel il existe au moins deux routes canoniques ne terminant pas par le même AS, pour au moins une même date. Pour tout préfixe p , s'il existe une date D telle que : $|\bigcup_{R=(,p,,,d)\in P} ORIG(R)| > 1$*, alors p est un préfixe MOAS.*

Pour une tomographie P, on peut associer un unique AS origine $ORIG(P, p)$ à un préfixe p non MOAS et plusieurs AS origines pour un préfixe MOAS.
Exemple : le préfixe p est MOAS si on trouve les deux routes canoniques suivantes avec le préfixe p pour une même date : 1 500 88 et 1 99 70 222

[2]Un client multi-homé a par définition plusieurs fournisseurs.

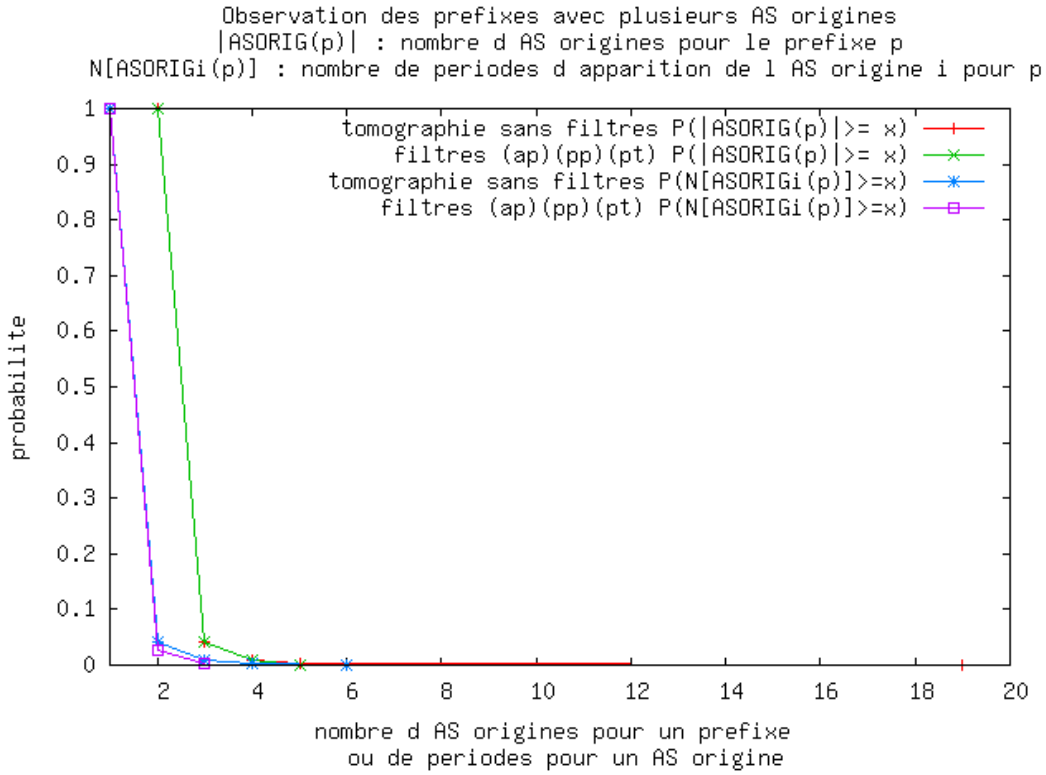


FIG. B.4 – Répartition des valeurs $M(ASORIG(p))$ et $N(ASORIG(p))$ avant et après filtrage (pp), (ap) et $(pt = 0.75)$

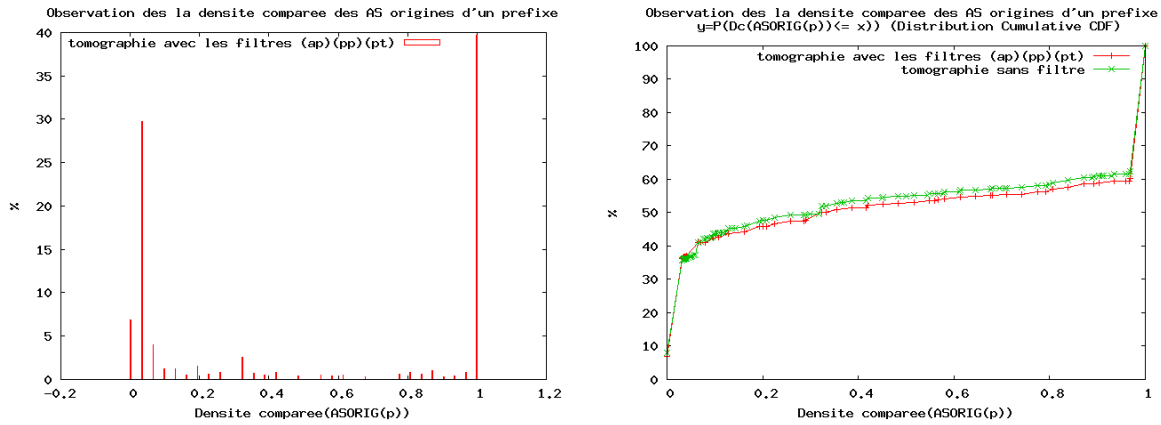
Nous proposons des indications pour détecter les différents cas de préfixes MOAS. Nous définissons au préalable l'observable qui va nous permettre de différencier les cas de figure :

$ASORIG(p) = \{X \in AS / \exists R \in P$ avec $ORIG(R) = X\}$ (prend pour valeurs les différents AS origines de p)

Tout d'abord, on peut se reporter à la figure B.4 qui montre pour les préfixes litigieux la répartition du nombre d'AS origines $M(ASORIG(p))$ et la répartition du nombre de périodes $N(ASORIG(p))$. Plus de 95% des préfixes litigieux apparaissent seulement avec 2 AS origines différents $(M = 2)$ et plus de 95% des préfixes litigieux apparaissent de façon stable dans le temps $(N = 1)$, soit un minimum de 90% des préfixes litigieux apparaissent avec deux AS origines différents et pour une période de temps stable.

Remarque : les filtres basiques $(pp)(ap)(pt)$ suppriment des préfixes qui peuvent avoir jusqu'à 20 AS origines et augmentent le nombre de préfixes qui apparaissent sans interruption.

Pour faire une classification des préfixes MOAS, nous proposons des guides d'analyses suivant :



(a) $D_c(ASORIG(p))$ pour la tomographie $T_{2005-01}$

FIG. B.5 – Indicateur $D_c(ASORIG(p))$ pour les préfixes MOAS

(A) $D_c(ASORIG(p)) \ll 1$ **:** les conflits d'AS origine pour p sont rares par rapport aux apparitions de p.

(A') $D_c(ASORIG(p)(i)) \ll 1$ **:** le i-ème AS origine détecté pour p entre peu souvent en conflit avec les autres AS origines de p comparé au nombre de conflits.

(B) $D_r(\, ASORIG(p)(i) \,/\, ASORIG(p)\,) \ll 1$ **:** le i-ème AS origine détecté pour p n'apparaît pas souvent par rapport à la durée d'observation du préfixe p.

(C') $ASORIG(p)(i_1) = ArgMax_i\{D_r(\, ASORIG(p)(i) \,/\, ASORIG(p)\,)\}$ **:** l'AS origine le plus observé dans le temps pour p (par rapport à la durée d'observation du préfixe p)

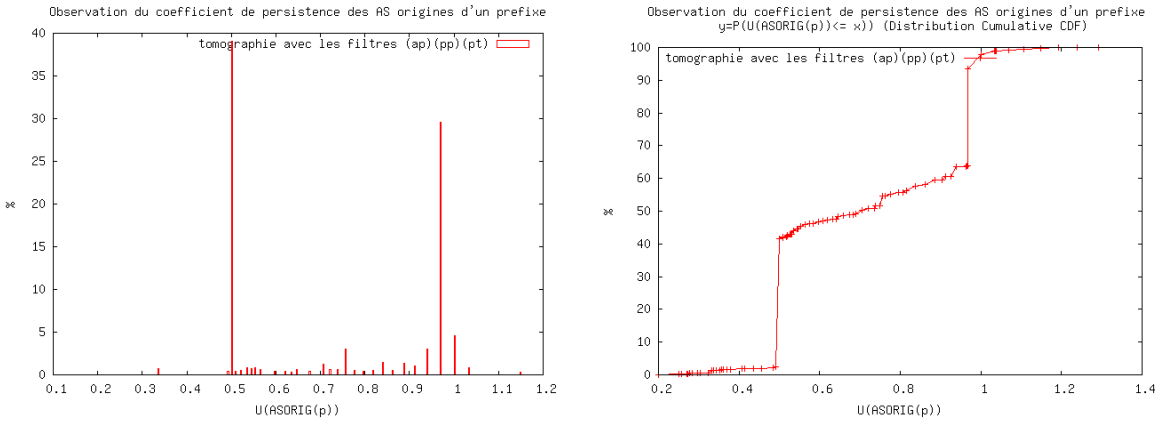
(C") $\beta(ASORIG(p)) = \frac{D_r(\, ASORIG(p)(i_2) \,/\, ASORIG(p)\,)}{D_r(\, ASORIG(p)(i_1) \,/\, ASORIG(p)\,)} \ll 1$ **:** l'AS origine le plus observé dans le temps $ASORIG(p)(i_1)$ est dominant d'un point de vue temporel pour le préfixe p par rapport à l'AS origine $ASORIG(p)(i_2)$.

