HAL
open science

Security management in telecommunication systems : models, polyhedra and algorithms

Mohamed Yassine Naghmouchi

To cite this version:

Mohamed Yassine Naghmouchi. Security management in telecommunication systems : models, polyhedra and algorithms. Information Theory [cs.IT]. Université Paris sciences et lettres; 11067281X, 2019. English. NNT : 2019PSLED008 . tel-03220642

HAL Id: tel-03220642

https://theses.hal.science/tel-03220642

Submitted on 7 May 2021

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

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

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Université Paris-Dauphine

Security risk management in telecommunication systems: models, polyhedra and algorithms

Soutenue par
M.Yassine Naghmouchi
Le **17/06/2019**

École doctorale n°543
ED de Dauphine

Spécialité
Informatique

Composition du jury :

A. Ridha Mahjoub Professeur, Université Paris-Dauphine	*Directeur de thèse*
Nancy Perrot Ingénieur de recherche, Orange Labs	*Co-encadrant*
Nelson Maculan Professeur, Université fédérale, Rio de Janeiro	*Rapporteur*
Alain Quilliot Maître de Conférences, Université Pierre et Marie Curie	*Examinateur*
Pierre Fouilhoux Professeur, ESSEC Business school, Paris	*Examinateur*
Ivana Ljubic Professeur, University of Primorska	*Examinateur*
Vangelis Paschos Professeur, Université Paris- Dauphine	*Examinateur*
Adam Ouorou Directeur de domaine de recherche, Orange Labs	*Examinateur*

Dauphine | PSL

orange

Remerciements

Je voudrais exprimer tout d'abord toute ma gratitude et mes remerciements les plus sincères à mes encadrants de thèse qui ont su me pousser à donner le meilleur de moi-même.

Mon directeur de thèse, A. Ridha Mahjoub, a encadré mon projet de fin d'études en école d'ingénieurs au laboratoire LAMSADE. Il a ensuite encadré mon stage de master et cette thèse en collaboration avec Orange Labs. J'ai beaucoup bénéficié de la grande expérience et la haute compétence de Ridha dans le domaine de l'optimisation combinatoire et les approches polyhèdrales en particulier. J'ai mûri scientifiquement à côté de Ridha qui a su, tout au long de ce parcours, me transmettre son enthousiasme pour le travail de recherche soigneusement accompli. Pour toutes ces raisons, je lui témoigne ma plus sincère gratitude.

Nancy Perrot a co-encadré mon stage de master ainsi que cette thèse à Orange Labs. Ce fut une parfaite collaboration aussi bien sur le plan scientifique que humain. J'ai eu beaucoup de chance d'être encadré par Nancy qui m'a fait participer à plusieurs activités de grande importance. Je pense notamment au prestigieux Salon de Recherche Orange Labs (SROL), le dépôt d'un brevet et plusieurs formations très intéressantes au sein de l'entreprise. Au delà de l'aspect scientifique et professionnel, je suis reconnaissant à Nancy pour sa compréhension, son soutien et sa confiance qui m'ont toujours aidé et motivé en particulier dans les moments difficiles.

Je suis très reconnaissant pour l'attention que Monsieur Nelson Maculan, Professeur à l'Université fédérale Rio de Janeiro, a portée à mon travail. Je souhaiterais le remercier pour m'avoir fait l'honneur de rapporter ma thèse et pour ces remarques précieuses.

Je voudrais remercier Monsieur Alain Quilliot, Professeur à l'Université Clermont Auvergne, pour avoir accepté d'être rapporteur de cette thèse. Je le remercie également pour sa lecture approfondie du manuscrit et ses suggestions pertinentes.

Je suis heureux que Monsieur Vangelis Paschos, Professeur à l'Université

Paris-Dauphine a présidé le jury de ma thèse. Je souhaiterais également remercier Monsieur Pierre Fouilhoux, Maitre de conférence à l'Université Pierre et Marie Curie, Madame Ivana Ljubic, Professeur à ESSEC Business school, et Monsieur Adam Ouorou, Directeur du domaine sécurité à Orange Labs, pour avoir bien voulu examiner mes travaux et accepter de participer au jury.

Je souhaiterais remercier tous les collègues à Orange Labs et en particulier les membres de l'équipe TRM, à leur tête Nabil Benameur puis Eric Gourdin, pour avoir rendu très agréables les trois premières années de ma thèse. J'étais très heureux de partager mon bureau d'abord avec Felipe dans les locaux d'Issy-Les-moulineaux ensuite avec Christian et Léonce à Châtillon que je remercie pour leurs conseils et amitié. Je remercie Alain pour ses discussions intellectuelles, Philippe pour sa bienveillance, Yannick pour nous avoir toujours réservé la meilleure table de la cantine. Je remercie Amel qui m'a permis de bénéficier jusqu'à aujourd'hui de ses conseils et de son expérience de recherche. Je souhaite également remercier Vincent, Paul et Iaad, mes collègues doctorants chez Orange et Ahlam ma soeur de thèse. Je remercie vivement Adam, Jean-Philippe et Nizar de m'avoir fait bénéficier de leur expertise technique sur la cybersécurité. Je remercie infiniment mes collègues au LAMSADE avec lesquelles j'ai développé une forte relation d'amitié. Je suis également reconnaissant à mes collègues de thèse à EDF et au Lip6 pour leur amitié.

Enfin, je salue chaleureusement toute ma famille à Paris, à Nice et en Tunisie que j'aime beaucoup. Je souhaiterais remercier ma soeur Takoua pour son amour et pour avoir été toujours fière de moi. Je voudrais lui dire que je serai toujours là pour elle, et que je suis confiant dans tout le bien que lui réserve la vie. Mon père, qui est tout pour moi dans ce monde, merci pour ton soutien et ton amour inconditionnel, je suis fier d'être ton fils. Enfin, j'ai une pensée toute particulière pour ma mère à laquelle je voudrais dire : tu es la plus belle chose en moi, et c'est à toi, Papa et Takoua que je dédie cette thèse.

Abstract

In this thesis, we propose a new risk management framework for telecommunication networks. This is based on the concept of Risk Assessment Graphs (RAGs). These graphs contain two types of nodes: access point nodes, or starting points for attackers, and asset-vulnerability nodes. The latter have to be secured. An arc in the RAG represents a potential propagation of an attacker from a node to another. A positive weight, representing the propagation difficulty of an attacker, is associated to each arc. First, we propose a quantitative risk evaluation approach based on the shortest paths between the access points and the asset-vulnerability nodes. Then, we consider a risk treatment problem, called Proactive Countermeasure Selection Problem (PCSP). Given a propagation difficulty threshold for each pair of access point and asset-vulnerability node, and a set of countermeasures that can be placed on the asset vulnerability nodes, the PCSP consists in selecting the minimum cost subset of countermeasures so that the length of each shortest path from an access point to an asset vulnerability node is greater than or equal to the propagation difficulty threshold.

We show that the PCSP is NP-Complete even when the graph is reduced to an arc. Then, we give a formulation of the problem as a bilevel programming model for which we propose two single-level reformulations: a compact formulation based on LP-duality, and a path formulation with an exponential number of constraints, obtained by projection. Moreover, we study the path formulation from a polyhedral point of view. We introduce several classes of valid inequalities. We discuss when the basic and valid inequalities define facets. We also devise separation routines for these inequalities. Using this, we develop a Branch-and-Cut algorithm for the PCSP along with an extensive computational study. The numerical tests show the efficiency of the polyhedral results from an algorithmic point of view.

Our framework applies to a wide set of real cases in the telecommunication industry. We illustrate this in several practical use cases including Internet of Things (IoT), Software Defined Network (SDN) and Local Area Networks (LANs). We also show the integration of our approach in a web application.

Key words : Security management, modern telecommunication systems, bilevel programming, polyhedral approach, facets, Branch-and-Cut algorithm.

Résumé court

Dans cette thèse, nous proposons une nouvelle approche de gestion de risques pour les réseaux de télécommunications. Celle-ci est basée sur le concept de graphes d'analyse de risques appelés Risk Assessment Graphs (RAGs). Ces graphes contiennent deux types de noeuds : des points d'accès qui sont des points de départ pour les attaquants, et des noeuds appelés bien-vulnérabilité. Ces derniers doivent être sécurisés. La propagation potentielle d'un attaquant entre deux noeuds est représentée par un arc dans le RAG. Un poids positif représentant la difficulté de propagation d'un attaquant est associé à chaque arc. D'abord, nous proposons une approche quantitative d'évaluation de risques basée sur le calcul des plus courts chemins entre les points d'accès et les noeuds bien-vulnérabilité. Nous considérons ensuite un problème de traitement de risque appelé Proactive Countermeasure Selection Problem (PCSP). Etant donnés un seuil de difficulté de propagation pour chaque paire de point d'accès et noeud bien-vulnérabilité, et un ensemble de contremesures pouvant être placées sur les noeuds bien-vulnérabilité, le problème PCSP consiste à déterminer le sous ensemble de contremesures de coût minimal, de manière à ce que la longueur de chaque plus court chemin d'un point d'accès à un noeud bien-vulnérabilité soit supérieure ou égale au seuil de difficulté de propagation.

Nous montrons que le PCSP est NP-complet même quand le graphe est réduit à un arc. Nous donnons aussi une formulation du problème comme un modèle de programmation bi-niveau pour lequel nous proposons deux reformulations en un seul niveau: une formulation compacte basée sur la dualité en programmation linéaire, et une formulation chemins avec un nombre exponentiel de contraintes, obtenue par projection. Nous étudions cette deuxième formulation d'un point de vue polyhèdral. Nous décrivons différentes classes d'inégalités valides. Nous discutons l'aspect facial des inégalités de base et des inégalités valides. Nous concevons aussi des méthodes de séparation pour ces inégalités. En utilisant ces résultats, nous développons un algorithme de coupes et branchements pour le problème. Nous discutons enfin d'une étude numérique approfondie montrant l'efficacité des résultats polyhèdraux d'un point de vue algorithmique.

Notre approche s'applique à une large gamme de cas réels dans le domaine de télécommunications. Nous l'illustrons à travers plusieurs cas d'utilisation

couvrant l'internet des objets (IoT), les réseaux orientés logiciel (SDN) et les réseaux locaux (LANs). Aussi, nous montrons l'intégration de notre approche dans une application web.

Mots clés: gestion de la sécurité, systèmes de télécommunications modernes, programmation bi-niveau, approche polyhèdrale, facette, algorithme de coupes et branchements.

Résumé long

Dans notre monde de plus en plus numérique, les cyberattaques prennent de plus en plus d'importance chaque jour. En 2015, l'International Data Group (IDG) [10] a détecté 38% plus d'incidents de sécurité que l'année précédente. Selon l'Identity Theft Resource Center (ITRC) [5], en 2016, plus de 29 millions de dossiers ont été exposés à 858 violations. Récemment, en janvier 2019 seulement, 1,76 milliard d'enregistrements ont été divulgués [26]. Les incidents provoqués par les cyberattaques sont divers : attaques par déni de service, dégradations de sites Web, accès à des informations sensibles, attaques contre des infrastructures critiques, pannes logicielles et matérielles, etc. Ces attaques endommagent plusieurs secteurs, notamment la finance, le gouvernement, la santé et l'éducation.

La multiplication des cyberattaques cause de plus en plus de dommages aux entreprises, aux États et aux particuliers. Selon l'étude de l'Institut Ponemon sur les violations des données, 383 organisations ont subi au moins une violation en 2016. Le coût moyen par brèche est de 4 millions de dollars. La cybercriminalité a coûté 500 milliards de dollars à l'économie mondiale en 2015. Selon JUNIPER Research [14], elle atteindra 2 000 milliards de dollars d'ici fin 2019 et pourrait coûter 5 200 milliards de dollars aux entreprises au cours des quatre prochaines années selon Accenture [1].

La gestion de la sécurité est devenue une question urgente et il n'est pas surprenant que les gouvernements et les entreprises du monde entier recherchent de meilleures stratégies de gestion des risques. Par exemple, l'Agence européenne chargée de la sécurité des réseaux et de l'information [7] a organisé un exercice de cybersécurité depuis 2010, auquel ont participé des pays européens et plus de 200 organisations, dont des organismes publiques, des entreprises de télécommunications, des fournisseurs d'énergie, des institutions financières et des fournisseurs de services Internet. Orange s'engage notamment à assurer la sécurité de ses services et de ses données en essayant de proposer des approches innovantes de gestion des risques. Pour ce faire, nous devons relever de sérieux défis liés à la procédure de gestion de la sécurité. Dans cette thèse, nous nous concentrons sur les défis induits par l'évolution des systèmes de télécommunication ainsi que sur les questions économiques de la cyberdéfense et les questions mathématiques qui lui sont associées.

Avant de présenter ces défis, nous définissons le *le processus de gestion des risques de sécurité*. En général, il y a deux étapes principales dans la gestion des risques de sécurité [116] : l'évaluation des risques et le traitement des risques. L'évaluation des risques comporte trois étapes. Elle commence par l'identification des risques qui permet d'identifier les vulnérabilités, c'est-à-dire les faiblesses liées à tout objet de valeur à protéger. La deuxième étape est l'analyse des risques, qui consiste à déterminer le risque induit pour chaque vulnérabilité en évaluant l'impact, c'est-à-dire le degré des pertes, et la vraisemblance, c'est-à-dire la probabilité d'occurrence. L'étape de l'évaluation des risques se termine par une évaluation du niveau de risque en fonction de l'impact et de la probabilité.

En fonction de cette évaluation et d'un seuil de risque, des décisions relatives à la stratégie de protection et à sa mise en œuvre doivent être prises. En particulier, il est important de déterminer la stratégie de protection optimale qui gère le niveau de risque requis tout en minimisant les coûts de déploiement des *contre-mesures*, qui sont les actions ou dispositifs qui peuvent prévenir ou atténuer les effets d'une attaque. C'est l'objectif de l'étape finale de gestion de la sécurité qui est le traitement du risque. Le processus de gestion de la sécurité attire l'attention des chercheurs qui proposent différentes méthodologies afin de sécuriser au mieux les réseaux. Cependant, ils font face à de sérieux défis à chaque étape de la gestion des risques. Par exemple, les défis de l'évaluation des risques sont étroitement liés à l'évolution de l'industrie des télécommunications vers des systèmes plus complexes. D'autre part, le traitement des risques est plutôt associé à des défis économiques et mathématiques. Nous présentons ces défis dans ce qui suit.

Défis liés à l'évaluation des risques: Il est impossible de dissocier l'évaluation de la gestion des risques de sécurité et l'évolution des systèmes de télécommunication actuels. Les systèmes que nous visons à sécuriser aujourd'hui évoluent vers des architectures de plus en plus complexes. Dans le passé, les systèmes d'information étaient conçus statiquement et n'évoluaient presque pas pendant l'exécution, alors que les systèmes actuels (les réseaux virtuels [119], les clouds [131], plateformes de services via les APIs, les réseaux définis par logiciel [29], etc.) évoluent assez fréquemment. Les systèmes deviennent dynamiques de par leur conception. Ils s'appuient sur des technologies de virtualisation utilisées à différents niveaux de l'infrastructure. La virtualisation comprend le réseau (par exemple, Network Function Virtualization [79]), le système (par exemple, les technologies KVM, Xen et VMWare [53]), et même la couche application (distributed data storage [55]). Ces systèmes sont complexes en ce sens que : 1) ils comprennent un grand nombre d'éléments hétérogènes ; 2) ces éléments sont reliés par des interactions non linéaires, souvent de types différents (p. ex. liens physiques et virtuels) ; 3) font l'objet de déductions externes et internes (p. ex. attaquants) ; et 4) le système évolue au fil du

temps (p. ex. évolution de la topologie et possibilité d'exploiter des points vulnérables).

D'un autre côté, les attaquants deviennent de plus en plus intelligents et gênants pour de tels systèmes. Ils peuvent utiliser la topologie du réseau pour se propager d'un bien à un autre et exploiter leurs vulnérabilités associées, ce qui induit la notion de *propagation du risque*. De plus, les attaquants peuvent s'adapter à l'évolution dans le temps des vulnérabilités ainsi qu'à la topologie, qui maintient la propagation du risque dans le temps. Un autre facteur qui peut influencer de manière significative la propagation du risque est la fréquence d'accès entre les biens du système. En fait, le risque pourrait être plus élevé si les biens du système sont plus fréquemment connectés et vice versa. Nous appelons cette notion de fréquence d'accès *l'accessibilité*. Ainsi, pour gérer efficacement la sécurité des systèmes de télécommunication modernes, il faut tenir compte des vulnérabilités, de la topologie du réseau, des accessibilités et de la façon dont elles évoluent toutes au fil du temps. Il s'agit là d'un défi de taille pour les méthodes actuelles d'évaluation des risques.

À partir de notre étude bibliographique, nous avons étudié deux méthodes bien connues d'évaluation des risques : *les méthodes de scoring* [6, 31] et *les modèles basées sur les graphes* tels que les graphes d'attaque et les graphes de dépendance [127]. Sur la base d'une référence commune, les méthodes de scoring attribuent un score à chaque vulnérabilité rencontrée. Cependant, l'une de leurs principales limites est l'analyse statique et qualitative des vulnérabilités. En outre, les méthodes de scoring ne tiennent pas compte de la notion de propagation du risque, qui est cruciale pour sécuriser correctement les systèmes. Par conséquent, ces méthodes ne peuvent pas être utilisées seules pour évaluer les risques des systèmes modernes de télécommunication.

Les modèles basés sur les graphes incluent les vulnérabilités élémentaires qui peuvent être identifiées dans un système cible, et leurs relations, afin de montrer comment une succession d'étapes élémentaires peut potentiellement permettre à un attaquant d'obtenir des privilèges en profondeur dans le système. Pour cela, les modèles de graphes proposent d'intégrer la topologie dans le processus d'évaluation des risques. Du point de vue de la propagation du risque, cela est crucial car la description de la topologie capture les relations causales entre les biens du système qui permettent de prendre en compte la propagation des attaquants dans le système. Toutefois, cela n'est pas suffisant pour procéder à une évaluation des risques à grain fin dans les systèmes de télécommunication modernes. En fait, à notre connaissance, l'accessibilité entre les biens du système n'a jamais été prise en compte dans ces méthodes basées sur les graphes. Pour cette raison, les modèles de graphes ne suffisent pas à évaluer efficacement les risques des systèmes actuels.

En conclusion, nous devons étendre les méthodologies existantes en pro-

posant des modèles rigoureux de risque qui tiennent compte à la fois des vulnérabilités, de la topologie et de l'accessibilité. Ces modèles devraient tenir compte de l'évolution de ces facteurs au fil du temps et tenir compte de tous les attaquants possibles ainsi que des scénarios d'attaque. Le modèle doit également nous permettre d'évaluer avec précision le risque du système afin de superviser efficacement sa sécurité. Dans ce qui suit, nous discutons des défis économiques et mathématiques associés au traitement des risques.

Défis liés au traitement des risques: D'un point de vue économique, deux axes majeurs ont retenu l'attention de la communauté de la sécurité : le coût des cybercrimes et celui de la cyberdéfense. Dans cette thèse, nous nous concentrons sur les coûts de la cyberdéfense. Selon la société d'études de marché Gartner [9], les dépenses mondiales consacrées à la cybersécurité s'élèveront à 80 milliards de dollars en 2016. D'ici 2020, les entreprises du monde entier devraient dépenser environ 170 milliards de dollars, ce qui représente un taux de croissance de près de 10% par rapport à 2015. Par conséquent, il est essentiel de prendre en compte les coûts de la cyberdéfense dans le traitement des risques.

En effet, en utilisant les risques système évalués lors de l'étape d'évaluation des risques et certaines exigences de sécurité donnant par exemple un seuil de risque système, on peut identifier si le système est sécurisé ou non. Si le risque dépasse le seuil, une alerte pourrait être envoyée pour commencer le traitement du risque qui peut consister à déployer des contre-mesures sur certains actifs du système afin de réduire le risque global. Toutefois, si une contre-mesure peut réduire les risques du système, son déploiement peut s'avérer coûteux. Il est donc très important de trouver dans le système les contre-mesures qui garantissent la sécurité dans un certain sens, à un coût minimal. Cette question peut être formulée mathématiquement sous forme d' *un problème d'optimisation combinatoire* [122]. Succinctement, un problème d'optimisation combinatoire est le problème de trouver la meilleure solution à partir de toutes les solutions réalisables compte tenu de certaines contraintes. Dans notre contexte, l'ensemble des solutions réalisables ne sont rien d'autre que toutes les contre-mesures possibles qui répondent aux exigences de sécurité (les contraintes), et la meilleure solution est celle avec un coût minimal.

On peut voir ce problème d'optimisation comme un "jeu" entre un défenseur et plusieurs attaquants. Les attaquants essaient de trouver le chemin le plus facile pour accéder à leur cible. Mais ils sont obligés d'agir selon une certaine hiérarchie. En fait, le défenseur choisira l'emplacement des contre-mesures afin de sécuriser le système en rendant la propagation plus difficile pour les attaquants. Pour faire cela à moindre coût, le défenseur anticipera toutes les réactions des attaquants à ses décisions. Du point de vue de l'optimisation, ceci est un problème de programmation à deux niveaux [51]: Un problème

d'optimisation (le leader) ayant d'autres problèmes d'optimisation paramétrés (les followers) comme partie de ses contraintes. Cela nous amène à d'importantes questions mathématiques. En fait, nous devons choisir le modèle de programmation à deux niveaux qui permet d'assurer le plus haut niveau de sécurité à un coût minimal. Ensuite, nous devons concevoir des algorithmes innovants afin de résoudre efficacement le modèle à deux niveaux dans un délai raisonnable. Pour cela, une étude théorique rigoureuse du problème d'optimisation est essentielle. En fait, une telle étude peut renforcer la résolution du problème d'un point de vue algorithmique.

L'un des nouveaux sujets de programmation à deux niveaux les plus populaires pour résoudre les problèmes de sécurité est *Shortest Path Network Interdiction Problems* (SPNIPs) [86] qui peut être défini comme suit. Étant donné un graphe [46] $G = (V, A)$ où V est l'ensemble des noeuds et A est l'ensemble des arcs, une source s dans V (l'attaquant), une cible t dans V et un poids positif associé à chaque arc, le SPNIP consiste à maximiser la longueur du plus court chemin entre s et t, afin de rendre la propagation plus difficile pour l'attaquant, soit en interdisant les arcs [92] soit en interdisant les nœuds [36, 47]. Ces problèmes présentent encore certaines limites auxquelles il faut remédier. En fait, en supprimant un nœud, nous supprimons complètement le risque qui lui est associé. Par conséquent, les SPNIPs ne nous permettent pas d'envisager des contre-mesures réalistes permettant de réduire l'effet d'un risque sans l'éliminer complètement.

Par conséquent, nous avons besoin d'un modèle à deux niveaux plus général qui améliore les SPNIPs en considérant des contre-mesures réalistes ainsi que des sources et des cibles multiples. En d'autres termes, au lieu de supprimer un nœud donné, on peut payer un prix donné pour augmenter la longueur des arcs en cours de ce nœud, afin de rendre plus difficile l'accès d'un attaquant à ce nœud. De plus, si l'augmentation de la longueur des arcs en cours d'un noeud donné conduit à une valeur très élevée, alors l'interdiction équivaut à supprimer le noeud et le problème se réduit au SPNIP classique [47].

Contributions

Dans cette thèse, nous proposons d'abord une nouvelle approche d'évaluation des risques qui répond aux défis des systèmes modernes de télécommunication. Basé sur la théorie des graphes [46, 137], nous introduisons le concept de *Risk Assessment Graphs*. (RAG). Un nœud dans le RAG est soit *un point d'accès*, c'est-à-dire le point de départ d'un attaquant, soit *un noeud bien-vulnérabilité* à sécuriser. Nous introduisons la potentialité et l'accessibilité comme paramètres essentiels pour la définition des RAG. Les deux sont des fonctions du temps et indiquent respectivement la probabilité d'exploitation d'un nœud dans le RAG, et la fréquence d'accès entre les nœuds. Un arc dans

le RAG représente la propagation potentielle d'un attaquant d'un noeud à un autre. A chaque arc est associé un poids positif représentant la difficulté de *propagation* d'un attaquant qui est obtenu à partir d'une combinaison du potentiel et de l'accessibilité. Les graphiques que nous proposons prennent en compte à la fois les vulnérabilités, la topologie du système, l'accessibilité et l'évolution dans le temps. Ils permettent d'analyser les systèmes en capturant les accessibilités topologiques ainsi que les informations de sécurité en termes de vulnérabilités et leurs potentialités. Ils tiennent compte non seulement de l'état actuel du système, mais aussi de la façon dont il évolue au cours d'une période donnée. En outre, tous les attaquants et scénarios d'attaque possibles sont explicitement considérés comme des *chemins* dans les RAGs.

En outre, nous proposons une nouvelle approche quantitative d'évaluation des risques qui utilise les RAGs pour calculer des mesures de sécurité innovantes à savoir *la difficulté de propagation le risque propagé le risque par neoud* et *le risque global*. La mesure de sécurité de base et la plus importante est la difficulté de propagation qui est définie, par rapport à un point d'accès s, un nœud bien-vulnérabilité t et un chemin P de s à t, comme la somme des poids des arcs de P. Comme son nom l'indique, la difficulté de propagation indique combien il est difficile pour un attaquant de se propager sur P. Le chemin $s-t$ le plus court qui est le chemin de difficulté de propagation minimale est considéré comme *le chemin le plus probable* pour un attaquant. Du point de vue de la protection, un chemin est dit *sécirisé* si sa difficulté de propagation est supérieure à un seuil donné de difficulté de propagation. Un système sera considéré comme *sécurisé* lorsque tous les chemins de chaque point d'accès à chaque noeud bien-vulnérabilité le sont. L'efficacité de notre mesure de la difficulté de propagation consiste à réduire la vérification de la sécurité de l'ensemble du système à ceux des chemins les plus probables. En fait, la difficulté de propagation du chemin $s-t$ le plus probable est le minimum sur tous les chemins $s-t$. Par conséquent, si le chemin $s-t$ le plus probable est sécurisé, il en est de même pour tous les chemins $s-t$. De plus, nous illustrons l'approche d'évaluation des risques dans un cas d'utilisation de SDN et nous effectuons des simulations numériques pour montrer la sensibilité de nos métriques à la potentialité, la topologie et les changements d'accessibilité. Les simulations sont représentées dans la Figure 1 et la Figure 2.

Le problème d'optimisation que nous considérons s'appelle le Proactive Countermeasure Selection Problem (PCSP) et peut être défini comme suit. Une instance du PCSP est donnée par un triplet (G, K, D):

- $G = (V, A)$ est le RAG défini comme suit : l'ensemble des noeuds V est partionné en deux sous-ensembles S et T où $V = S \cup T$ et $S \cap T = \emptyset$. Un nœud dans S représente un point d'accès et un nœud dans T représente une paire bien-vulnérabilité. L'ensemble des arcs A est défini de telle sorte que pour tous les $u, v \in V$, un arc de u à v existe si $v \notin S$ et son

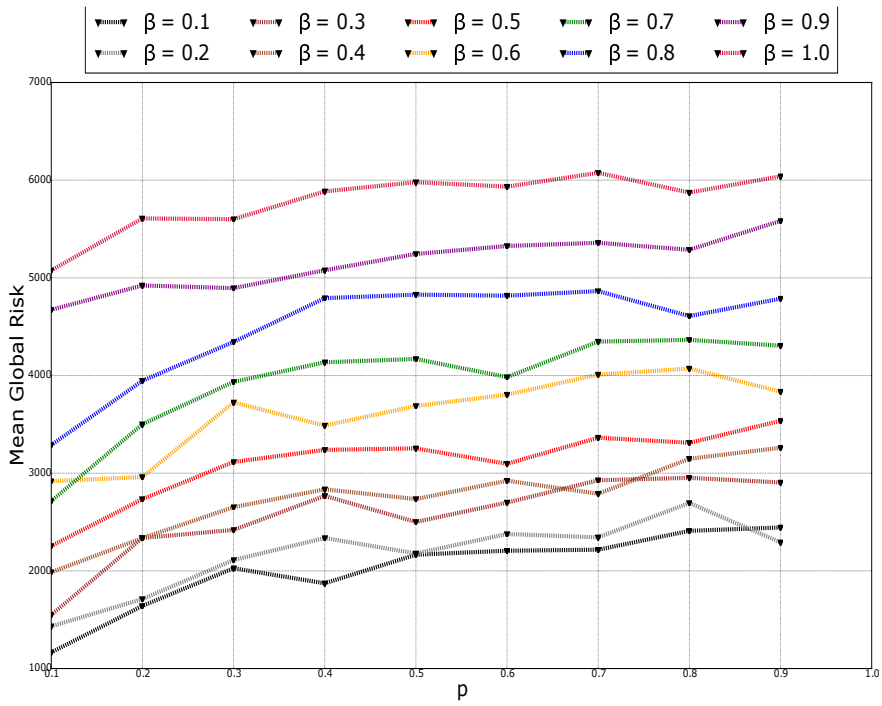


Figure 1: Impact de la topologie et la rapidité de convergence de l'accessibilité sur le risque (p et β)

exploitation depuis u est possible. A chaque arc $(u, v) \in A$ est associé un poids $w_{uv} \in \mathbb{R}_+$ représentant la difficulté de propagation d'un attaquant de u à v.