(D) $C(ASORIG(p)(i)) \ll 1$ **:** le i-ème AS origine détecté pour p n'apparaît pas souvent en conflit par rapport à ses apparitions.

(E) $U(ASORIG(p)) \simeq 1$ **:** les différents AS origines apparaissent quasiment en séquence dans le temps.

Les métriques (A-E) sont discriminantes pour connaître le ou les AS origines d'un préfixe donné. On ne montrera ici que des cas simples et standards[3]. Soit un préfixe p litigieux, alors :

[3]Une corrélation plus précise des différentes valeurs et un arbre de décision pourraient être définis pour filtrer toutes les routes invalides des préfixes litigieux.



(a) $U(ASORIG(p))$ pour la tomographie $T_{2005-01}$

FIG. B.6 – Indicateur $U(ASORIG(p))$ pour les préfixes MOAS

Si $D_c(ASORIG(p)) = 0$: aucun conflit à la même date n'est observé entre deux routes correspondant à deux AS origines différents. Le préfixe litigieux n'est pas un MOAS au sens stricte du terme. Si le préfixe est cédé d'un AS à un autre, alors (E) doit être vrai. Si (E) n'est pas vrai, les différents AS origines ont annoncé p à tour de rôle et sans conflits pendant la période de la tomographie.

Sinon : p est un préfixe MOAS d'après la tomographie.

Supposons désormais que p est un préfixe litigieux et MOAS, alors :

Si un p est cédé d'un AS pour un autre AS : les AS origines sont supposés être deux et il y a peu de chevauchement entre les observations des différents AS origines. Autrement dit, pour une observation assez longue (une tomographie passive d'une ou deux dizaines de jours), les propriétés (A) et (E) sont vérifiées.

Si p appartient à un client multihomé avec plusieurs fournisseurs "actifs" : $D_c(ASORIG(p)$ est proche de 1 ((A) n'est pas vérifiée) et $U(ASORIG(p)) \simeq \frac{1}{M(ASORIG(p))}$ (les AS origines sont observés sur quasiment toute la période) . Le cas d'un client parfaitement multihomé ou d'un partage des AS origines se produit lorsque $D_c(ASORIG(p)) \simeq 1$. C'est-à dire que le préfixe est toujours annoncé par plusieurs AS origines quelque soit la date d'observation.

Si p est un préfixe d'un point d'interconnexion annoncé par plusieurs ISP : même conclusions que l'item précédant.

Les erreurs des administrateurs et les annonces ponctuelles dans le temps de certains préfixes par un autre AS que l'AS présumé AS origine peuvent être détectées si on suppose la période d'observation assez longue par rapport au temps de correction des erreurs et

Tomographie	valeur coupure	préfixes filtrés
$T_{2005-01}(pp, ap, pt = 0.75)$	0.15	404
$T_{2006-08}(pp, ap, pt = 0.75)$	0.15	673

TAB. B.5 – Filtrage préfixe-erreurs (pe)

aux temps des tests ponctuels d'annonce. Sélectionnons, pour chaque préfixe MOAS, l'AS origine i_1 maximisant $D_r(\, ASORIG(p)(i) \,/\, ASORIG(p)\,)$ (la durée d'observation du préfixe avec l'AS origine i divisé par la durée d'observation du préfixe p, tout AS origines confondus, voir (C) et (C')). L'AS origine i_1 est appelé l'AS origine ***dominant*** d'un préfixe. Pour filtrer les erreurs, on va supprimer les routes qui indiquent un autre AS origine que l'AS origine dominant pendant une période négligeable par rapport à la période d'observation de l'AS origine dominant.

Définition 47. *Filtrage préfixe-erreurs (pe)*
On dira appliquer un filtre préfixe-erreur ou (pe) sur une tomographie P, si on retire les routes à destination des préfixes dont le rapport $\frac{D_r(\, ASORIG(p)(i) \,/\, ASORIG(p)\,)}{D_r(\, ASORIG(p)(i_1) \,/\, ASORIG(p)\,)}$ est inférieur à une valeur de seuil.

La valeur du seuil pour le filtre (pe) correspond à un rapport $\frac{1}{k}$. Le filtre (pe) supprime les routes qui indiquent un AS origine non dominant apparaissant au moins k fois moins souvent que l'AS origine dominant. Le filtre (pe) va aussi supprimer les routes à destination de préfixes dont la cession s'est effectuée en début ou en fin de tomographie. Pour les autres préfixes cédés pendant la période de la tomographie, on supprimera les routes grâce au filtre (pc).

Remarque : le nombre de routes filtrées dans les tableaux correspond au nombre de routes qui disparaissent de la tomographie après filtrage. Cela n'implique pas nécessairement que des routes canoniques disparaissent.

Les préfixes qui sont annoncés de façon stable par deux AS sont des cas à traiter à part. On conservera les routes qui indiquent toute sondes confondues, deux AS origines différents.

Définition 48. *Filtrage préfixe-multihomés (pm)*
On dira appliquer un filtre préfixe-multihomés ou (pm) sur une tomographie P, si on garde les routes à destination des préfixes qui ont deux AS origines seulement après filtre (pe) et dont le rapport $D_c(ASORIG(p))$ est proche de 1.

Après traitement des erreurs, des annonces ponctuelles et des préfixes multihomés, on s'attend à pouvoir trouver un AS origine dominant pour chaque préfixe litigieux restant. En

Tomographie	valeur coupure	préfixes conservés
$T_{2005-01}(pp, ap, pt = 0.75, pe = 0.15)$	0.75	1 026
$T_{2006-08}(pp, ap, pt = 0.75, pe = 0.15)$	0.75	1118

TAB. B.6 – Filtrage préfixe-multihomés (pm)

Tomographie	valeur coupure	préfixes affectés	routes filtrées
$T_{2005-01}(pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75})$	$\beta = 0.4$ et $U = 1.25$	325	33 446
$T_{2006-08}(pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75})$	$\beta = 0.4$ et $U = 1.25$	755	

TAB. B.7 – Filtrage route-moas (rm)

particulier, On va sélectionner, pour chaque préfixe, les deux AS origines i_1, i_2 maximisant $D_r(\, ASORIG(p)(i) \,/\, ASORIG(p)\,)$ (la durée d'observation du préfixe avec l'AS origine i divisé par la durée d'observation du préfixe p, tous AS origines confondus) (voir (C) et (C')). Puis on examine le rapport $\beta(ASORIG(p))$ indiquant si l'AS origine dominant i_1 apparaît vraiment plus souvent que les autres AS origines.

Définition 49. *Filtrage route-MOAS (rm)*
On appelle filtre route-MOAS ou (rm), un filtre qui, quel que soit p un préfixe MOAS dont les rapports $D_c(ASORIG(p))$ et $\beta(ASORIG(p))$ sont faibles, supprime les routes à destination de p indiquant un autre AS origine que l'AS origine dominant.

Définition 50. *Filtrage préfixe-cédé (pc)*
On dira appliquer un filtre préfixe-cédé ou (pc) sur une tomographie P, si on retire les routes à destination des préfixes dont le rapport $D_c(ASORIG(p))$ est proche de 0 (ou nul) et pour lesquels le filtre (rm) n'a pas d'effet (valeur de $\beta(ASORIG(p))$ trop grande, c'est-à-dire que l'AS origine dominant n'apparaît pas majoritairement plus souvent que les autres AS origines).

Les filtres (pe,pm,rm,pc) permettent de traiter une bonne partie des préfixes litigieux en supprimant des routes de la tomographie. Cependant, si des cas de préfixes MOAS

Tomographie	valeur coupure	préfixes filtrés	routes filtrées
$T_{2005-01}(pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}))$	0.4	2	545
$T_{2006-08}(...)$	0.4	0	0

TAB. B.8 – Filtrage préfixe-cédé (pc)

Tomographie	préfixes filtrés	routes filtrées
$T_{2005-01}(pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}), pc_{0.4})$	689	204 018
$T_{2006-08}(pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}), pc_{0.4})$	755	

TAB. B.9 – Filtrage préfixe origine oscillant (poo)

subsistent, on supprimera ces derniers via le filtre (poo).