- $K = \{(t, k) : k \in K_t, t \in T\}$ est un ensemble de contre-mesures disponibles tel que K_t est l'ensemble des contre-mesures associées à t. Le placement de k sur t a un coût positif $c_k^t \in \mathbb{R}+$, et donne une augmentation d'un effet positif $\alpha_t^k \in \mathbb{R}_+$ dans les poids des arcs entrants en t.

- $D = (d_t^s)_{s \in S, t \in T} \in \mathbb{R}_+$ est un vecteur de seuil de difficulté de propagation.

Le PCSP consiste à sélectionner un ensemble de contre-mesures $K' \subseteq K$ de coût minimum de sorte que les *contraintes de sécurité* soient respectées : pour chaque $(s, t) \in S \times T$ la longueur du plus court chemin entre s et t après avoir placé K' est supérieur ou égale à d_s^t.

Nous montrons que le PCSP est NP-complet même lorsque le RAG est réduit à un arc. Nous formulons ensuite le problème comme un modèle de programmation à deux niveaux dans lequel le leader joue le rôle du défenseur et le follower le rôle de l'attaquant. Nous utilisons la dualité pour convertir le

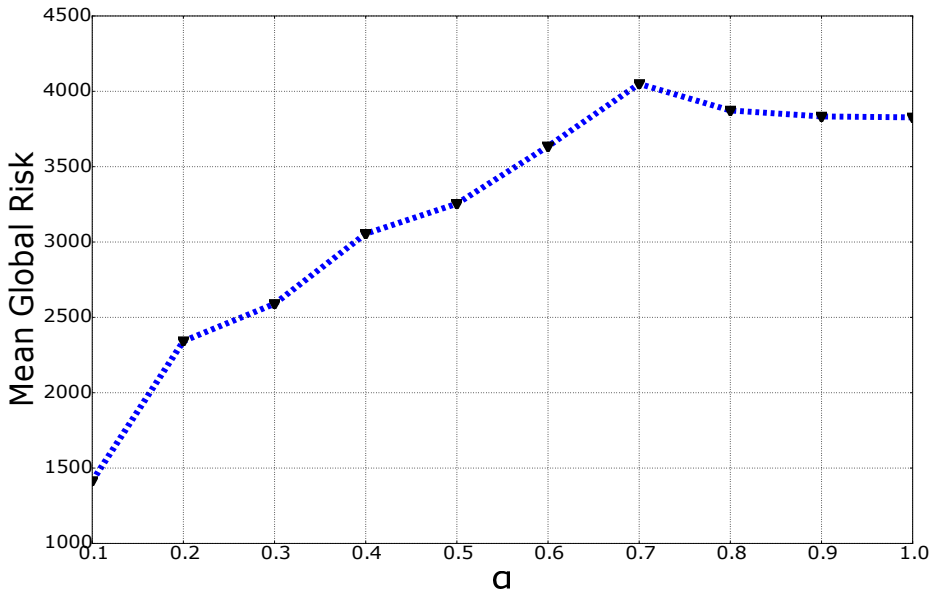


Figure 2: Impact de la rapidité de convergence de la potentialité α sur le risque

modèle à deux niveaux en une formulation compacte PCSP1 à un seul niveau qui est directement résolue à l'aide du solveur ILP Cplex [2]. Nous donnons également une deuxième formulation en projetant la formulation compacte sur un sous-ensemble de variables. Ceci induit *une formulation chemin* avec un nombre exponentiel de contraintes appelées *inégalités de base*, et sera résolu à l'optimum en utilisant un algorithme Branch-and-Cut. De plus, nous étudions les conditions d'optimalité pour le PCSP qui permettent d'identifier certaines inégalités qui sont vérifiées par toute solution optimale du problème. Ceci peut être utilisé pendant le prétraitement afin d'améliorer l'aspect algorithmique. Les deux formulations PCSP1 et PCSP2 sont donnés par:

$$\text{PCSP1: Min} \sum_{(t,k)\in K} c_t^k x_t^k$$

$$\lambda_t^{st} - \lambda_s^{st} \geq d_s^t, \tag{1}$$

$$\sum_{w\in\Gamma^+(v)} z_{vw}^{st} - \sum_{u\in\Gamma^-(v)} z_{uv}^{st} = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{if } v \notin \{s,t\} \\ -1 & \text{if } v = t \end{cases} \quad \forall v \in V, \tag{2}$$

$$\lambda_v^{st} - \lambda_u^{st} \leq w_{uv} + \sum_{k\in K_v} \alpha_v^k x_v^k \quad \forall uv \in A, \tag{3}$$

$$\sum_{uv\in A} (w_{uv} z_{uv}^{st} + \sum_{k\in K_v} \alpha_v^k y_{k,uv}^{st}) = \lambda_t^{st} - \lambda_s^{st}, \tag{4}$$

$$y_{k,uv}^{st} \leq 1/2(x_v^k + z_{uv}^{st}) \quad \forall uv \in A, k \in K_v, , \tag{5}$$

$$y_{k,uv}^{st} \geq x_v^k + z_{vu}^{st} - 1 \quad \forall uv \in A, k \in K_v, \tag{6}$$

$$x_t^k \in \{0,1\} \quad \forall (t,k) \in K, \tag{7}$$

$$z_{uv}^{st} \in \{0,1\} \quad \forall uv \in A, \tag{8}$$

$$y_{k,uv}^{st} \in \{0,1\} \quad \forall uv \in A, t \in K_v, \tag{9}$$

$$\lambda_v^{st} \text{ free} \quad \forall v \in V. \tag{10}$$

$$\text{PCSP2: Min} \sum_{(t,k)\in K} c_t^k x_t^k$$

$$\sum_{ij\in P} \sum_{k\in K_j} \alpha_j^k x_j^k \geq d_s^t - L_G(P) \quad \forall s \in S, t \in T, P \in P_{s,t}, \tag{11}$$

$$x_t^k \in \{0,1\} \quad \forall (t,k) \in K. \tag{12}$$

Une méthode très efficace qui peut aussi renforcer significativement l'aspect algorithmique est l'approche polyèdrale [101]. Une autre contribution consiste en une étude polyèdrale pour la formulation chemin. Nous caractérisons la dimension du polytope en considérant le *les contre-mesures essentielles*, c'est-à-dire les contre-mesures telles que si nous enlevons au moins une parmi elles, le PCSP n'a pas de solution. Nous présentons ensuite plusieurs classes d'inégalités valides et discutons l'aspect faciale de ces inégalités. Cette investigation fournira une bonne base pour l'étude algorithmique.

De plus, nous utilisons les résultats polyèdrales dans un algorithme Branch-and-Cut pour la formulation chemin. Nous développons une phase de prétraitement en considérant les équations de contre-mesures essentielles et les inégalités de conditions d'optimalité. Nous concevons des routines de séparation pour les inégalités de base et valides. Nous proposons également une heuristique primale pour permettre un élagage rapide des branches inintéressantes de l'arbre et ainsi accélérer l'algorithme Branch-and-Cut. De plus, nous

présentons des tests numériques de la formulation compacte et de la formulation chemin. Le but de l'étude computationnelle est d'examiner, d'un point de vue algorithmique, l'efficacité des résultats polyèdrale. Pour cela, nous étudions l'impact des conditions d'optimalité et les inégalités valides dans la résolution du problème. De plus, nous étudions la sensibilité de notre algorithme à la densité du graphe et au nombre de nœuds. Les tests sont exécutés sur des instances aléatoires et réalistes.

Afin de varier le type d'instances aléatoires, nous générons des instances avec différentes densités et tailles de S et T. Ensuite, pour chaque $\mid S \mid$ fixe et $\mid T \mid T \mid$ fixe, nous générons une instance (G, K, D) du PCSP comme suit :

- Le sous-graphe induit par l'ensemble des noeuds T est un Erdős - Rényi random graph [59] de paramètres $|T|$ et p, où p est la probabilité d'existance d'un arc dans le graphe.
- Nous connectons chaque point d'accès dans S à un certain nombre de nœuds en T choisis uniformément entre 1 et $\mid T \mid$.
- Nous générons uniformément un poids positif pour chaque arc dans l'intervalle $[0, 100]$.
- Nous fixons le même seuil d_s^t choisi aléatoirement dans l'intervalle $[0, 100]$ pour tous $(s, t) \in S \times T$.
- Chaque nœud a un nombre de contre-mesures entre 0 et 10 choisies uniformément dans l'ensemble des contre-mesures décrites dans le tableau 1.

Name	effect	cost
k_1	10	5
k_2	20	10
k_3	30	20
k_4	40	30
k_5	50	40
k_6	60	90
k_7	70	80
k_8	80	70
k_9	90	60
k_{10}	100	50

Table 1: L'ensemble de countre-measures

Cette procédure aléatoire de génération d'instances sera utilisée pour construire deux familles d'instances désignées par F_p et $F_{S,T}$:

- F_p : nous fixons $\mid S \mid = 50$, $\mid T \mid = 100$ et modifions le paramètre p dans $\{0.1, 0.2, \ldots, 1\}$. Ensuite, pour chaque p, nous générons aléatoirement, comme décrit ci-dessus, une famille de cinq instances qui seront notées par I_p.
- $F_{S,T}$: nous fixons $p = 0.3$ et faisons varier en même temps le nombre de nœuds $\mid S \mid$ et $\mid T \mid$. Pour chaque $\mid S \mid$ et $\mid T \mid$, une famille de cinq instances notées $I_{S,T}$ est générée aléatoirement comme décrit précédemment.

Pour chaque famille de cinq instances, nous rapporterons les résultats moyens.

Les instances réalistes que nous considérons sont obtenues à partir de la bibliothèque *SNDlib* [25]. Le sous-graphe induit par T est un graphe SNDlib. Nous mettons $\mid S \mid = 10$ et nous connectons aléatoirement chaque $s \in S$ à quelques noeuds dans T de taille entre 1 et 5. Les poids des arcs sont choisis aléatoirement dans l'intervalle $[1, 5]$. Nous fixons le même seuil d_s^t dans l'intervalle $[20, 40]$ pour chaque couple $(s, t) \in S \times T$. Pour chaque nœud bien-vulnérabilité, nous associons exactement 5 contre-mesures choisies aléatoirement parmi un ensemble de 30 contre-mesures décrites dans le tableau 2.

Nous présentons certains résultats dans les Tables 3, 4, 5 et 6 dont les colonnes sont les suivantes :

Nom	effet	coût
k1	0.5	1
k2	1	10
k3	0.8	100
k4	3	50
k5	2	150
k6	5	300
k7	4	250
k8	7	200
k9	6	400
k10	9	350
k11	8	450
k12	11	650
k13	10	550
k14	13	600
k15	12	500
k16	15	700
k17	14	800
k18	17	750
k19	16	1000
k20	19	900
k21	18	950
k22	21	850
k23	20	1050
k24	23	1250
k25	22	1150
k26	25	1200
k27	24	1100
k28	27	1400
k29	26	1350
k30	29	1450

Table 2: Ensemble de contre-mesures pour les instances réalistes

$\|V\|$	:	nombre de nœuds dans le graphe G ;
$\|S\|$	:	nombre de points d'accès;
$\|T\|$	:	nombre de nœuds bien-vulnérabilité;
$\|A\|$	:	nombre d'arcs;
$\|K\|$	:	nombre de contre-mesures;
$\|\Gamma\|$	:	nombre d'attaquants;
p	:	la probabilité des arcs du graphe Erdős - Rényi induit par T;
I_x	:	nom de l'instance aléatoire, où x est p ou (S, T);
$\|K^*\|$	:	nombre de contre-mesures essentielles;
N	:	nombre de noeuds dans l'arbre Branch & Cut;
OI	:	nombre d'inégalités de conditions d'optimalité générées;
Sec	:	nombre d'inégalités de sécurité générées;
PCI	:	nombre des inégalités PCI générées;
$CmPI$	:	nombre d'inégalités $CmPI$ générées;
Opt	:	la valeur de la solution optimale;
$NOpt$	:	le nombre d'instances résolues en optimalité par rapport au nombre total d'instances;
Gap	:	l'erreur relative entre la meilleure borne supérieure (la solution optimale si le problème a été résolu à l'optimalité) et la borne inférieure obtenue à la racine;
CPU	:	le temps total de résolution (en hh:mm:ss).

					Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	2.2	694.4	24.6	0	43.6	0.06	0:01:2	5/5
I_20,200	500.2	627.2	667.2	6.4	2.8	932.2	98.6	12	153	0.08	0:12:2	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	8.2	1027.8	643.8	12.6	177	0.08	0:49:3	5/5
I_40,400	1864.8	11910.4	981.8	37.4	1.2	2555.6	944.2	22.2	63.8	0.09	3:49:3	3/5
I_50,500	2676.4	1009.6	1674.4	30.8	16.2	4170.4	365.4	7	157.2	0.09	3:22:3	4/5
I_60,600	3276.2	2988.2	1613	40.2	5	3987.4	476.2	8.6	79.8	0.10	1:52:3	3/5
I_70,700	3988.4	2534.8	1876.2	27.8	15.2	4456.6	434.4	18.8	104	0.09	2:57:6	3/5
I_80,800	4694	2987.2	2854.4	63.2	15.4	4483	629.8	7.4	92.6	0.13	3:52:7	2/5
I_90,900	6204.6	1910.4	2765.4	52.2	11	12019	761.8	6	79	0.12	3:19:5	3/5
I_100,1000	10866.4	7479.4	3596	53.8	6	15888.8	613	7	107	0.19	4:42:3	1/5
I_110,1100	14444.6	15986.8	4987.2	67.2	12	14327.2	752	13	103	0.23	4:49:5	1/5
I_120,1200	19546.6	21907.6	5503.6	55	7.2	14230.6	644.4	7.2	69	0.37	-	0/5

Table 3: Efficacité de l'algorithme Branch and Cut dans la résolution de la famille $F_{S,T}$

					Compact formulation				Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	34	0.07	0:1: 5	5/5	2.2	694.4	24.6	0	43.6	0.06	0:01:2	5/5
I_20,200	500.2	627.2	667.2	6.4	70.8	0.09	1:38:2	5/5	2.8	932.2	98.6	12	153	0.08	0:12:2	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	116	0.16	4:10:5	1/5	8.2	1027.8	643.8	12.6	177	0.08	0:49:3	5/5
I_40,400	1864.8	11910.4	981.8	37.4	120.2	0.39	-	0/5	1.2	2555.6	944.2	22.2	63.8	0.09	3:49:3	3/5
I_50,500	2676.4	1009.6	1674.4	30.8	138	0.23	3:41:1	1/5	16.2	4170.4	365.4	7	157.2	0.09	3:22:3	4/5
I_60,600	3276.2	2988.2	1613	40.2	88.2	0.41	-	0/5	5	3987.4	476.2	8.6	79.8	0.10	1:52:3	3/5
I_70,700	3988.4	2534.8	1876.2	27.8	57.8	0.49	-	0/5	15.2	4456.6	434.4	18.8	104	0.09	2:57:6	3/5
I_80,800	4694	2987.2	2854.4	63.2	68.2	0.51	-	0/5	15.4	4483	629.8	7.4	92.6	0.13	3:52:7	2/5
I_90,900	6204.6	1910.4	2765.4	52.2	54	0.45	-	0/5	11	12019	761.8	6	79	0.12	3:19:5	3/5
I_100,1000	10866.4	7479.4	3596	53.8	-	-	-	0/5	6	15888.8	613	7	107	0.19	4:42:3	1/5
I_110,1100	14444.6	15986.8	4987.2	67.2	-	-	-	0/5	12	14327.2	752	13	103	0.23	4:49:5	1/5
I_120,1200	19546.6	21907.6	5503.6	55	-	-	-	0/5	7.2	14230.6	644.4	7.2	69	0.37	-	0/5

Table 4: Comparaison de la formulation chemin (avec Branch and Cut) et la formulation compacte pour la famille $F_{S,T}$

							basic formulation				Branch and Cut							
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*Sec*	*OI*	*PCI*	*PCmI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	4	11	193	0.08	0:00:02	7	20	24	2	20	0.06	0:00:01	10360
janos-us	36	26	94	260	130	0	2	28559	0.09	0:52:35	0	18	220	0	7953	0.07	0:13:29	13201
nobel-germany	27	17	36	49	85	5	2	1405	0.11	0:00:26	2	9	79	3	52	0.07	0:00:07	14060
dfnwin	20	10	55	61	50	1	1	272	0.07	0:00:07	1	16	95	0	47	0.06	0:00:12	10400
pioro40	50	40	99	54	200	0	11	16646	0.11	0:16:01	7	76	88	0	148	0.05	0:00:25	14050
india35	45	35	90	50	175	0	4	420	0.11	0:00:18	4	39	165	0	39	0.07	0:00:10	10460
cost266	47	37	67	51	185	0	8	147149	0.09	1:30:09	6	41	115	0	261	0.07	0:00:30	14200
geant	32	22	46	38	110	0	2	7233	0.11	0:02:45	2	34	76	2	148	0.07	0:00:15	14800
sun	37	27	112	270	135	3	19	77364	0.12	2:40:24	7	34	233	5	2837	0.08	0:05:28	16573
atlanta	25	15	32	42	75	1	0	1149	0.05	0:00:20	0	17	33	0	48	0.04	0:00:04	11102
nobelu	38	28	51	44	140	3	14	2713	0.09	0:01:16	7	31	62	4	47	0.04	0:00:06	13250
janos-us-ca	49	39	132	390	195	2	3	2068	0.06	0:06:33	3	29	239	0	502	0.04	0:01:30	12720
newyork	26	16	59	74	80	2	0	368	0.08	0:00:12	2	25	148	0	17	0.06	0:00:07	10100
dfnwin	21	11	57	57	55	1	11	145	0.10	0:00:03	3	11	59	1	11	0.07	0:00:02	9960
germany50	60	50	98	67	250	3	6	5759	0.10	0:04:36	3	84	210	1	69	0.09	0:00:23	12460
norway	37	27	61	117	135	2	3	7848	0.13	0:05:02	3	33	117	0	74	0.02	0:01:18	14700
diuan	21	11	52	45	55	1	7	4768	0.08	0:01:52	4	26	50	1	70	0.06	0:00:12	11150
giul39	49	39	182	390	195	1	14	720	0.05	0:02:32	8	40	453	0	296	0.04	0:01:05	12810

Table 5: Efficacité de l'algorithme Branch and Cut pour les instances réalistes

							compact formulation			Branch and Cut							
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*N*	*Gap*	*CPU*	*Sec*	*OI*	*PCI*	*PCmI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	0	0	0	0:00:01	7	20	24	2	20	0.06	0:00:01	10360
janos-us	36	26	94	260	130	0	134	0.09	0:18:15	0	18	220	0	7953	0.07	0:13:29	13201
nobel-germany	27	17	36	49	85	0	1	0.03	0:00:01	2	9	79	3	52	0.07	0:00:14	14060
dfnwin	20	10	55	61	50	0	1	0.03	0:00:01	1	16	95	0	47	0.06	0:00:12	10400
pioro40	50	40	99	54	200	0	0	0	0:00:01	7	76	88	0	148	0.05	0:00:25	14050
india35	45	35	90	50	175	0	0	0	0:00:01	4	39	165	0	39	0.07	0:00:10	10460
cost266	47	37	67	51	185	0	0	0	0:00:01	6	41	115	0	261	0.07	0:00:30	14200
geant	32	22	46	38	110	0	0	0	0:00:01	2	34	76	2	148	0.07	0:00:15	1480
sun	37	27	112	270	135	0	40	0.12	0:18:18	7	34	233	5	2837	0.08	0:05:28	16573
atlanta	25	15	32	42	75	0	0	0	0:00:01	0	17	33	0	48	0.04	0:00:04	11102
nobelu	38	28	51	44	140	0	0	0	0:00:01	7	31	62	4	47	0.04	0:00:06	13250
janos-us-ca	49	39	132	390	195	0	166	0.06	0:24:13	3	29	239	0	502	0.04	0:01:30	12720
newyork	26	16	59	74	80	0	0	0	0:00:01	2	25	148	0	17	0.06	0:00:07	10100
dfnwin	21	11	57	57	55	0	0	0	0:00:01	3	11	59	1	11	0.07	0:00:02	9960
germany50	60	50	98	67	250	0	0	0	0:00:01	3	84	210	1	69	0.09	0:00:23	12460
norway	37	27	61	117	135	0	0	0	0:00:02	3	33	117	0	74	0.02	0:00:18	14700
diuan	21	11	52	45	55	0	0	0	0:00:01	4	26	50	1	70	0.06	0:00:12	11150
giul39	49	39	182	390	195	0	12	0.05	0:16:10	8	40	453	0	296	0.04	0:01:05	12810

Table 6: Comparaison de la formulation chemin (avec Branch and Cut) et la formulation compacte pour les instances realistes

Notre travail répond à la fois à des exigences de flexibilité et de généralité, et s'applique à un large éventail d'applications. Dans cette thèse, nous illustrons l'approche complète de gestion des risques que nous développons dans plusieurs cas d'utilisation pratique, dont Internet of Things (IoT) [35] et Software Defined Network (SDN). Nous montrons également l'intégration de notre framework dans une application web que nous avons développée et illustrée dans un cas d'utilisation Local Network Areas (LANs). Cette application web couvre l'ensemble du cadre de gestion des risques que nous proposons dans cette thèse, y compris l'évaluation et le traitement des risques, ainsi qu'une interface de visualisation comme le montre les Figures 3 et 4.

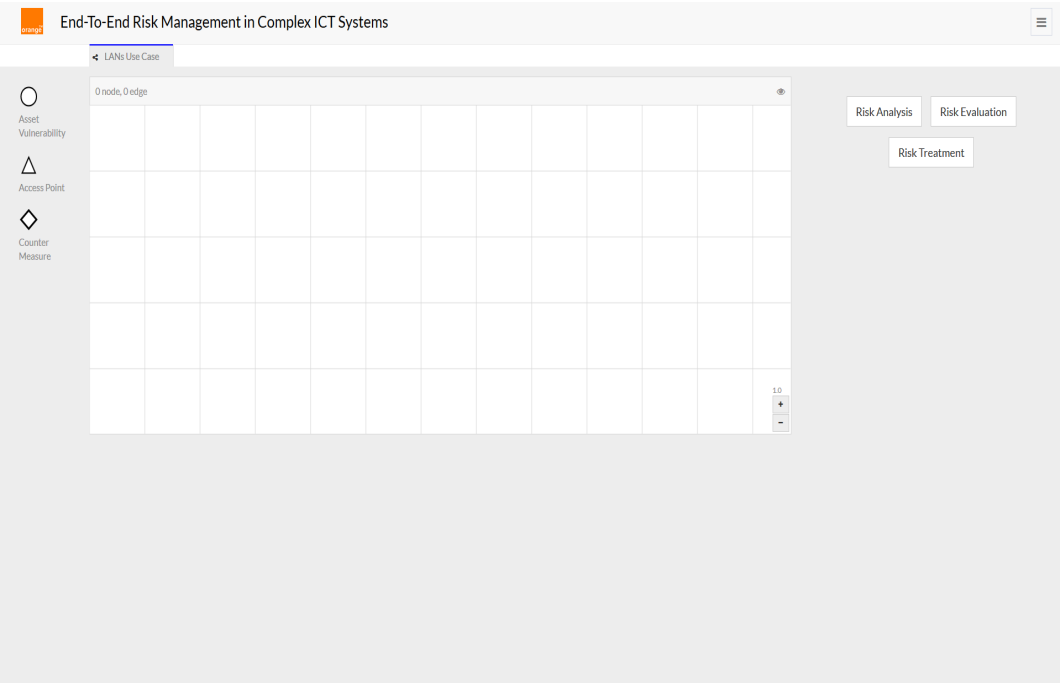


Figure 3: Vue d'ensemble de l'application

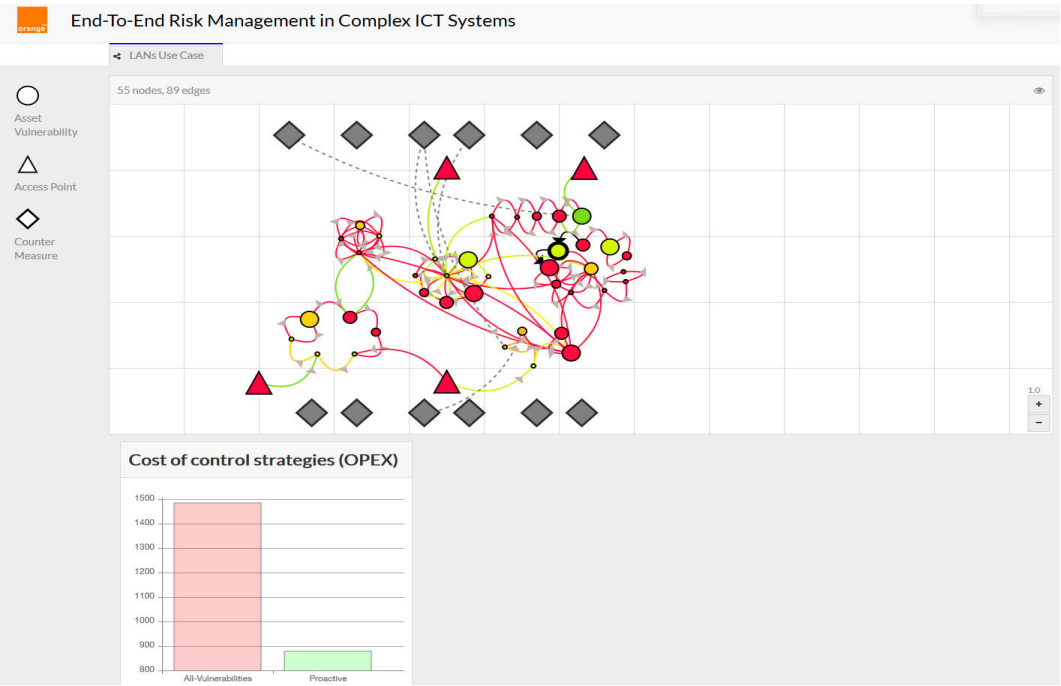


Figure 4: Placement et coût des contre-mesures

Contents

Table of Contents 1

Introduction 1

1 Security risk management and modern telecommunication systems 9

1.1 Security risk assessment and management 10

1.1.1 Basic concepts 10

1.1.2 Risk assessment state of the art 12

1.1.2.1 Scoring methods 12

1.1.2.2 Graph-based methods 13

1.2 Modern telecommunication systems and associated security challenges 15

1.2.1 Software Defined Networks (SDN) 15

1.2.2 Next generation 5G networks 17

1.2.3 Internet of Things (IoT) 18

1.3 Concluding remarks 19

2 Mathematical background 21

2.1 Graph theory: definitions and notations 22

2.1.1 Undirected graphs 22

2.1.2 Directed graphs 24

2.1.3 Graph classes 26

2.2 Combinatorial optimization 28

2.3 Algorithmic and complexity theory 29

2.4 Polyhedral approach and Branch-and-Cut 30

2.4.1 Elements of the polyhedral theory 30

2.4.2 Cutting plane method 34

2.4.3 Branch-and-Cut algorithm 35

2.4.4 Primal heuristics 37

2.5 Bilevel Programming 38

2.5.1 History: Stackelberg games 38
2.5.2 Generality 39
2.5.3 State of the art: Shortest Path Network Interdiction Problems (SPNIPs) 43
2.6 Concluding remarks 44

3 Security risk assessment: models and risk evaluation algorithm 45
3.1 Approach overview 46
3.1.1 Risk analysis 46
3.1.2 Risk evaluation 48
3.2 The Risk Assessment Graphs (RAGs) 49
3.2.1 Security metrics 49
3.2.2 The RAGs model 51
3.3 Most likely paths-based risk evaluation approach 52
3.3.1 Risk propagation: the most likely path 52
3.3.2 Risk evaluation algorithm 54
3.4 SDN case study 56
3.4.1 The Risk Assessment Graphs 56
3.4.2 Risk Evaluation 58
3.5 Simulations 59
3.5.1 Random systems generation 60
3.5.2 Impact of the number of nodes 61
3.5.3 Impact of the topology and the accessibility changes p and β 61
3.5.4 Impact of the potentiality convergence speed α 62
3.6 Concluding remarks 63