Définition 51. *Filtrage préfixe-origine-oscillant (poo)*
On appelle filtre préfixe-origine-oscillant ou (poo) , le filtre qui supprime les routes à destination des préfixes MOAS dont les filtres (pe, pm, rm, pc) n'ont pas d'effet (valeurs de $\beta(ASORIG(p))$ et $D_c(ASORIG(p))$ non discriminantes)

Finalement, après application de tous les filtres concernant les préfixes litigieux, on obtient une tomographie où chaque préfixe correspond à un unique AS origine lorsque le préfixe n'est pas multihomé et à plusieurs AS origines sinon. Les seuils des différents filtres permettent de gérer les cas où il manque des routes pour certaines dates.

B.3.3 Suppression des AS sources avec des tables partielles

Observation des NLRI agrégés

Considérons les préfixes d'une tomographie filtrées avec (ap), (pp) et (pt) et avec les filtres des préfixes MOAS ($pe, pm, \beta, U, pc, poo$). La figure B.7 indique les préfixes observés par le même nombre de next hop virtuels ou le même nombre d'AS sources. On remarque sur cette figure, qu'un préfixe n'est pas toujours vu par le même nombre de collecteurs ou d'AS sources. La majorité des préfixes sont vus soit vus par au moins 70 AS sources, soit par moins de 15 AS sources. De la même manière, un préfixe est vu soit par au moins 125 collecteurs, soit par moins de 25 collecteurs. Ceci indique qu'il manque des routes vers certains préfixes partant de certains AS sources ou que les AS font des agrégations de route.

Le phénomène d'agrégation se traduit par une variation des préfixes visibles au niveau des routeurs BGP d'un AS. Les préfixes dont les routes sont propagées jusqu'à un AS, dépendent de la position de l'AS dans l'Internet. Certains préfixes ne sont pas visibles par un AS car les routes reçues pour ces préfixes concernent des préfixes agrégés (les NLRI reçus sont moins spécifiques que les NLRI non visibles) ou des préfixes découpés (les NLRI

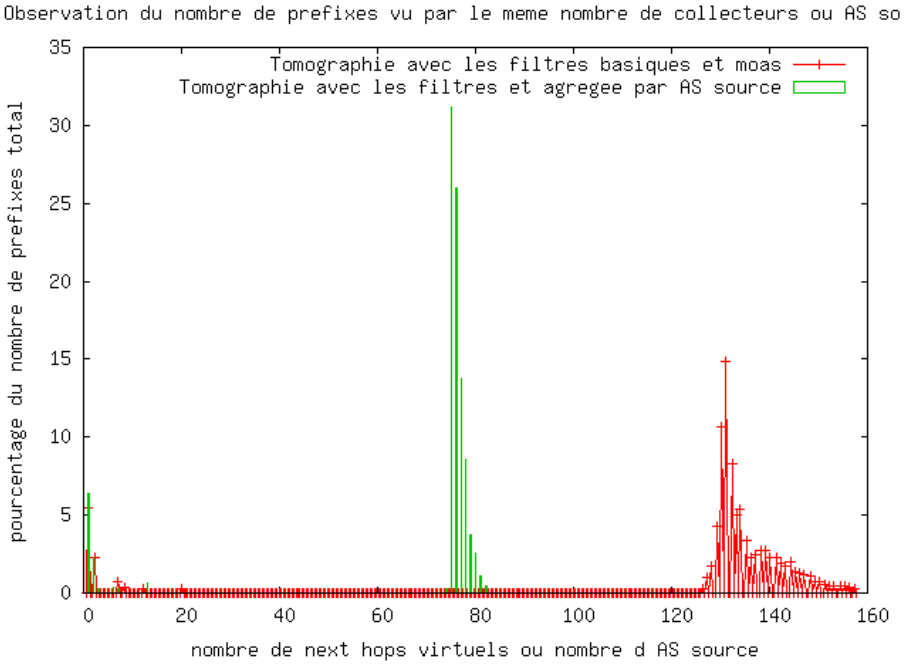


FIG. B.7 – Préfixes observés par le même nombre de collecteurs (next hop virtuels) ou le même nombre d'AS sources

Tomographie	Entrées	Collecteurs	Sondes	AS Sources	Next hop BGP
$T_{2005-01}$	28 267 547	3040	18	241	1116

TAB. B.10 – Le nombre d'entrées désigne le nombre de couples (next hop BGP virtuel, préfixe, route canonique, motif de répétition) indépendamment de la date d'observation

reçus sont plus spécifiques que les NLRI non visibles). La partie gauche de la figure B.7 indique un faible nombre de préfixes uniquement vu par certains AS ou certains routeurs. Le phénomène d'agrégation touche 10% des préfixes. Dans la suite, on définit le filtre (ea) qui permettra de filtrer les AS sources dont le nombre restreint de préfixes visibles n'est pas causé par les agrégations et les découpages de préfixes.

Filtrage des AS sources avec une vision complète

Le nombre de next hop virtuels pour les tomographies de test est reporté tableau B.10. Le regroupement des routes pour différents collecteurs d'un même AS source montre que certains AS sources ne fournissent que peu de routes car ils ne voient que peu de préfixes. Pour quantifier l'apport de chaque AS source, on peut s'intéresser par exemple au nombre de routes qu'un AS source fournit. Puis, plus précisément, on peut compter le nombre de préfixes visibles. Ce qui discrimine correctement les AS sources qui fournissent la quasi

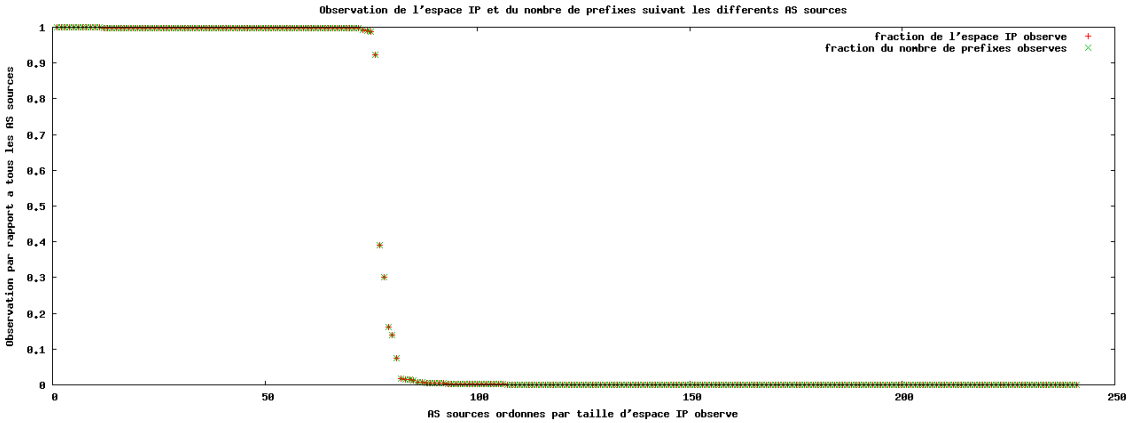


FIG. B.8 – Nombre de préfixes et fraction de l'espace d'adresses IP visible par chaque AS source. Les AS sources sont classés par fraction d'espace d'adresses IP visible

totalité de leurs préfixes de ceux qui n'exportent que quelques préfixes, est la partie de l'adressage IP couvert par les préfixes. On calcule pour chaque AS source, le rapport entre le nombre total d'adresses IP dans l'union des plages observées par l'AS source et le nombre total d'adresses IP dans l'espace total observé. L'indicateur de pourcentage d'espace d'adresse observé est très efficace pour détecter les tables de routage partielles en provenance de certains AS, puisque chaque routeur BGP est censé pouvoir joindre n'importe quel adresse parmi l'espace IP publique total[4].

Définition 52. *Filtre espace-AS source (ea)*
On appel filtre espace-AS source ou (ea), un filtre qui supprime les routes des AS sources dont l'espace IP total observé est en dessous d'un pourcentage minimum de l'espace d'adresse total. La valeur de coupure sera déterminée expérimentalement en fonction de chaque tomographie.

Ainsi, sur la figure B.8 on peut discerner deux types d'AS sources avec la valeur de coupure 0.85 (voir tableau B.11). Le filtre (ea) permet de conserver uniquement les AS sources de la partie gauche de la figure. La figure B.8 indique également que malgrès les agrégations, le nombre de préfixe permet une détermination approximative de l'espace d'adresses total observé.

Définition 53. *AS source virtuel*
On appelle AS source virtuel d'une tomographie, un AS pour lequel on peut déduire des

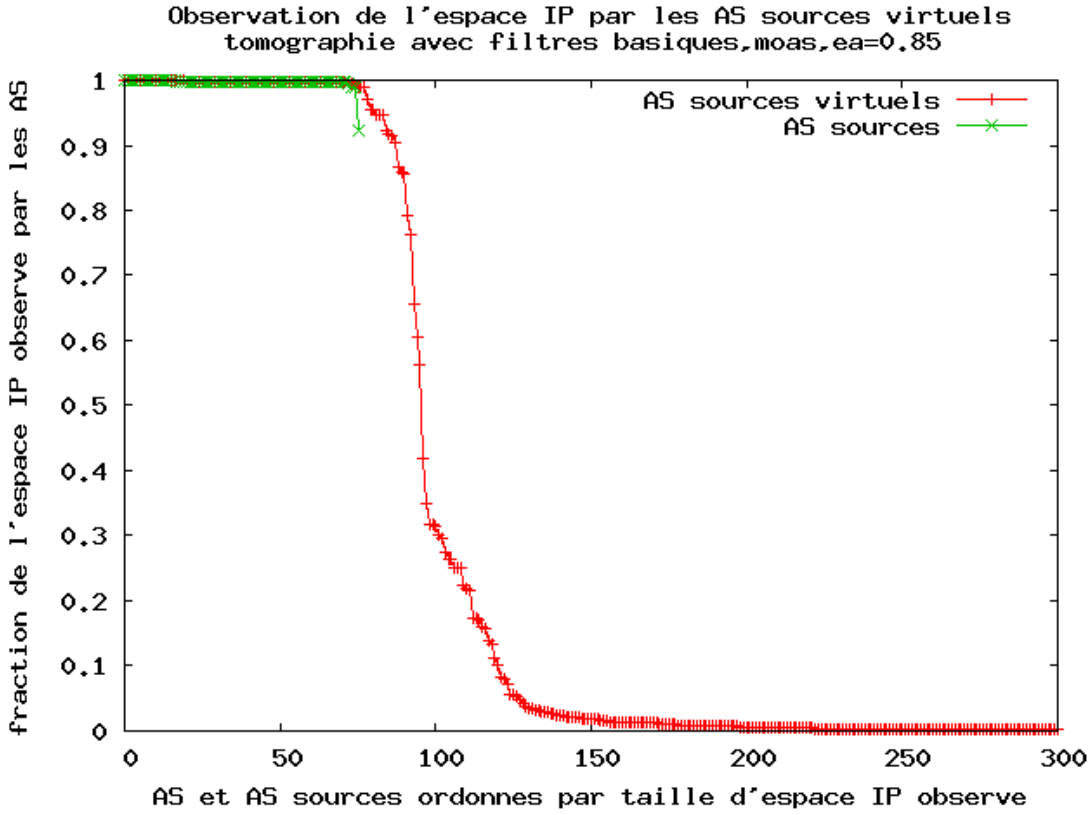
[4]sauf cas particuliers pour des plages d'adresse de petite taille.

Tomographie	$T_{2005-01}$	$T_{2006-08}$
déjà appliqués : $pp, ap, pt_{0.75}, pe_{0.15}, pm_{0.75}, rm(\beta_{0.4}, U_{1.25}), pc_{0.4}, poo$		
AS sources	241	322
AS sources ave plus de 85% de l'espace d'adresse	71	101
nombre de routes AS source vers préfixe filtrées	171 143	
nombre de routes next hop vers préfixe filtrées	412 460	

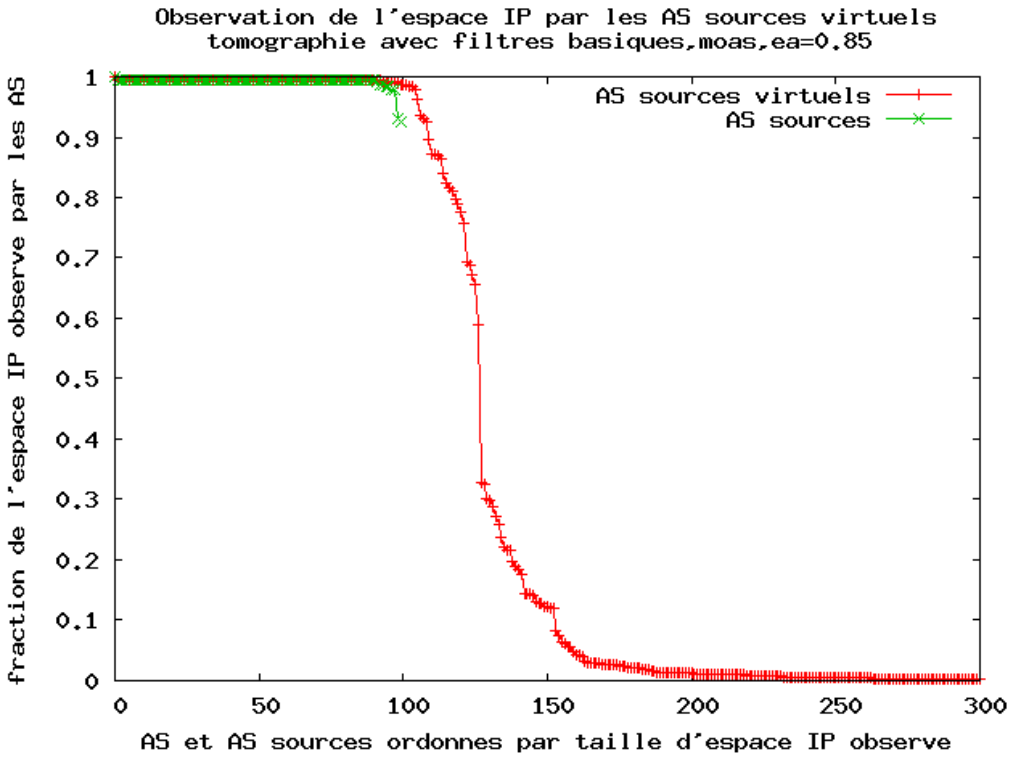
TAB. B.11 – Filtre espace-AS source (ea)

routes BGP en direction d'assez de préfixes pour obtenir un espace d'adresses IP suffisamment proche de l'espace d'adresses IP total observé avec la tomographie.

Les AS sources virtuels peuvent être des opérateurs Tiers-1 dans la mesure où on peut observer leurs routage pour la majeur partie des préfixes. Voir figure B.9.



(a) tomographie $T_{2005-01}$



(b) tomographie $T_{2006-08}$

FIG. B.9 – Fraction de l'espace d'adresses IP visible par chaque AS source et par chaque AS source virtuel.

Annexe C

Détermination des facettes du polyèdre de l'enveloppe convexe de points de l'espace

Cette partie de l'annexe présente une méthode pour déterminer un système d'équation minimal décrivant un polyèdre 0-1 dont les points extrèmes sont connus. Le système d'équation ainsi calculé décrit complètement l'enveloppe convexe des points. Cet algorithme est utilisé dans le chapitre 3 pour linéariser la formulation du problème MaxVCAP3 (voir d)).

Le problème résolu est NP-complet [147] d'autant plus que le nombre (fini) d'équations générées varie de façon exponentielle. Le principe de la méthode proposée est de rechercher parmi les différentes combinaisons de points 0-1 (tous points extrèmes) pouvant définir un hyperplan dont un des deux demi-espaces engendré serait valide vis à vis de l'enveloppe convexe de tous les points. La méthode différencie les cas où le polyèdre est de pleine dimension ou non. L'intérêt de notre approche est d'obtenir les équations analytiques des facettesdu polyèdre, ce qui n'a pas été possible avec les outils et algorithmes [1, 58, 59, 65] issus de la littérature. En effet, la majorité de ces algorithmes permettent de connaitre les points extrèmes d'un polyèdre à partir des équations des faces et facettes.

C.1 Notations et définitions

Nous commencons par rappeler quelques définitions de géométrie polyédrale. Pour plus d'informations sur ce sujet, on pourra se référer aux travaux suivants [57, 124, 205] et à

leurs citations.

Soient $n \geq 2$ un entier et l'ensemble $E = \{0,1\}^n$ contenant tous les points "0-1" de taille n.

Définition 54. *Enveloppe convexe, polytopes et polytopes {0,1}*
Un polytope P de $\mathbb{R}^n$ est l'enveloppe convexe de points de l'espace $\mathbb{R}^n$.
Soient k points $\{q_1, \ldots, q_k\}, k \geq 1$ de l'ensemble $\mathbb{R}^n$ appelés sommets du polytope. On note $Conv(\{q_1, \ldots, q_k\})$ l'enveloppe convexe des points telle que :

$$P = Conv(\{q_1, \ldots, q_k\}) = \left\{ x \in \mathbb{R}^n \middle/ \begin{array}{l} \exists \lambda_1, \ldots \lambda_k \geq 0 \\ \sum_{i=1}^{k} \lambda_i = 1 \\ x = \sum_{i=1}^{k} \lambda_i q_i \end{array} \right\}$$

Un polytope {0,1} est l'enveloppe convexe de points de E.