4 PCSP bilevel programming model, reformulations and optimality conditions 66
4.1 Definition and complexity 67
4.1.1 Problem statement 67
4.1.2 PCSP complexity 68
4.2 Problem examples 70
4.3 The Bi-level Model 72
4.3.1 The follower problem 72
4.3.2 The bilevel formulation 73
4.4 Single-Level Reformulations 74
4.4.1 Compact Single-Level Formulation 74
4.4.2 Path formulation by projection 75

4.5 Optimality conditions: dominance of countermeasures 78
4.6 Concluding remarks 80

5 The PCSP: a polyhedral investigation 81
5.1 ILP formulation and the associated polytope 82
5.2 Dimension of $PCSP(G, K, D)$ 83
5.2.1 Essential Countermeasures 83
5.2.1.1 Characterization of essential countermeasures 84
5.2.1.2 An instance of PCSP with essential countermeasures 86
5.2.2 Dimension 86
5.3 Facial investigation of basic inequalities 89
5.3.1 Trivial Inequalities 89
5.3.2 Security inequalities 90
5.4 Valid inequalities and facial aspect 93
5.4.1 Path Covering Inequalities (PCI) 94
5.4.2 Countermeasures Path Inequalities (CmPI) 97
5.4.3 Essential -by Subsets Removing- Countermeasures (ESRC) inequalities 100
5.5 Concluding remarks 106

6 Branch-and-Cut algorithm and computational study 107
6.1 Branch-and-Cut algorithm 108
6.1.1 Preprocessing 108
6.1.2 Algorithm description 109
6.1.3 Feasibility test 110
6.1.4 Separation problems and algorithms 110
6.1.4.1 Separation of security inequalities 110
6.1.4.2 Separation of Path Covering inequalities . . . 111
6.1.4.3 Separation of Countermeasures Path inequalities 112
6.1.4.4 Separation of essential – by subsets removing – countermeasure inequalities 115
6.1.5 Implementation's features 117
6.1.6 Branching strategy 118
6.1.7 Primal heuristic 118
6.2 Computational study 119
6.2.1 Random instances 121
6.2.1.1 Description 121
6.2.1.2 F_p random instances 122
6.2.1.3 $F_{S,T}$ random instances 126

6.2.2 Realistic instances 129
6.3 Concluding remarks 133

7 Application in telecommunication industry 135
7.1 Internet of Things (IoT) 135
7.1.1 System description and risk assessment 135
7.1.2 IoT risk treatment 142
7.2 Software Defined Network (SDN) 145
7.3 Integration in a web application 146
7.4 Concluding remarks 151

Conclusion 152

Bibliography 170

Introduction

Context and motivation

In our increasingly digital world, cyberattacks are growing in prominence every day. In 2015, the International Data Group (IDG) [10] detected 38 percent more security incidents than the year prior. According to the Identity Theft Resource Center (ITRC) [5], in 2016 more than 29 million records were exposed to 858 violations. Recently, in January 2019 alone, 1.76 billion records were leaked [26]. The incidents caused by cyberattacks are various: denial of service attacks, website defacements, access to sensitive information, attacks on critical infrastructure, software and hardware failures, etc. These attacks damage several sectors including finance, government, health care and education.

The growth of cyberattacks is causing an increasing damage to enterprises, states and individuals. According to the Ponemon Institute's data breach study of cost [19], 383 organizations suffered at least one breach in 2016. The average cost per breach is 4 million of dollars. Cybercrime cost the global economy 500 billion in 2015. It will reach 2 trillion of dollars by the end of 2019, according to JUNIPER Research [14] and could cost companies 5.2 trillion of dollars over the next four years according to Accenture [1].

Security management has become a matter of urgency and it's not surprising that governments and businesses around the world are searching for better risk management strategies. For instance, the European Network and Information Security Agency [7] held a cyber security exercise since 2010, involving European countries and more than 200 organizations, including government bodies, telecommunication companies, energy suppliers, financial institutions and Internet service providers. Orange in particular is committed to ensure the security of its services and data by trying to propose innovative risk management approaches. For this, we need to address some serious challenges related to the security management procedure. In this thesis, we focus on the challenges induced by the evolution of telecommunication systems as well as economical issues of cyberdefense and their associated mathematical questions.

Before presenting these challenges, we define *the security risk management process*. Generally, there are two main steps in security risk management [116]: risk assessment and risk treatment. The risk assessment consists of three steps. It starts through the risk identification that allows to identify the vulnerabilities which, i.e., the weaknesses related to any valuable thing to be protected. The second step is risk analysis, which is the process of determining the induced risk for each vulnerability by evaluating the impact, i.e., the degree of losses, and the likelihood, i.e., the probability of occurrence. The risk evaluation step ends the risk assessment by giving an evaluation of the risk level using to the impact and the likelihood.

According to this evaluation and giving a risk threshold, decisions related to the protection strategy and its implementation need to be taken. In particular, it is important to determine the optimal protection strategy that handles the required level of risk while minimizing the deployment costs of *countermeasures*, which are the actions or devices that can prevent or mitigate the effects of an attack. This is the purpose of the final security management step which is the risk treatment. The security management process draws the attention of researchers who are proposing different methodologies in order to best keep the networks secure. However, they encounter serious challenges in each risk management step. For instance, the risk assessment challenges are closely related to the evolution of the telecommunication industry towards more complex systems. On the other hand, the risk treatment is rather associated to economical and mathematical challenges. We present these challenges in what follows.

Risk assessment challenges: It is impossible to dissociate the security risk management assessment and the evolution of today's telecommunication systems. The systems we aim at securing today are evolving towards increasingly complicated architectures. In the past, information systems were statically designed, and almost did not evolve during runtime, whereas current systems (virtual networks [119], clouds [131], service platforms through APIs, Software Defined Networks [29], etc.) evolve quite frequently. Systems are becoming dynamic by design. They rely on virtualization technologies used at different levels of the infrastructure. The virtualization includes the network (e.g., Network Function Virtualization [79]), the system (e.g., KVM, Xen and VMWare technologies [53]), and even the application layer (distributed data storage [55]). These systems are complex in the sense that: 1) they include a large number of heterogeneous elements; 2) these elements are connected by non-linear interactions, often of different types (e.g. physical and virtual links); 3) subject to external and insider inferences (e.g. attackers); and 4) the system evolves over the time (e.g., the evolution of the topology and the chance of exploiting vulnerabilities).

On the other hand, attackers are becoming more clever and troublesome

for such systems. They can use the network topology in order to propagate from an asset to another and exploit their associated vulnerabilities, which induces the notion of *risk propagation*. In addition, attackers can adapt to the evolution over the time of the vulnerabilities as well as the topology, which maintains the risk propagation over the time. Another factor that can significantly influence the risk propagation is the frequency of access between the system assets. In fact, the risk could be higher if the system assets are more frequently connected and vice versa. We refer to this frequency of access notion as *the accessibility*. Thus, in order to conduct efficient security management of modern telecommunication systems, one should consider the vulnerabilities, the network topology, the accessibilities and the way all of them evolve over the time. This is very challenging for the existing risk assessment methodologies

From our review of the literature, we have studied two well known risk assessment methodologies: *scoring methods*, [6, 31] and *graph-based models* such as attack graphs and dependency graphs [127]. Based on a common reference, scoring methods assign a score to each encountered vulnerability. However, one of their major limitation is the static and qualitative analysis of the vulnerabilities. In addition, scoring methods do not consider the notion of risk propagation, which is crucial to appropriately secure the systems. Consequently, these methods cannot be used on their own to assess the risks of modern telecommunication systems.

Graph-based models include elementary vulnerabilities that may be identified in a target system, and their relationships, in order to show how succession of elementary steps can potentially enable an attacker to gain privileges deep into the system. For this, the graph models offer to integrate the topology in the risk assessment process. From a risk propagation point of view, this is crucial since the topology description captures the causal relationships between the system assets which allows to take into account the propagation of attackers in the system. However, this is not sufficient to conduct fine grained risk assessment in modern telecommunication systems. In fact, to the best of our knowledge the accessibility between the assets in the system has never been considered in these graph-based methods. For this reason, graph-based models are not enough to conduct efficient risk assessment for today's systems.

In conclusion, we need to extend existing methodologies by proposing rigorous models of risk that take into account at the same time the vulnerabilities, the topology and the accessibility. Such models should consider the evolution of these factors over the time, and take into account all possible attackers as well as attack scenarios. The model also must allow us to accurately evaluate the system risk in order to efficiently supervise its security. In what follows, we discuss the economical and mathematical challenges associated to risk treatment.

Risk treatment challenges: From an economical point of view, two major axes have got security community attention's: the cost of cybercrimes and the one of cyberdefense. In this thesis we focus on cyberdefense costs. According to market research firm Gartner [9], global spending on cybersecurity is floating around 80 billion of dollars in 2016. By 2020, companies around the world are expected to spend around 170 billion of dollars, which is a growth rate of nearly 10 percent compared to 2015. Consequently, it is essential to take into account the cyberdefense costs while dealing with risk treatment.

Actually, using the system risks evaluated during the risk assessment step and some security requirements giving by a system risk threshold for example, one can identify whether the system is secured or not. If the risk exceeds the threshold, an alert could be sent to start risk treatment which can consist in the deployment of countermeasures on some system assets in order to reduce the global risk. However, while a countermeasure could reduce the system risks, its deployment might be expensive. Hence, it is very important to find the countermeasure locations in the system that guarantee the security in a certain sense, at minimal cost. This question can be mathematically formulated as *a combinatorial optimization problem* [122]. Succinctly, a combinatorial optimization problem is the problem of finding the best solution from all feasible solutions giving by some constraints. In this context, the set of feasible solutions are nothing but all possible countermeasure locations that satisfy the security requirements (the constraints), and the best solution is the one with minimal cost.

One can see this optimization problem as a "game" between a defender and several attackers. Attackers try to find their most easy path to access their target. But they are forced to act according to a certain hierarchy. In fact, the defender will select the countermeasures placement in order to secure the system by making the propagation more difficult for attackers. To do so at minimal cost, the defender will anticipate all the reactions of the attackers to its decisions. From an optimization point of view, this is a bilevel programming problem [51]. That is an optimization problem (the leader) having other parametric optimization problems (the followers) as part of its constraints. This leads us to important mathematical questions. In fact, we must choose the bilevel programming model that permits to ensure the highest level of security at minimal cost. Then, we need to design innovative algorithms in order to efficiently solve the bilevel model in a reasonable time. For this, a rigorous theoretical study of the bilevel optimization problem is essential. In fact, such study can strengthen the problem resolution from an algorithmic point of view.

One of the most popular new bilevel programming topics to solve security problems are *Shortest Path Network Interdiction Problems* (SPNIPs) [86] which can be defined as follows. Given a graph [46] $G = (V, A)$ where V is the set of nodes and A is the set of arcs, a source s in V (the attacker), a

target t in V, and a positive weight associated to each arc, the SPNIP consists in maximizing the shortest s-t path length, in order to make the propagation more difficult for the attacker, either by interdicting arcs [92] or by interdicting nodes [36, 47]. These problems still present some limitations that need to be addressed. In fact, by removing a node, we completely remove its associated risk as well. Hence, SPNIPs do not permit us to consider realistic countermeasures allowing to reduce the effect of a node risk without completely eliminating it.

Consequently, we need a more general bilevel model that improves SPNIPs by considering realistic countermeasures as well as multiple sources and targets. In other words, instead of removing a given node, the leader can pay a given price to increase the length of the ongoing arcs of that node, in order to make it more difficult for an attacker to gain access to that node. Moreover, if increasing the length of the ongoing arcs of a given node leads to a very large value, then the interdiction is equivalent to removing the node and the problem reduces to the classical SPNIP [47].

Contributions

In this thesis, we first propose a new risk assessment approach that addresses the modern telecommunication systems' challenges. Based on graph theory [46, 137], we introduce the concept of *Risk Assessment Graphs* (RAGs). A node in the RAG is either *an access point*, i.e., the start point of an attacker, or *an asset-vulnerability* node to be secured. We introduce the potentiality and the accessibility as essential metrics for the definition of the RAGs. Both of them are functions of time and indicate respectively the probability of exploiting a node in the RAG, and the frequency of access between the nodes. An arc in the RAG represents a potential propagation of an attacker from a node to another. To each arc is associated a positive weight representing the *propagation difficulty* of an attacker which is obtained from a combination of the potentiality and the accessibility. The graphs we propose takes into account at the same time the vulnerabilities, the system topology, the accessibility and the way all of these evolve over the time. They allow to analyse the systems by capturing the topological accessibilities as well as security information in terms of vulnerabilities and their potentialities. They take into account not only the current system state, but also the way it evolves throughout a time horizon. In addition, all possible attackers and attack scenarios are explicitly considered as *paths* in the RAGs.

In addition, we propose a new quantitative risk evaluation approach that uses the RAGs to compute innovative security metrics namely *the propagation difficulty* , *the propagated risk* , *the node risk* and *the global risk*. The basic and most important security metric is the propagation difficulty which is defined,

with respect to an access point s, an asset-vulnerability node t and a path P from s to t, as the sum of the arc weights of P. As its name implies, the propagation difficulty indicates how it is difficult for an attacker to propagate on P. The shortest $s-t$ path which is the path of minimum propagation difficulty is considered as *the most likely path* for an attacker. From a protection point of view, a path is said to be *secured* if its propagation difficulty is greater than a given propagation difficulty threshold. A system will be considered as *secured* when all the paths from each access point to each asset-vulnerability node are so. The efficiency of our propagation difficulty metric consists in reducing the verification of the whole system security to the one of the most likely paths. In fact, the propagation difficulty of the most likely $s-t$ path is the minimum over all the $s-t$ paths. Hence, if the most likely $s-t$ path is secured, then so it is for all $s-t$ paths. Moreover, we illustrate the risk assessment approach in a SDN use case and we conduct numerical simulations to show the sensitivity of our metrics to the potentiality, the topology and the accessibility changes.

The optimization problem we consider is called the Proactive Countermeasure Selection Problem (PCSP) and can be defined as follows. Given a RAG as previously stated; a propagation difficulty thresholds for each access point s and each asset vulnerability node t; and a set of countermeasures that can be placed on the asset-vulnerability nodes with a given installation cost and a given effect, the PCSP consists in selecting a subset of countermeasures at minimum cost such that the *security constraints* are respected: the length of each shortest $s-t$ path is greater than or equal to the $s-t$ propagation difficulty threshold. We show that the PCSP is NP-Complete even when the RAG is reduced to an edge. We then formulate the problem as a bilevel programming model in which the leader plays the role of the defender and the follower plays the role of attackers. We use primal-dual optimality conditions to convert the bilevel model into a compact single level formulation that is directly solved using the ILP solver Cplex [2]. We also give a second formulation by projecting the compact formulation on a subset of variables. This induces *a path formulation* with an exponential number of constraints called *basic inequalities*, and will be solved to optimality using a Branch-and-Cut algorithm. Moreover, we study the optimality conditions for the PCSP which permits to identify some inequalities that are verified by any optimal solution of the problem. This can be used during the preprocessing in order to improve the algorithmic aspect.