On définit la dimension d'un polytope comme le nombre maximum de points affinement indépendants moins 1. C'est la dimension du plus petit sous-espace affine qui contient P. Ainsi, s'il existe $n+1$ points affinement indépendants dans un polytope P, on dit que P est de pleine dimension ($dim(P) = n$).

Définition 55. *Polyèdre convexe*

On appele demi-espace de $\mathbb{R}^n$ un ensemble de la forme suivante :

$$\left\{ x \in \mathbb{R}^n / \sum_{j=1}^{n} \alpha_j . x_j \leq \beta, \, (\alpha_1, \ldots, \alpha_n, \beta) \in \mathbb{R}^n \setminus \{(0, \ldots, 0)\} \times \mathbb{R} \right\}$$

Un polyèdre convexe est l'intersection d'un nombre fini de demi-espaces. Un polyèdre peut être borné ou non. Autrement dit :
$\exists k \geq 1, \, (a_{1,1}, \ldots, a_{1,n}), \ldots, (a_{k,1}, \ldots, a_{k,n}) \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n, \, (b_1, \ldots, b_k) \in \mathbb{R}^k$, *tel que :*

$$P = \left\{ x \in \mathbb{R}^n \middle/ \forall i \in \{1, \ldots, k\}, \, \sum_{j=1}^{n} a_{i,j} . x_j \leq b_i \right\}$$

Remarque :polyèdre convexe borné est un polytope d'après le théorème de Minkownski-Weyl [172])

Définition 56. *Faces, faces propres et facettes*

Soit P un polyèdre de $\mathbb{R}^n$.
Soit le demi-espace $D = \{x \in \mathbb{R}^n / \sum_{i=1}^{n} a_i x_i \leq b\}$ pour un vecteur non nul $(a_1, \ldots, a_n) \in \mathbb{R}^n$ et un réel $b \in R$.
On note H l'hyperplan $\{x \in \mathbb{R}^n / \sum_{i=1}^{n} a_i x_i = b\}$.
On note F l'ensemble $\{x \in P / \sum_{i=1}^{n} a_i x_i = b\} = H \bigcap P$.
Le demi-espace D est dit valide pour P si $P \subset D$. Dans ce cas, l'ensemble F est appelé face de P. On dit que H et F sont définis par l'inégalité $\left\{x \in \mathbb{R}^n / \, a^T x \leq b\right\}$.
Une face F est dite propre lorsque $F = H \bigcap P \neq \varnothing$.
Une face F est appelé facette lorsque dim(F) = dim(P) -1 .
Une face de dimension 0 est appelé noeud ("vertex") ou point extrème de P (prouver qu'un noeud est un point extrème est reporté dans [124], proposition 1.17).
Un point extrème x de P est tel que $\forall x_1 \neq x_2 \in P, x \neq \frac{1}{2}x_1 + \frac{1}{2}x_2$.

Lemme 57. *Soit $k \geq 1 \in \mathbb{N}$. Soient k points deux à deux distincts $p_1, \ldots, p_k \in E$. On note $K = \{p_1, \ldots, p_k\}$ et $Conv(K)$ l'enveloppe convexe de K. Alors :*

$$\forall q \in E \backslash K,\ q \notin P$$

Preuve :
Montrons que $\forall q \in E \backslash K,\ q \notin P$ par contraposée.

Soit $q \in E \backslash K$. Supposons $q \in Conv(K)$.

Par défintion de l'enveloppe convexe de K, $\exists(\lambda_1, \ldots, \lambda_k) \in [0,1]^k$ tels que $q = \sum_{i=1}^{k} \lambda_i . p_i$ et $\sum_i \lambda_i = 1, \lambda_i \geq 0, \forall i$.

– Si $\exists i \in \{1, \ldots, k\} / \, \lambda_i = 1$, alors $q = p_i \in K$ ce qui est une contradiction. Donc $\forall i,\ \lambda_i < 1$.

Soit $i_0 \in \{1, \ldots, k\}$ tel que $\lambda_{i_0} \neq 0$. Donc Il existe i_1 tel que $\lambda_{i_1} \neq 0$ car $\forall i,\ \lambda_i < 1$ et $\sum_i \lambda_i = 1$.

On a $\forall j \leq n,\ (q)_j = \sum_{i=1}^{k} \lambda_i . (p_i)_j = \lambda_{i_0} . (p_{i_0})_j + \lambda_{i_1} . (p_{i_1})_j + \sum_{i=1, i \neq i_0, i \neq i_1}^{k} \lambda_i . (p_i)_j$.
Or $p_{i_0} \neq p_{i_1}$ donc $\exists j_2 \, / \, (p_{i_0})_{j_2} \neq (p_{i_1})_{j_2}$ et alors $(p_{i_0})_{j_2} + (p_{i_1})_{j_2} = 1$ car $p_{i_0}, p_{i_1} \in E$. On en déduit :

$$\sum_{i=1,i\neq i_0,i\neq i_1}^{k} \lambda_i.(p_i)_{j_2} + Min(\lambda_{i_0},\lambda_{i_1}) \leq (q)_{j_2} \leq \sum_{i=1,i\neq i_0,i\neq i_1}^{k} \lambda_i.(p_i)_{j_2} + Max(\lambda_{i_0},\lambda_{i_1})$$

$$0 < 0 + Min(\lambda_{i_0},\lambda_{i_1}) \leq (q)_{j_2} \leq \underbrace{\sum_{i=1,i\neq i_0,i\neq i_1}^{k} \lambda_i + Max(\lambda_{i_0},\lambda_{i_1})}_{=1-\lambda_{i_0}-\lambda_{i_1}+Max(\lambda_{i_0},\lambda_{i_1})} < 1$$

Ce qui montre finalement que $(q)_{j_2} \notin \{0,1\}$. Soit $q \notin E$ ce qui est une contradiction. □

Lemme 58. *Soit un entier $k \geq 1$. Soient k points distincts $p_1,\ldots,p_k$ de l'esemble E tels qu'il existe une sous-famille de $n+1$ points affinement indépendants. On note $K = \{p_1,\ldots,p_k\}$ et $Conv(K)$ l'enveloppe convexe de K (qui est de pleine dimension). Alors :*

1. *Il existe un système d'équation linéaire minimal I (à une multiplication de chaque égalité par un scalaire près) décrivant un polytope P dont l'intersection avec l'ensemble $E\backslash K$ est vide et dont l'ensemble des points extrêmes est K.*
2. *Chaque inégalité de I définit une facette de P.*
3. *Chaque hyperplan F de l'espace affine R^n contenant n points affinement indépendants parmi ceux de K définit un hyperplan. Cet hyperplan supporte une facette de P définissant alors une face propre de P, ou dans le cas contraire, chacun des deux demi-espaces définis par F n'est pas valide pour $P = Conv(K)$.*

Preuve :

1. Remarquons que nous avons supposé $Conv(K)$ comme étant de pleine dimension. Maintenant utilisons le lemme précédant et le résultat du corollaire 1.26 dans [124].
2. Voir corollaire 1.26 dans [124].
3. Soit H un hyperplan de $\mathbb{R}^n$ contenant au moins n points affinement indépendants de K. $dim(H) = n-1 = dim(P)-1$. Il y a alors deux possibilités :

   Cas 1 : Les deux demi-espaces définis par F ne sont pas valides pour $Conv(K)$,

   Cas 2 : Un des deux demi-espaces définit par F est valide pour $Conv(K)$.

   Dans ce cas, par construction, l'ensemble $F \bigcap P$ contient au moins n points affinement indépendants de K, ce qui prouve qu'il est une facette de $Conv(K)$.

□

Après avoir prouvé l'existence d'un tel système linéaire d'inéquations I pour n'importe quel ensemble de points K dont la dimension est n, nous proposons un algorithme de

génération des inéquations des hyperplans supportant les facettes dans ce cas. Puis, nous proposons un algorithme en deux phases permettant de générer les inégalités de I dans le cas général (ensemble K quelconque).

C.2 Algorithme de génération du système d'inéquations décrivant le polytope de l'enveloppe convexe

Nous commençons par décrire l'algorithme qui permet de calculer les inéquations décrivant l'enveloppe convexe d'un ensemble de points 0-1 où il existe une sous-famille de $n+1$ points affinement indépendants (cas où le polytope est de pleine dimension).