A very efficient method that can also significantly strengthen the algorithmic aspect is the polyhedral approach [101]. Our further contribution consists in a polyhedral investigation for the path formulation. We characterize the dimension of the polytope by considering *the essential countermeasures*, i.e., the countermeasures such that if we remove at least one of them, the PCSP does not have a solution. We then introduce several classes of valid inequalities and discuss when these inequalities define facets. This investigation will give

a good base for the algorithmic study.

Moreover, we use the polyhedral results within a Branch-and-Cut algorithm for the path formulation. We develop a preprocessing phase considering the essential countermeasures equations and the optimality condition inequalities. We devise separation routines for the basic and valid inequalities. We also propose a primal heuristic to enable a fast pruning of uninteresting branches of the tree and hence accelerate the Branch-and-Cut algorithm. Furthermore, we present numerical tests of the compact formulation and the path formulation. The aim of the computational study is to examine, from an algorithmic point of view, the efficiency of the polyhedral results. For this, we investigate the impact of the optimality condition inequalities and valid inequalities in the resolution of the problem. Moreover, we study the sensitivity of our algorithm to the density of the graph and the number of nodes. The tests are executed on random and realistic instances.

Our work achieves both flexibility and generality requirements, and applies to a wide set of applications. In this thesis, we illustrate the complete risk management approach we develop in several practical use cases including Internet of Things (IoT) [35] and Software Defined Network (SDN). We also show the integration of our framework in a web application that we have developed and illustrate in a Local Network Areas (LANs) use case. This web application covers the complete risk management framework we propose in this thesis including risk assessment and risk treatment, along with a visualization interface.

Manuscript organization

This thesis is organized as follows.

In Chapter 1, we present the basic elements of security risk management process. We describe the advances beyond the state of the art about the first part of this process which is risk assessment. We describe some modern telecommunication networks such as SDN, 5G and IoT and present the security challenges that are related to them.

In Chapter 2, we introduce the mathematical background that is used in order to develop our security risk treatment approach. In particular, we give some notations that will be used throughout the manuscript and present basic notions of graph theory, combinatorial optimization, complexity theory, polyhedral approaches, and bilevel programming. We also discuss the state of the art related to the bilevel Shortest Path Network Interdiction Problem (SPNIPs).

In Chapter 3, we present our RAGs-based security risk assessment methodology as well as our risk evaluation approach. A SDN use case is studied and further simulation results are discussed.

In Chapter 4, we present the PCSP problem and study its complexity. We present a bilevel formulation of the problem and provide two single-level reformulations. We also study the optimality conditions for the PCSP.

In Chapter 5, we study the polytope associated with the PCSP path formulation. We characterize its dimension and study the facial aspect of its basic inequalities as well as further valid inequalities.

Based on these results, we devise in Chapter 6, a Branch-and-Cut algorithm. We describe the separation routines and present substantial computational results.

Chapter 7 is devoted to some real applications to telecommunication industry. That is an IoT a SDN use cases. We also show the integration of our framework in a web application.

We give at the end some concluding remarks and perspectives.

Chapter 1

Security risk management and modern telecommunication systems

Contents

1.1 Security risk assessment and management 10

1.1.1 Basic concepts 10

1.1.2 Risk assessment state of the art 12

1.2 Modern telecommunication systems and associated security challenges 15

1.2.1 Software Defined Networks (SDN) 15

1.2.2 Next generation 5G networks 17

1.2.3 Internet of Things (IoT) 18

1.3 Concluding remarks 19

In this Chapter, we provide the basic notions on security risk management process and present the advances beyond the state of the art related to the first step of this process which is security risk assessment. Knowing that a good security management of a given telecommunication system requires a strong knowledge of the latter, we will also focus on modern telecommunication systems which are more and more complex and dynamic. In particular, we describe characteristics and operating principle of Software Defined Networks (SDN), Internet of Things (IoT) and 5G systems, and present some security challenges for each of them.

1.1 Security risk assessment and management

In this section, we review some general security risk analysis and management notions presented in [88, 116, 118, 130, 125, 129, 114]. We then discuss related works that correspond to the risk assessment step and highlight the advances beyond the state of the art.

1.1.1 Basic concepts

In order to discuss risk analysis and management notions, we first need clear definitions of the related terms that will be used throughout the manuscript.

- *An asset* is any valuable thing to be protected (e.g. server, Virtual Machine (VM), router, human, financial account, etc.).
- *A system* is a set of assets that can virtually or be connected.
- *A threat* is a potential action with the propensity to cause damage by exploiting an asset.
- *A vulnerability* is defined as an inherent weakness in an asset that can be exploited, such as an open port connected to the internet or the use of unpatched software [125]. If there is no vulnerability, there is no concern to threat activity.
- *The impact* is the degree of losses as a result of threat activity. In fact, the threat activity generates a negative effect such as compromised security, lost time, diminished quality, lost money, lost control, lost understanding, etc.
- *The likelihood* is the probability of occurrence associated to each threat activity.
- *The risk* is defined as a potential problem that the system or its users may encounter. Risk is characterized by its likelihood and its impact.
- *A countermeasure* is an action, process or device that can prevent or mitigate the effects of threats to an asset. Countermeasures can take the form of software, hardware and modes of behaviour. Software countermeasures include application firewalls, anti-virus software, pop-up blockers, spyware detection/removal programs. The most common hardware countermeasure is a router that can prevent the IP address of an individual computer from being directly visible on the Internet. Other hardware countermeasures include biometric authentication systems, physical restriction of access to computers, intrusion detectors and alarms.

Behavioural countermeasures include regular scanning for viruses and other malwares, regular installation of updates and patches for operating systems, refusing to click on links that appear within e-mail messages, refraining from opening e-mail messages and attachments from unknown senders, staying away from questionable Web sites, regularly backing up data on external media, etc.

While security community is working on risk management in many forms, they all follow the same goals: to provide decisions on whether risks are acceptable and obtain reliable information on how we can deal with them, if necessary. The risk management process includes two major steps which are *risk assessment* and *risk treatment* as represented in Figure 1.1.1

Risk assessment consists of three steps:

- *Risk identification:* the process of determining what could occur, how and when.
- *Risk analysis:* the process of determining the impact of each risk as well as the likelihood of those consequences. The result could be expressed as a qualitative, semi-quantitative, or quantitative form.
- *Risk evaluation:* about giving an evaluation of the level of risk according to the impact and the likelihood. Therefore, decisions on whether the risk is acceptable or not will be taken.

Risk treatment is the process of making decisions about protection strategy and their implementation. It involves selection from a set of countermeasures and includes cost analysis. In general, there are three strategies for treating risk:

- *Avoiding the risk:* changing requirements for security or other system characteristics,
- transferring the risk: allocating the risk to other systems, people, organizations, or assets; or buying insurance to cover any financial loss should the risk become a reality,
- assuming the risk: accepting it, controlling it with available resources, and preparing it to deal with the loss if it occurs.

These strategies are part of the security risk treatment helping to establish a good security posture. This is the process of implementing and maintaining countermeasures that reduces the effects of risk to an acceptable level.

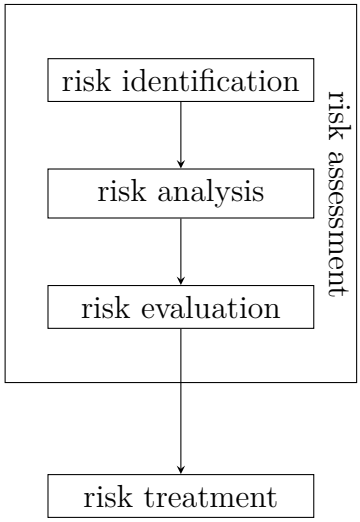


Figure 1.1: The risk management process

1.1.2 Risk assessment state of the art

The current literatures on risk assessment include multiple contributions to evaluate and quantify risks in Information and Communications Technologies (ICT) systems. We study two well-known categories: scoring methods and graph-based methods.

1.1.2.1 Scoring methods

International standard organizations, such as the National Vulnerability Database (NVD) [17], have provided many risk scoring methods which assign a numerical value, in terms of a score, to each encountered vulnerability, based on a common reference. Common Vulnerabilities and Exposures (CVE) [3] is a list of vulnerabilities containing an identification number, a description, and at least one public reference for each publicly known vulnerabilities. CVE vulnerabilities are used in numerous security products and services around the world, including the NVD. In Table 1.1, we describe the most common known vulnerabilities.

Common Vulnerability Scoring System (CVSS) [106] has become a widely accepted industry standard for risk measurements. It considers both specific characterization of vulnerabilities, such as the impact and the likelihood. These two factors may be put into categories of “high", “medium", or “low" without any objective model. In addition, these methods are often used to evaluate only the risk of individual events without evaluating the propagation of risk within the system. This is not sufficient to construct a representative

security model. Furthermore, the score of a vulnerability is not estimated at different times. This makes risk evaluation over the time a difficult question. Consequently, these static and non contextual risk analysis can not give efficient protection strategies with dynamic systems.

While scoring methods provide a common base and reference to share information about vulnerabilities and their impact and likelihood, they cannot be used as a standalone metric to leverage risks in a target real-world system. Therefore, current approaches in the literature usually compose elementary vulnerabilities that may be identified in a target system, and their relationships, through a graph-based model in order to show how succession of elementary steps can potentially enable an attacker to gain privileges deep into the system. These graphs enable up to a certain level to leverage the context through which a given vulnerability affects a system. Nonetheless, current graph models are still suffering from some limitations.

1.1.2.2 Graph-based methods

Attack graphs [115, 111, 32, 83, 134, 139, 105, 94, 138, 97, 82, 81] are used to assess the risks associated with system vulnerabilities. This kind of graphs highlight the cumulative effect of attack steps. Each path in the attack graph carry on an undesirable state, (e.g. gaining administrator access to a data base). Many parameters can be used. The probability and the cost of each transition between the states can be added, and many other information can be used to construct the attack graph.

The most related approaches to our work are [136], [82] and [120] . In [136], authors use attack graphs and the Hidden Markov Model to explore the probabilistic structure of actual states. This is based on a middle-ware approach using dependency attack graph representing network assets and vulnerabilities. The parameters used to construct these attack graphs are the network assets and vulnerabilities from the NVD. However, Wang et al. system [136] lacks the ability to generate the graphs and incorporate the results into the system. The attack graphs are created manually and not automatically generated. Furthermore, the topological information of the network is missed. The scope of our work is different from this point of view, since and the topological context in which the vulnerability appears is considered while generating the RAGs.

In Hong and Kim [82], both vulnerability information and the topological characteristics of the system are considered by proposing a two-layer graph model. The topological layer can contain cycles depending on the network structure. The vulnerability layer has a directed tree structure leading to the target of the intruder. However, no implementation of the method is developed. The topological information considered in this work are still static and

do not consider the accessibility as a factor of risk. Our work is drastically different since our model is adaptable to the target system accessibility changes. In fact, we consider the topology as well as the accessibility metric as a function of time indicating the frequency of connection between the assets of the system.

Finally, authors in [120] address the problem of simultaneous attacks by presenting a new formal description of individual, coordinated, and concurrent attacks. The generation of simultaneous attacks is based on set theory and graph theory. The graphs are automatically generated using a logical approach based on Situation Calculus [133]. This is a dialect of first-order logic with second order-logic terms for representing dynamic change. It basically consists of situations, predicates and actions. Nevertheless, in this work, the risk inferred by simultaneous attacks on a system network is not evaluated. Our approach is different since the risk inferred by intruder threats on a system can be evaluated. Our graph-based model accurately reflects the context within vulnerabilities appear by covering all intruders, system assets, as well as their respective interactions in the system.

Dependency graphs are yet another tool for risk assessment [110, 93, 87]. Such graphs represent the way the system assets interact with each other. For example, Kheir et al. [93] propose a dependency graph to evaluate the Confidentiality, Integrity and the Availability (CIA) impacts. However, this approach leverage only attack impacts, but not their potentiality. Therefore, it can be used only for intrusion response against ongoing attacks, but it cannot be used as a standalone mechanism for dynamic risk management or to balance between risks and featured reaction strategies.

In practice, the risk assessment process depends on the intuition of the security expert dealing with it. This gives a qualitative risk metrics whose indications are not always efficient. In the first part of Chapter 3, we propose a mathematical framework for risk assessment, based on rigorous tools that relay on graph theory and give efficient security metrics. The framework we propose extends related work by providing a new risk assessment framework that takes into account at the same time the vulnerabilities, the system topology, the accessibility and the way all of these evolve over the time. We will propose the concept of *Risk Assessment Graphs (RAGs)* as a tool for risk analysis. These graphs allow analysing the complex systems by capturing both the topological accessibility features of the target system, and security information in terms of vulnerabilities as well as their causal relationships. They take into account not only the current system state, but also the way it evolves throughout a period of time. In addition, all possible attackers and attack scenarios are explicitly considered as *paths* in the RAGs.

Evaluation of system risk is an essential step to secure any system. A. Atzeni et al. discuss the importance of security metrics in [34]. Different

works have been proposed, for example, in [27], the authors present a method of calculating a policy security score that combines two measures - the existing vulnerability and the historical vulnerability. In [30] vulnerabilities future predictions and a risk propagation metric have been incorporated to enrich the previous work. However, to the best of our knowledge, none of them has gave a quantitative risk evaluation approach by developing algorithms using a tool such as the RAGs to give innovative security metrics taking into account at the same time the vulnerabilities, the system topology and the evolution over the time. This is the purpose of the second part of Chapter 3.

Remark 1.1 As the attack graph models, RAGs could be used to assess the vulnerabilities of systems. But, these tools are different. In fact, attack graphs consist in modelling the systems and their behaviour by using nondeterministic Büchi automaton [124], and the nodes correspond to the system states. However, RAGs model is based on graph theory [46, 137], and a node in the RAG corresponds to an asset-vulnerability combination. Another difference is that the system state in our work is represented by the whole RAG and not a node as it is the case with attack graphs, which gives more security factors details. In addition, several RAGs are constructed for each time slot in order to capture the change of the state over the time.