Algorithme 9. *Génération des facettes pour un polytope {0,1} de pleine dimension d'après ses points extrèmes (version avec énumération)*

Soit un ensemble de points K parmi les points de $E = \{0,1\}^n$ tel qu'il existe $n+1$ points affinement indépendants parmi K. L'enveloppe convexe de K notée P, est un polytope de pleine dimension. On peut aussi remarquer que $|K| \geq n+1$.

Cet algorithme détermine un ensemble d'hyperplans de $\mathbb{R}^n$ définissant un système d'inégalités minimal pour la description de P.

On définit l'ensemble G comme l'ensemble des permutations distinctes de n points de K : $G = \{\{x_1, \ldots, x_n\} / \forall i \in \{1, \ldots, n\}, x_i \in K\}$.

remarquons que $|G| = C^n_{|K|}$.

Entrée

$K = \{p_1, \ldots, p_k\} \in \{0,1\}^n$, $k \geq n+1$, il existe $n+1$ points de K affinement indépendants.

Sortie

$Facets = \{f_1, \ldots, f_m\} \in \mathbb{R}^{n+1} \times \ldots \times \mathbb{R}^{n+1}$.

$Vertices = \{V_1, \ldots, V_m\} \in 2^K \times \ldots \times 2^K$

Algorithme

if ! $TEST\_FULL\_DIMENSION(K)$: retourner ERREUR

$Facets := \varnothing$, $Vertices := \varnothing$

$R = \varnothing$, $S = \varnothing$, $Others = K$

$FIND\_FACETS(n, Facets*, Vertices*)$

retourner SUCCES

Procédure

$FIND\_FACETS(n, Facets*, Vertices*)$

procédure qui tente de trouver les facettes de P en cherchant des hyperplans H qui contiennent n points de K affinement indépendants.

On itère sur l'ensemble G pour trouver les combinaisons de points adéquats.

---

n : taille de l'espace.

$Facets$: liste de points de $\mathbb{R}^{n+1}$ qui sont des équations d'hyperplans.

$Vertices$: liste de sous-ensembles de points de K.

Pour chaque $g = \{p_{i_1}, \ldots, p_{i_n}\} \in G$,

Si $HAS\_BIGGER\_SET(Vertices, g)$ Alors

Aller à la prochaine étape de la boucle sur g.

fin

Soit M la matrice suivante :

$$M = \begin{pmatrix} (p_{i_2} - p_{i_1})_1 & \ldots & (p_{i_2} - p_{i_1})_n \\ & \ldots & \\ (p_{i_n} - p_{i_1})_1 & \ldots & (p_{i_n} - p_{i_1})_n \end{pmatrix}$$

Pour $i = 1 \ \ldots \ n$ faire

soit e_i le vecteur de taille n qui a sa i^{me} composante à 1.

soit N la matrice $n*n$ suivante :

$$N = \begin{pmatrix} \\ M \\ \\ \\ e_i \end{pmatrix}$$

Si $Determinant(N)$ est nul Alors

Aller à la prochaine étape de la boucle sur i.

fin

soit $X := (0 \ldots 0)$ et $b := (0 \ \ldots \ 0\,1)$.

$SOLVE\_NX\_Equal\_b(N, b, X)$

$y := p_{i_1}^T X$, $PointsSet := g$, $Under := 0$, $Upper := 0$.
Pour $p \in K \backslash g$ Faire

$z := X^T p - y$.
Si $z < 0$ Alors : $Under := Under + 1$.
Sinon si $z > 0$ Alors : $Upper := Upper + 1$.
Sinon : $PointsSet := PointsSet \bigcup \{p\}$

fin
Si $|Under| \neq 0$ et $|Upper| \neq 0$ Alors

Aller à la prochaine étape de la boucle sur g.

Sinon si $|Upper| \neq 0$
$y := -1 * y$.
$X := -1 * X$.
fin
$Vertices := Vertices \bigcup \{PointsSet\}$
$Facets := Facets \bigcup \{(X_1 \ \ldots \ X_n \ y)\}$
Aller à la prochaine étape de la boucle sur g.

fin

fin

Procédure
$HAS\_BIGGER\_SET(Vertices, g)$

Teste si un ensemble de points n'est pas déjà inclu dans un autre sous-ensemble appartenant à $Vertices$.

---

g : un ensemble de points de K.
retourne : Vraie si l'ensemble de points g est inclu dans un élément de $Vertices$, Faux sinon.

Procédure
$SOLVE\_NX\_Equal\_b(N, b, X*)$

Resoud le système d'équation $N.X = b$.

N : une matrice $n*n$.

b : un vecteur de taille n.

X : solution du système.

fin

fin de l'algorithme

L'enveloppe convexe d'un polyèdre qui n'est pas de pleine dimension est plus difficile à déterminer dans le sens où il y a plus de liberté dans la détermination des hyperplans définissant des facettes. Un idée simple consiste à trouver les hyperplans contenant tous les points de K puis de réduire la dimension du problème. Malheureusement, après réduction du problème (par un changement de base par exemple), les points du polyèdre ne sont plus en coordonnées 0-1, et le premier algorithme ne peux donc pas s'appliquer. Donc nous avons décidé de compléter la famille de points pour obtenir au moins une sous-famille de n points affinement indépendants (par un procédé de Gram-Schmidt). Nous appliquons alors notre premier algorithme qui nous donne un ensemble de demi-espaces. A cet ensemble, nous ajoutons chacun des deux demi-espaces définis par les points rajoutés. Nous appliquons finalement une vérification séquentielle de l'essentialité de chaque demi-espace de l'ensemble final (en utilisant la méthode reportée dans [57]).

Algorithme 10. *Génération des facettes d'un polytope {0,1} d'après ses points extrèmes*

Soit un ensemble de points K de $E = \{0,1\}^n$ dont l'enveloppe convexe est notée P.

Cet algorithme calcule les équations des hyperplans de $\mathbb{R}^n$ qui définissent le système minimal décrivant P.

Entrée

$K = \{p_1, \ldots, p_k\} \in \{0,1\}^n$.

Sortie

$Facets = \{f_1, \ldots, f_m\} \in \mathbb{R}^{n+1} \times \ldots \times \mathbb{R}^{n+1}$.

$Vertices = \{V_1, \ldots, V_m\} \in 2^K \times \ldots \times 2^K$.

Algorithme

$Q = \varnothing$

$Complete\_Point\_Set(K, *Q)$

$K' = K \bigcup Q$, $P' = ConvexHull(K')$.

Utiliser l'algorithme 1 pour calculer les facettes de P' qui est de pleine dimension.

$Facets = \{f_1, \ldots, f_m\} \in \mathbb{R}^{n+1} \times \ldots \times \mathbb{R}^{n+1}$.

$Vertices = \{V_1, \ldots, V_m\} \in 2^{K'} \times \ldots \times 2^{K'}$.

fin

Pour chaque point $q \in Q$ Faire

$f_q^+ = (q_1, \ldots, q_n, \sum (p_1)_i . q_i)$

$f_q^- = (-q_1, \ldots, -q_n, -\sum (p_1)_i . q_i)$

$Facets := Facets \bigcup \{f_q^+\} \bigcup \{f_q^-\}$

$Vertices := \{Vertices_0, \ldots, Vertices_{|Vertices|}, \{K\}, \{K\}\}$

fin

$Remove\_Redundant\_Inequalities(*Facets, *Vertices)$

Procédure

$Complete\_Point\_Set(K, *Q)$

Recherche des points de E pour former un ensemble K de dimension n.

K : points de E.

Q : points de $E \backslash K$ à déterminer.

Soit $V = \{p_2 - p_1, \ldots, p_n - p_1\}$.

En appliquant la méthode de Gram-Schmidt sur V dans l'espace vectoriel $\mathbb{R}^n$, on obtient :

$B = \{b_1, \ldots, b_p\}$ une base de l'espace vectoriel engendré par les vecteurs de V.

$R = \{r_1, \ldots, r_{n-p}\}$ un ensemble de vecteurs de $\mathbb{R}^n$ complétant B pour former une base de $\mathbb{R}^n$.

fin

Soit $H' = \{h'_1, \ldots, h'_{n-p} / h'_i = (\, (r_i)_1, \ldots, (r_i)_n, \sum_j (r_i)_j . (p_1)_j \,)\}$.

L'ensemble H définit une famille d'hyperplans contenant tous les points de K.

Pour chaque élément e de $E \backslash \{K \bigcup Q\}$ faire

Pour chaque $h = (h_1, \ldots, h_n, h_{n+1}) \in H'$ faire

Si $\sum_{i=1\ldots n} h_i.e_i \neq h_{n+1}$ Alors

Soit $e' = (e - p_1) - \sum_{j=1\ldots|B|}(e - p_1)_j.b_j * b_j$.
Si le vecteur e' est non nul Alors

$B = B \bigcup \{e'\}$
$Q = Q \bigcup \{e\}$
Si |Q| < |R| Alors retourner.

Sinon

Aller à la prochaine étape de la boucle sur e.

fin

fin

fin

fin

Procédure
$Remove\_Redundant\_Inequalities(*Facets, *Vertices)$

Tente de supprimer les inégalités redondantes dans l'esemble $Facets$ et supprime aussi les éléments correspondants dans $Vertices$.

$Facets$: liste d'équations d'hyperplans de $\mathbb{R}^n$.
$Vertices$: liste d'ensemble de points de E.

Pour chaque élements h, v dans $Facets, Vertices$ faire

$h = (h_1, \ldots, h_n, h_{n+1})$ décrit l'hyperplan :
$\{x \in \mathbb{R}^n \,/\, \sum_{i=1\ldots n} h_i.x_i \leq h_{n+1}\}$.

Soit $f^* = \begin{array}{l} maximisers^T x \\ s.c. \quad Ax \leq b \\ \qquad\quad (h_1, \ldots, h_n)^T x \leq h_{n+1} + 1 \end{array}$.

où le système $Ax \leq b$ correspond à l'appartenance de x à l'intersection des demi-espaces définis par les autres hyperplans éléments de $Facets$.
Si $f^* \leq h_{n+1}$ Alors :

Supprimer h de $Facets$ et v de $Vertices$.

fin

fin

fin

fin de l'algorithme

C.3 Résultats numériques

Notre algorithme est applicable lorsque le nombre de points du polytope dont on recherche les facettes est petit. En effet, le fait de rechercher parmi toutes les combinaisons de points ralentit l'exécution de l'algorithme. Pour accélérer la recherche, on peut ajouter des facettes initiales afin de réduire le nombre de combinaisons pour lesquelles on recherche une éventuelle facette. Les candidats potentiels sont par exemple les facettes de l'hypercube. On présente ici quelques résultats avec des ensembles de points quelconques.

Annexe D

Un arbre de stockage de plages d'adresses IP

Cette partie de l'annexe présente une structure de données destinée au stockage de plages d'adresses IP (version 4 ou 6). Chaque plage d'adresse représente un réseau. Cette structure de données est un arbre permettant l'insertion, la recherche et la suppression d'un réseau en $\Theta(log(n))$ où n est le nombre de réseaux stockés dans l'arbre. La recherche du plus petit réseau contenant une adresse ou un autre réseau a une complexité en $\Theta(log(n))$ également. Les propriétés de passage à l'échelle offertes permettent de stocker quelques centaines de milliers de réseaux et obtenir de bonnes perfomances. Notre structure de données n'offre pas un stockage spatial optimal, mais règle de façon simple la recherche de la plage contenante la plus spécifique ("Longest Matching Prefix").

D.1 Définitions et notations

Soit $m \in \mathbb{N}$ la taille en nombre de bits des adresses IP (32 pour IP version 4 et 128 pour IP version 6). Une adresse IP est représentée par un tableau de m bits. On note $\mathbb{A}$ l'ensemble des adresses possibles.

Définition 59. *Préfixes réseaux (CIDR)*
Un préfixe réseau désigne une plage d'adresses contigües de $\mathbb{A}$. Un préfixe réseau "a/l" est un objet muni d'une adresse a et d'une longueur de masque l. La longueur de masque est un entier entre 0 et m. Elle indique le nombre de bits invariables dans l'adresse du réseau. Les bits de poids faible de l'adresse a sont nuls à partir de la position l. Autrement dit si la taille du masque est nulle, l'adresse est nulle, et si la taille du masque est égale

à m, le réseau est une seule adresse (toutes les valeurs sont possibles).
On définit la taille d'un réseau par le nombre d'adresses possibles dans la plage correspondante : $|a/l| = 2^{m-l}$

On note $\mathbb{L}$ l'ensemble des longueurs de masque de réseau possibles et $\mathbb{P}$ l'ensemble des préfixes réseau possibles. Alors : $\mathbb{P} = \{a/l \text{ tel que } a \in \mathbb{A} \, et \, l \in \mathbb{L}\}$. Notons que l'ensemble des adresses $\mathbb{A}$ et l'ensemble des longueurs de masque $\mathbb{L}$ sont munis de la relation d'ordre naturelle sur $\mathbb{N}$. Remarquons aussi qu'un préfixe a/l représente la plage de l'adresse a jusqu'à l'adresse $a + |a/l|$.

Définition 60. *Tronquage d'adresse*
Soit a une adresse de A et $n \in \mathbb{N}$ tel que $0 \leq n \leq m$. On dit qu'on tronque l'adresse a à n bits lorque l'on place ses $m - n$ derniers bit à 0. On notera le résultat $a[n :]$.

En particulier, l'adresse d'un réseau de taille l est tronquée à l bits, pour tout $l \in \mathbb{L}$.
L'ensemble de préfixes réseau $\mathbb{P}$ est muni de la relation partielle d'inclusion $\subset$ telle que :

$$\forall p_1 = (a_1/l_1),\, p_2 = (a_2/l_2) \in \mathbb{P},\, \{p_1 \subset p_2\} \equiv \left\{ \begin{array}{l} a_1 = a_2 \text{ et } l_1 > l_2 \\ \text{ ou} \\ a_1 > a_2 \text{ et } a_1[l_2 :] = a_2 \end{array} \right\} (1)$$

Cette relation d'inclusion est la relation naturelle d'inclusion entre plages d'adresses. On remarque le fait suivant :

$$\forall p_1 = (a_1/l_1),\, p_2 = (a_2/l_2) \in \mathbb{P},\, \left\{ \begin{array}{l} p_1 \subset p_2 \\ \text{ou } p_2 \subset p_1 \\ \text{ou } p_1 \cap p_2 = \oslash \end{array} \right\} (2)$$

On dit d'un préfixe p est parent (resp. fils) d'un préfixe q si et seulement si $p \subseteq q$ (resp. $p \supseteq q$).

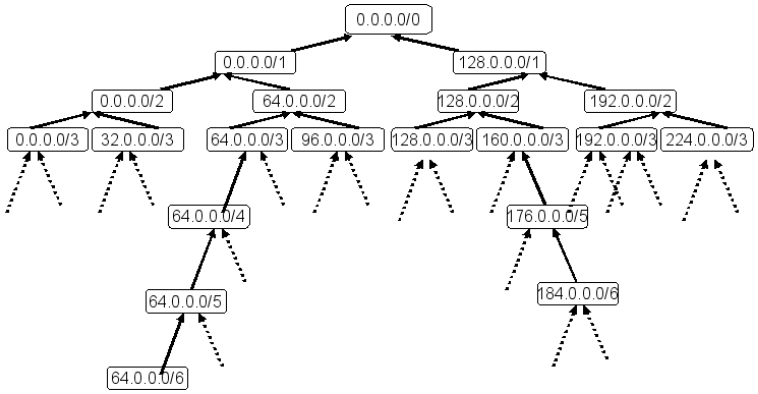


FIG. D.1 – Arbre binaire contenant tous les préfixes réseaux

D.2 Définition d'un arbre simple de préfixe réseaux

L'arbre maximum contenant tous les préfixes possibles est un arbre binaire de taille 2^m (cf. figure D.1).

Un arbre de stockage de préfixes réseau contient un sous ensemble des 2^m éléments de $\mathbb{P}$. Nous considérons le cas d'arbres doublement orientés où 2 préfixes $p \subset q$ sont reliés de façon symétrique si et seulement si $q = Min_{\subseteq} \{z \in \mathbb{P} / z \supset p\}$. Ainsi, un préfixe a un seul parent dans l'arbre (la racine n'en a pas) et plusieurs fils possibles (aucun pour les feuilles). La figure D.2 montre un exemple avec des préfixes IP version 4.