1.2 Modern telecommunication systems and associated security challenges

1.2.1 Software Defined Networks (SDN)

Communications are transported in the form of digital packets thanks to telecommunication devices; routers and switches. *The control plane* that decides how to manage network traffic), and *the data plane* that forwards traffic according to the decisions made by the control plane are implemented inside network devices. That's why traditional networks are hard to manage [45]. To change network policies for example, network operators need to configure each individual network device separately. In addition to that configuration difficulty, traditional networks can not adapt to the dynamics of faults and load changes since the automatic reconfiguration and response mechanisms are virtually non-existent.

By decoupling the control plane from the data forwarding, Software Defined Networks [95, 103] change the limitations of traditional network infrastructures. Network switches and routers become simple forwarding devices and the control policies are implemented in a virtual centralized controller (or

network operating system), simplifying policy enforcement and network reconfiguration and evolution. A simplified representation of SDN architecture is given in Figure 1.2.

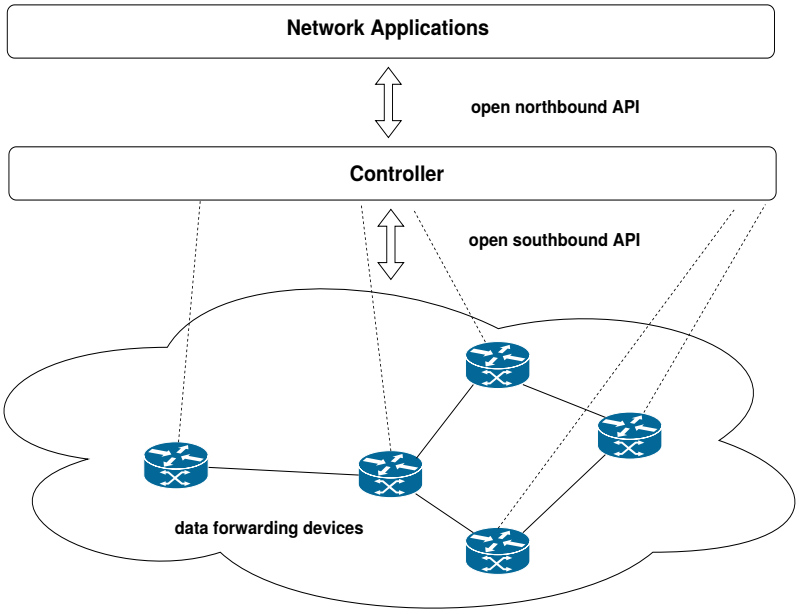


Figure 1.2: Simplified SDN architecture

The separation of the control plane from the data plane can be realized by a programming interface between the switches and the SDN controller. The controller directly controls the state in the data plane devices via this application programming interface (API), as represented in Figure 1.2. The most known example of such an API is OpenFlow [104]. The controller allows OpenFlow to perform certain actions (dropping, forwarding, modifying, etc.) on the traffic. Depending on the rules installed by a controller application, an OpenFlow switch can behave like a router, switch, firewall, or perform other roles (e.g., load balancer, traffic shaper, etc.).

From a security point of view, the SDN separation of the control and the data planes is a double-edged sword [123]. The controller offers the advantages of security services insertion and security policies alternation, rather than the hardware replacement. As a logically centralized entity, it also maintains a global view of the network, and mitigates then the risk of policy collision. However, since the controller is responsible for managing the entire network, when a switch encounters a packet with no forwarding rules, it passes the data to the controller for decision. Consequently, an attacker may send data through a SDN switch to exploit a vulnerability on the controller. As a result,

a security flaw of the controller can compromise the whole network which represents a big challenge.

1.2.2 Next generation 5G networks

Mobile devices are essential in our daily lives. The mobile network infrastructure connecting them has become critical. It will take an important role with the next-generation 5G mobile systems [68, 28, 78, 109, 73] which is supposed to be a source of huge number of services and devices in order to meet the drastic subscriber demands in near future.

Even 5G systems are yet to be determined, it is clear that it would be a convergence of two complementary axis that are orienting the research and industrial activity on 5G. One is focusing on scaling up and enhancing the efficiency of mobile networks (e.g., 1000xtraffic volume, 100x devices, and 100x throughput). Radio access is considered as the major research that focus around this view by investigating new technologies and spectrum bands (e.g.,massive MIMO, millimeter waves [117]).

The other axis is service-oriented and lead 5G systems to a wide range of services having different requirements and types of devices. This axis adds various types of machine-type communications to the conventional human-type communications. Consequently and depending on the service in question, the network takes different forms, which implies the notion of slicing [140] the network on a per-service basis.

Security is a facet of principle importance in cellular networks and in 5G in particular. In fact, the more the number of users, the bigger is the possibility of attacks [84, 102, 108]. Several vulnerability categories have been identified by 3GPP security workgroup (3rd Generation Partnership Project 3GPP, 2007), which are open research problems in this field. A a major security concerns associated to 5G is energy efficiency [69]. In fact, it is easy for an attacker to forge the energy state of a device since the energy cannot be encrypted. This can imply loss of critical information, to an unauthorized user in the network.

Choosing appropriate security measures is crucial, for 5G networks. Most of the security issues are mitigated by cryptographic solutions. Many organizations have taken considerable actions in order to secure 5G networks. ETSI [8] and IETF [11] are involved in the standardization of 5G security. They aim at defining new trust models for securing 5G systems. ISO [12] is also involved in the standardization of 5G networks, as they are expected to play an important role in security.

1.2.3 Internet of Things (IoT)

The IoT [35] is a new paradigm that will soon be everywhere. Several organization such as IBM [48] and Gartner [132] predict that there will be billions of connected objects in the future. IoT is a system of connected computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without human-to-human or human-to-computer interaction. The basic idea of this concept is the presence around us of a variety of things or objects – such as Radio-Frequency IDentification (RFID) tags, sensors, actuators, mobile phones, etc. which are able to interact with each other and cooperate with their neighbours to reach common goals.

The IoT structure is generally divided into three layers, including perception layer, network layer, and application layer [141]. The perception layer control transmission and collect information equipments such as RFID, zigbee, all kinds of sensors. The network layer consists of the transmission system such as mobile communication network and internet. Finally, application layer includes cloud computing, some analytical services, intelligent transportation and smart homes.

Several security issues of IoT are associated with each of the three-layer system structure. The main equipment in perception layer includes RFID [89], zigbee [66], all kinds of sensors. The attackers can easily gain access to the equipment,control or physically damage them. For instance, DPA (DifferentialPower Analysis) is a very effective attack. Several types of attackers such as node capture, fake node and malicious data, denial of service attack, timing attack, SCA (Side Channel Attack) can damage the perception layer [96].

General security problems of communication network can damage the network layer and will threat to data confidentiality and integrity. Although the existing communication network has a relatively complete security protection measures, there are still some common threats, including illegal access networks, dropping information, confidentiality damage, integrity damage, DoS attack, Man-in-the-middle attack, virus invasion, exploit attacks, etc.

The security issues of application layer are different. They include the data access permission due to the large number of users for each application, the data protection and recovery, the ability of dealing with mass data because of the huge amount of data transmission which can lead to data, and the application layer software vulnerabilities

As a network of networks, IoT security involves various different layers. A lot of security measures applied to each independent networks have been proposed. In particular, the mobile communication network and the Internet network security research have a long time. For sensor networks in IoT, the

diversity of resource and the network heterogeneity make security research much more difficult. More details in security measures for IoT security can be found in [141].

1.3 Concluding remarks

In this chapter, we have presented the basic elements of security risk management process. We have also introduced the advances beyond the state of the art about the first part of this process which is risk assessment. We have then described some modern telecommunication networks such as SDN, 5G and IoT and presented the security challenges that are related to them. In order to have all the bases needed to accomplish the whole security management process, in the next chapter we introduce the mathematical background that is used in order to develop our security risk treatment approach, which is the final step of any security management method.

Table 1.1: Vulnerability categories

Name	Description
Denial of Service	An attack meant to shut down a machine or network, making it inaccessible to its intended users.
Execute Code	An attacker execute arbitrary commands or code on a target machine or in a target process
Overflow	An attacker make a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations
XSS	It is a client-side code injection attack. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application.
Directory (Path) Traversal	An HTTP attack which allows attackers to access restricted directories and execute commands outside of the web server's root directory.
Bypass Something	An attacker circumvent security mechanisms to get system or network access, The point of entry is through a mechanism that enables the user to access the system without going through the security clearance procedures such as authentication.
Gain Information	A attacker exploits systems security procedures, administrative controls or Internet controls to gain unauthorized access to information
Gain privilege	A type of network intrusion that takes advantage of programming errors or design flaws to grant the attacker elevated access to the network and its associated data and applications.
Sql Injection	An attacker manipulate the query itself and force it to return different data than what it was supposed to return.
File Inclusion	An attacker include a file, usually exploiting a "dynamic file inclusion" mechanisms implemented in the target application. The vulnerability occurs due to the use of user-supplied input without proper validation.
Memory Corruption	A attacker modify the contents of a memory location using a programmatic behavior that exceeds the intention of the original programmer or program/language constructs; this is termed violating memory safety.
Cross-Site Request Forgery (CSRF)	An attack forces an end user to execute unwanted actions on a web application in which they're currently authenticated.
Http Response Splitting	Occurs when an attacker injects data in a web application through an untrusted source, most frequently an HTTP request, or when the data is included in an HTTP response header sent to a web user without being validated for malicious characters.

Chapter 2

Mathematical background

Contents

2.1 Graph theory: definitions and notations 22

2.1.1 Undirected graphs 22

2.1.2 Directed graphs 24

2.1.3 Graph classes 26

2.2 Combinatorial optimization 28

2.3 Algorithmic and complexity theory 29

2.4 Polyhedral approach and Branch-and-Cut 30

2.4.1 Elements of the polyhedral theory 30

2.4.2 Cutting plane method 34

2.4.3 Branch-and-Cut algorithm 35

2.4.4 Primal heuristics 37

2.5 Bilevel Programming 38

2.5.1 History: Stackelberg games 38

2.5.2 Generality 39

2.5.3 State of the art: Shortest Path Network Interdiction Problems (SPNIPs) 43

2.6 Concluding remarks 44

This chapter is devoted to the mathematical and algorithmic background used to develop our risk treatment approach. We give some basic definitions and notations related to graph theory that is used throughout the manuscript. We then present the basic elements of combinatorial optimization and complexity theory. Next, we introduce polyhedral approaches and explain in particular the principles of cutting planes and branch-and-cut method to solve

optimization problems to optimality. We end this chapter with bilevel programming. We discuss particularly a class of bilevel programming problems related to ours called Shortest Path Network Interdiction Problem (SPNIPs).

2.1 Graph theory: definitions and notations

In this section, we present some basic definitions and notations of graph theory that will be necessary for the subsequent chapters. Also, we present the theory of random graphs [59]. There are two types of graphs, either directed or undirected.

2.1.1 Undirected graphs

An undirected graph is denoted $G = (V, E)$ where V is the set of vertices or nodes and E is the set of edges. If e is an edge between two vertices u and v, then u and v are called the ends of E, and we write $e = uv$ or $e = (u, v)$. If u is an extremity of e, then u (resp. e) is said to be incident to e (resp. u). Similarly, two vertices u and v forming an edge are said to be adjacent. Since the graph G may have multiple edges, it may be that $e = uv$ and $f = uv$ but $e \neq f$.

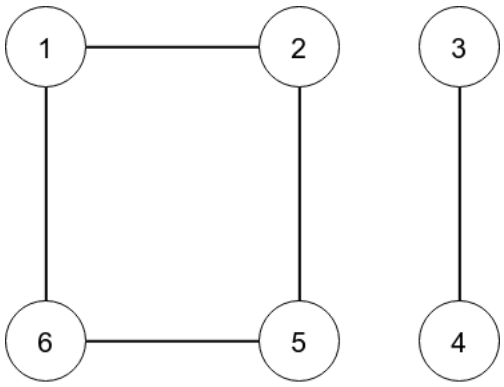


Figure 2.1: An undirected graph G

If $F \subseteq E$ is a subset of edges, then $V(F)$ represents the node set of edges of F. If $W \subseteq V$ is a subset of vertices, then $E(W)$ denotes the set of edges having their two ends in W. Let $V(H)$ and $E(H)$ be the sets U and F, respectively.

A subgraph $H = (U, F)$ of G is a graph such that $U \subseteq V$ and $F \subset E$. A subgraph $H = (U, F)$ of G is called *covering* or *spanning* if $U = V$. Let

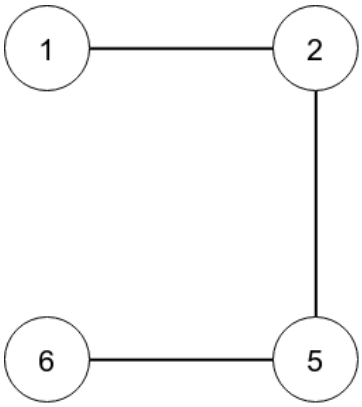


Figure 2.2: Subgraph H_1 of G

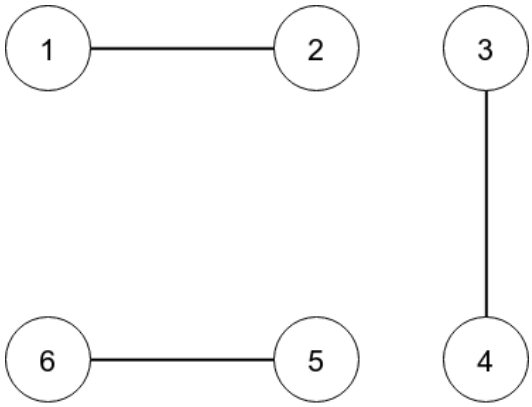


Figure 2.3: Spanning subgraph H_2 of G

$W \subseteq V$, $H = (W, E(W))$ is said to be *subgraph* of G induced by W and will be denoted by $G[W]$.

If $F \subset E$ (resp. $W \subset V$), it is noted in $G \setminus F$ (resp. $G \setminus W$) the graph obtained from G by removing the edges of F (resp. nodes of W and the edges incident to W). If F (resp. W) is reduced to a single edge e (resp. a single vertex v), we write $G \setminus e$ (resp. $G \setminus v$). Let $W \subseteq V$, $\emptyset \neq W \neq V$, a subset of vertices of V. The set of edges having one end in W and the other in $V \setminus W$ is called *cut* and noted $\delta(W)$. By setting $\overline{W} = V \setminus W$, we have that $\delta(W) = \delta(\overline{W})$. If W is reduced to a single vertex v, we write $\delta(v)$. The cardinality of the cut $\delta(W)$ of a subset W is called the degree of W and noted $d(W)$. Given W and W' two disjoint subsets of V, then $[W, W']$ represents the set of edges of G which have one end in W and the other in W'.