Une opération très courante en routage est la sélection du plus petit préfixe réseau contenant une adresse ou un réseau définie de la façon suivante :

Définition 61. *Plus petit préfixe contenant (Longest matching prefix)*
Soit E un ensemble de prefixes réseaux contenant le préfixe nul $0/0$. Pour tout préfixe p (issu de $\mathbb{P}$), le plus petit préfixe (au sens de l'inclusion) contenant p parmi les éléments de E est le préfixe vérifiant :

$$Min_{\subseteq}\{q \in E \text{ tel que } p \subseteq q\}$$

Notons qu'obtenir le plus petit préfixe contenant une adresse a donnée revient à rechercher le plus petit préfixe contenant le réseau a/m ($m = 32$ pour les adresses IP version 4). Pour tout ensemble de préfixe E, le plus petit préfixe contenant un élément quelconque

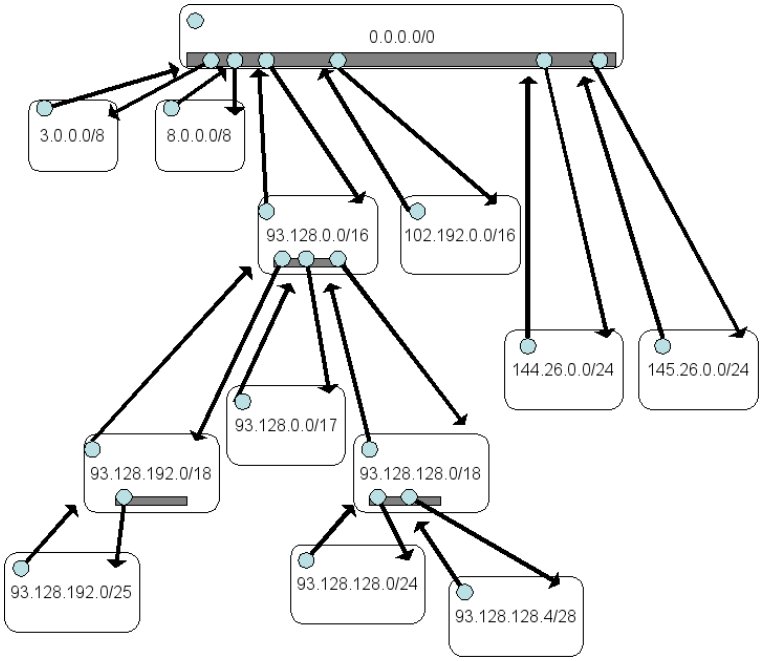


FIG. D.2 – Exemple d'arbre de préfixe

existe toujours si le préfixe 0/0 est élément de E. Par la suite, on supposera que tous les ensembles de préfixes considérés contiennent le réseau 0/0. Le stockage d'un arbre de préfixe nécessite un conteneur de noeuds (un noeud de l'arbre est détaillé figure D.3). Nous désirons définir un conteneur qui permet rapidement d'ajouter, rechercher et supprimer un élément. Pour cela nous choisissons un arbre rouge et noir [17] et un ordre total sur l'ensemble $\mathbb{P}$. L'arbre rouge et noir permet en une complexité $\Theta(log(n))$ d'effectuer les trois opérations de recherche, insertion, suppression. La structure définie alors est en fait un double arbre : un arbre maillé dans le sens des inclusions de plages d'adresses et un arbre de stockage des noeuds de l'arbre précédent. En définissant un ordre total particulier sur les noeuds de l'arbre d'inclusion, on peut aboutir à une recherche plus rapide du plus petit préfixe contenant un préfixe vis à vis de l'arbre de stockage.

Définition 62. *Ordre total sur les préfixes pour l'arbre rouge et noir*
On définit l'ordre total $<_{rb}$ *sur les préfixes réseaux de la manière suivante*

$$\forall x = (a_x/l_x), y = (a_y/l_y) \in P,\ x <_{rb} y \quad \equiv \quad \left\{ \begin{array}{l} a_x = a_y \ et\ l_x > l_y \\ \ ou \\ a_x > a_y \end{array} \right\}$$

Voici un exemple de classement de l'ordre $<_{rb}$:
$128.0/30 <_{rb} 10.128.128/24 <_{rb} 10.128.0/24 <_{rb} 10.128.0/17 <_{rb} 10.64.0/24 <_{rb} 10.0.0/24 <_{rb} 10.0.0.0/16 <_{rb} 4.0/8 <_{rb} 0/0$

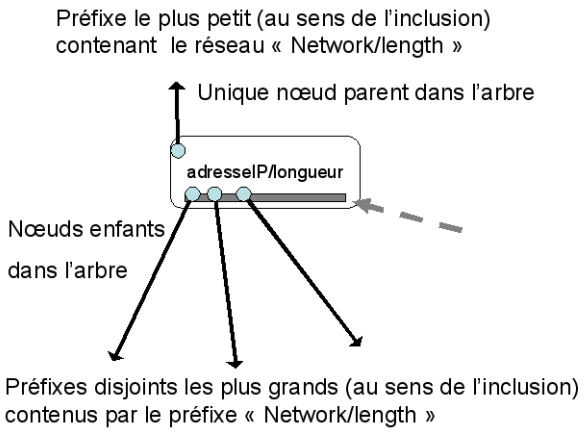


FIG. D.3 – Un noeud dans l'arbre de préfixes

On remarque que l'ordre total $<_{rb}$ vérifie les propriétés suivantes :

$$\forall x, y \in \mathbb{P}, \quad x \subset y \Rightarrow x <_{rb} y \quad \text{et} \quad x <_{rb} y \Rightarrow \left\{ \begin{array}{c} x \subset y \\ \text{ou} \\ x \not\subseteq y \text{ et } y \not\subseteq x \end{array} \right\}$$

Lemme 63. *Propriétés entre adresses de préfixes parent et fils*
$\forall x = (a_x/l_x), y = (a_y/l_y) \in \mathbb{P}, \quad x \subsetneq y \Rightarrow a_y \leq a_x < a_y + |y|$

Preuve :
Par définition de l'adresse d'un réseau r=(a/l), tous les bits à partir de la position l sont nuls. Ici les l_x premiers bits de a_x et a_y sont égaux car $x \subset y$, donc $a_x \leq a_y$.
La dernière adresse de la plage d'adresse du réseau y est $a_y + |y| - 1$.
ce qui implique $a_x \leq a_y + |y| - 1$.
Les arbres Rouge et noir disposent d'une propriété intéressante. Il est possible de trouver la borne inférieure d'un élément avec une complexité de l'ordre de $\Theta(log(n))$: $LowerBound(x) = Min_{<_{rb}}\{y \in arbre / y \not<_{rb} x\}$

Théoreme 64. *Recherche du plus petit préfixe contenant dans l'arbre*
En utilisant l'ordre total $<_{rb}$ dans l'arbre rouge et noir, la recherche du plus petit préfixe s'effectue avec une complexité de l'ordre de $\Theta(log(n))$.

Preuve :
Soit $x \in \mathbb{P}$ le préfixe dont on cherche le plus petit contenant.
Notons y^* le plus petit contenant de x.
$y^* = Min_{\subseteq}\{y \in arbre / x \subsetneq y\} = Min_{\subseteq} U^*$
Notons y^0 la borne inférieure de x dans l'arbre rouge et noir.

$y^0(x) = Min_{\leq_{rb}}\{y \in arbre / x < y \text{ ou } x = y\} = Min_{\leq_{rb}} U$ les deux ensembles U et $U*$ sont tels que $U* \subseteq U$.

si $x \in arbre$: y^* est le parent de $x = y^0$ dans l'arbre d'inclusion. La recherche de y^* s'effectue bien en $\Theta(log(n))$ (car recherche de y_0).

si $x \notin arbre$:

alors $y^0 = Min_{\leq_{rb}}\{y \in arbre / x < y\}$ puisque $y^0 \neq x$.

cas $x \subset y^0$: $y^0 \in U^* \Rightarrow y^0 \geq y^*$

$U^* \subseteq U \Rightarrow minU^* \geq minU \Rightarrow y^0 \leq y^*$

finalement $y^0 = y^*$.

cas $x \not\subseteq y^0$: par définition, $y^0 \geq_{rb} x$. Or $y^0(x) \neq x$, donc $y^0(x) \geq_{rb} x$ puis $a_{y^0(x)} + |y^0(x)| \leq a_x(1)$.

$U^*(x) \subseteq U(x) \Rightarrow minU^*(x) \geq minU(x) \Rightarrow y^0 \leq_{rb} y^*$

$x \not\subseteq y^0(x)$ donc $y^*(x) \neq y^0(x)$ par définition de $y^*(x)$.

Donc $y^*(x) >_{rb} y^0(x) \quad \Rightarrow \quad y^0(x) \subsetneq y^*(x)$ ou $y^0(x)$ et $y^*(x)$ disjoints

Montrons finalement que $y^0(x)$ et $y^*(x)$ sont disjoints, ce qui prouvera que $y^0(x) \subsetneq y^*(x)$, autrement dit que $y^*(x)$ est le premier parent de $y^0(x)$ qui contient x (le fait d'accéder au parent d'un préfixe dans l'arbre s'effectue en $\Theta(1)$).

Sachant que $x \subset y^*(x)$ et que $y^*(x) < x$ on a $a_{y^*(x)} \leq a_x < a_{y^*(x)} + |y^*(x)|$ (2).

Supposons que $y^*(x) \bigcap y^0(x) = \varnothing$:

$y^*(x) > y^0(x) \Rightarrow a_{y^*(x)} + |y^*(x)| \leq a_{y^0(x)} <^{(1)} a_x - |y^0(x)| \leq a_x <^{(2)} a_{y^*(x)} + |y^*(x)|$: contradiction.