An edge $e = v_1v_2 \in E$ is called a *cut edge* if G is connected and $G \setminus e$ is not connected, with $v_1, v_2 \in V$.

If $\{V_1, \ldots, V_p\}, p \geq 2$, is a partition of V, then $\delta(V_1, \ldots, V_p)$ is the set of edges having one end in V_i and the other one in V_j and $i \neq j$.

The *support graph* of an inequality is the graph induced by the vertices of variables having a non-zero coefficient in the inequality.

Let $G = (V \cup T, E)$ be a graph defined by a set of vertices $V \cup T$ where T is a set of distinguished nodes and E is a set of edges. We denote by $V(H)$, $T(H)$ and $E(H)$ its sets of nodes, terminals and edges, respectively. We denote by $t(G)$ the number of terminal in G, *i.e.*, $|T(G)| = t(G)$.

A *path* P is a set of p distinct vertices $v_1, v_2, \ldots, v_p$ such that for all $i \in \{1, \ldots, p-1\}$, v_iv_{i+1} is an edge. P is called *elementary* if it passes more than once by the same node (except for v_0 and v_k if they represent the same vertex in G). A basic chain is totally identified with its set of edges.

Two paths between two nodes u and v are called edge-disjoint (resp. node-disjoint) if there is no edge (resp. no node different from of u and v) appearing in both chains.

Vertices $v_2, \ldots, v_{p-1}$ are called *the internal vertices* of P. Given a path P between two terminals $t, t' \in T$ such that $P \cap T = \{t, t'\}$, the set of internal vertices of P will be called *a terminal path* and denoted by $P_{tt'}$. A terminal path is *minimal* if it does not strictly contain a terminal path.

Given a graph $G = (V \cup T, E)$ and two subgraphs $G_1 = (V_1 \cup T_1, E_1)$, $G_2 = (V_2 \cup T_2, E_2)$ of G. Graph G_1 is said to be *completely included* in G_2, if $V_1 \cup T_1 \subseteq V_2 \cup T_2$.

2.1.2 Directed graphs

A directed graph is denoted $G = (V, A)$ where V is the set of nodes and A the set of arcs.

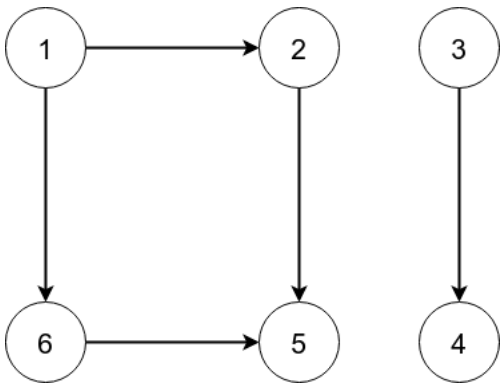


Figure 2.4: A directed graph G

If $a \in A$ is an arc connecting a vertex u to vertex v, then u will be called initial end and v final end and we write $a = (u, v)$. We say that a is an outgoing arc of u and v of an incoming arc. The vertices u and v are called ends of a. Vertex v (resp. a) is said to be incident to a (resp. v) if v is an end (initial or final) of a.

If $B \subseteq A$ is a subset of arcs, then $V(B)$ represents the node set of arcs of B. If $W \subseteq V$ is a subset of vertices, $A(W)$ is the set of arcs having their ends in W.

A subgraph $H = (U, F)$ of G is a graph such that $U \subseteq V$ and $F \subset A$. A subgraph $H = (U, F)$ of G is said covering if $U = V$.

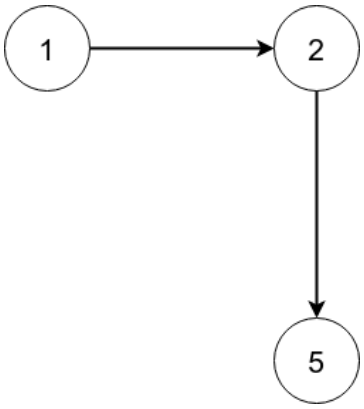


Figure 2.5: Directed Subgraph H_3 of G

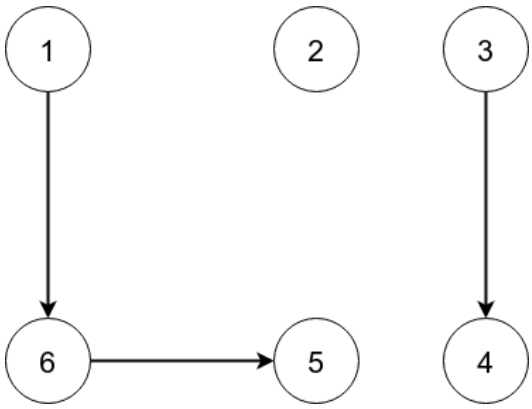


Figure 2.6: Covering directed subgraph H_4 of G

If $F \subset A$ (resp. $W \subset V$), we denote by $G \setminus F$ (resp. $G \setminus W$) the graph obtained from G by removing the F arcs (resp. node of W and edges incident to W). If F (resp. W) is reduced to a single arc a (resp. a single vertex v), we write $G \setminus a$ (resp. $G \setminus v$).

Let $W \subseteq V$, $\emptyset \neq W \neq V$, a subset of vertices V. The set of arcs having their initial end in W and their final nodes in $V \setminus W$ is called outgoing cut and denoted $\delta^+(W)$. The cardinality of the outgoing cut $\delta^+(W)$ of a subset W is called outgoing degree of W and denoted $d^+(W)$. If $u \in W$ and $v \in V \setminus W$, then the outgoing cut is also called uv-outgoing cut. If W is reduced to a single vertex v, we write respectively $\delta^+(v)$ and $d^+(v)$ instead of $\delta^+(\{v\})$ and $d^+(\{v\})$. The set of arcs having the final end in W and the initial end in $V \setminus W$ is called incoming cut and denoted $\delta^-(W)$. The cardinality of the incoming cut $\delta^-(W)$ of a subset W is called incoming degree of W and denoted $d^-(W)$. If $u \in W$ and $v \in V \setminus W$, then the incoming cut is also known as uv-incoming cut. If W is reduced to a single vertex v, we write respectively $\delta^-(v)$ and $d^-(v)$ instead of $\delta^-(\{v\})$ and $d^-(\{v\})$.

The cut of a set $W \subseteq V, \emptyset \neq W \neq V$, is denoted $\delta(W)$ and is the union of the arcs of the incoming cut and outgoing cut, i.e., $\delta(W) = \delta^+(W) \cup \delta^-(W)$. The cardinality of the cut is called the degree of W and denoted $d(W)$. If $u \in W$ and $v \in V \setminus W$, then the cut is also called uv-cut. If W is reduced to a single vertex v, we write respectively $\delta(v)$ and $d(v)$ instead of $\delta(\{v\})$ and $d(\{v\})$. If all W associated with the outgoing cut $\delta^+(W)$ contains the vertex u but not the vertex v, then we call it uv-outgoing cut.

Given disjoint subsets $W_1, W_2, \ldots, W_k$ of V, then $[W_1, W_2, \ldots, W_k]$ represents the set of arcs of G having one end in W_i and the other in W_j, $i \neq j$.

A directed graph $G = (V, A)$ is *weakly connected* if no cut of G is empty. The graph d is said to be *k-connected graph* if $d^-(W) \geq k$ for all $W \subseteq V, \emptyset \neq$

$W \neq V$. A vertex $v \in V$ is called *cut vertex* of G if the number of connected components of the graph $G \setminus v$ is strictly greater than the number of related components of G.

If a graph $G = (V, A)$ does not contain circuit, then G is said acyclic.

For more details the reader is referred to [121]. In the next section, we introduce some graph classes.

2.1.3 Graph classes

A graph class $\mathcal{G}$ is the set of all graphs satisfying a certain property. In the following, we define all classes of graphs which will be appeared throughout this thesis.

A graph $G = (V, E)$ is called *complete* and denoted by K_n where $|V| = n$, if for any pair $u, v \in V$, $uv \in E$. An undirected graph which any two vertices of it are connected by exactly one path is called *tree* and a graph which is a collection of trees is known by *forest*. A *star* is a tree where at most one vertex has a degree greater than 1 or, equivalently, it is isomorphic to $K_{1,\ell}$ for some $\ell \geq 0$. The vertices of degree 1 (except the center when $\ell \leq 1$) are called *leaf* of the star while the remaining vertex is called *center* of the star. A ℓ-star is a star of ℓ leaves; when $\ell = 0$, the star is called *trivial* and it is reduced to a single vertex (the center).

A *bipartite graph* $G = (V, E)$ is an undirected graph in which the vertex set can be partitioned into two parts L and R such that the induced graph of each part makes an independent set. If in a bipartite graph, $N_G(u) = R$ for each vertex $u \in L$, it is called *complete bipartite graph* and is denoted by $K_{L,R}$. A *split graph* $G = (C \cup I, E)$ is an undirected graph where the vertex set $C \cup I$ is decomposable into a clique C and an independent set I.

A *k-tree* is a graph which can be formed by starting from a k-clique and then repeatedly adding vertices in such a way that each added vertex has exactly k neighbors completely connected together (this neighborhood is a k-clique). A graph is a *partial k-trees*, if it is a subgraph of a k-trees.

A graph is *planar*, if it can be embedded in a plane. It means that, it can be drawn on the plane in such a way that all the edge intersections placed at the endpoints of edges.

An *interval graph* is a graph in which there exists a family of intervals on the real line and there is a bijection between the vertices of the graph and the

family of intervals such that there is an edge in the graph if and only if the corresponding intervals have a non-empty intersection.

If for all cycles of four or more vertices of graph G, there is an edge that is not part of the cycle but connects two vertices of the cycle, the graph and the connected edge is called *chordal graph* and *chord* respectively. There are many characterizations of chordal graphs. One of them, known as Dirac's theorem, affirms a graph G is chordal if and only if each minimal vertex separator of G is a clique. For any integer $k \geq 3$, a graph is called *k-chordal* if it has no induced cycle of length greater than k. Thus, chordal graphs are precisely the 3-chordal graphs. In particular the class of 4-chordal graphs contains another well known class of graphs called *weakly triangulated* graphs or also *weakly chordal*. This class is introduced in [80], in view of extending chordal graphs as the class with no chordless cycle on five or more vertices in $G = (V, E)$ or in its complement $\overline{G} = (V, \overline{E})$, or equivalently, the graph contains neither a *hole* nor an *anti-hole*.

Given a graph H, a graph is H-free, if it does not contain H as an induced subgraph. A *cograph* is a graph which can be formed by starting from a single vertex and by repeating application of complementation and vertex-disjoint union. These are precisely the P_4-free graphs. A *line graph* of a graph G, denoted by $L(G)$ is a graph such that whose vertices represent the edges of G and two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G.

Another interesting graph class is *random graphs*. The theory of random graphs [62, 59, 64, 60, 61, 63] is in the intersection of graph theory and probability theory. It has been found originally by Erdős - Rényi to give a probabilistic construction of a graph with large girth and large chromatic number.

A random graph is obtained by starting with a set of n isolated vertices and adding successive edges between them at random. The aim of the study in this field is to determine at what stage a particular property of the graph is likely to arise. Different random graph models produce different probability distributions on graphs. Most commonly studied is the one proposed by Edgar Gilbert [72], denoted $G(n, p)$, in which every possible edge occurs independently with probability $0 < p < 1$. A closely related model, the Erdős - Rényi model denoted $G(n, M)$, assigns equal probability to all graphs with exactly M edges.

Its practical applications are found in all areas in which complex networks need to be modelled a large number of random graph models are thus known, mirroring the diverse types of complex networks encountered in different areas.

Graphs are usually used as an underlined structure to combinatorial opti-

mization problems. The next section is devoted to combinatorial optimization.

2.2 Combinatorial optimization

Combinatorial Optimization is a branch of operations research related to the computer science and applied mathematics. It aims to study optimization problems where the set of feasible solutions is discrete or can be reduced to a discrete one. Combinatorial optimization deals with problems that can be formulated as follows. Let $E = \{e_1, \ldots, e_n\}$ be a finite set called *basic set*, where with each element e_i is associated a weight $c(e_i)$. Let $\mathcal{F}$ be a family of subsets of E. If $F \in \mathcal{F}$, then $c(F) = \sum_{e_i \in F} c(e_i)$ is the weight of F. The problem consists in finding an element F^* of $\mathcal{F}$ whose weight is minimum or maximum. The set $\mathcal{F}$ represents the set of feasible solutions of the problem.

The term *optimization* means that we are looking for the best feasible solution among the elements of $\mathcal{F}$. The term *combinatorial* refers to the discrete structure of $\mathcal{F}$. In general, this structure is related to a discrete underlying one, which is, most of the time a graph.

It is also worth to mention that, in general, the number of feasible solutions $|\mathcal{F}|$ is exponential, which makes it difficult or even impossible to solve the associated combinatorial optimization problem with an enumerative procedure. Such a problem is hence considered as a hard problem.

Efficient methods have therefore been developed to formulate and solve this type of problems. In the literature, we find various methods to solve combinatorial optimization problems such as graph theory, linear and non-linear programming, integer programming, etc. In particular, polyhedral approaches have proved to be powerful for optimally solving these problems. This will be detailed in further sections of the chapter.

During the last decades, combinatorial optimization has developed considerably from both theoretical and practical points of view. Indeed, many real-world problems from areas as diverse as transport, telecommunications, biology, VLSI circuit and statistical physics have been formulated and solved using efficient combinatorial optimization techniques.

These techniques have been proved to be effective from a complexity point of view. And this shows that combinatorial optimization is closely related to other fundamental theories, especially algorithmic and complexity theories, issues that will be discussed in the next section.

2.3 Algorithmic and complexity theory

The interest to computational theory and complexity began with the works of Cook [49], Edmonds [56] and Karp [90]. Algorithmic and complexity theory is a branch of computer science whose objective is to classify problems according to their inherent difficulty. In particular, problems of combinatorial optimization are considered as either "easy" or "difficult" problems. For more details on this topic, the reader is referred to [70].

A *problem* is a question to which we wish to find an answer. This question usually depends on some input parameters. A problem is posed by giving a list of these parameters as well as the properties that these parameters must satisfy. An *instance* of a problem is obtained by giving specific values to all its input parameters. An *algorithm* is a sequence of elementary operations that, when given an instance of a problem as input, gives the solution of this problem as output. The number of input parameters necessary to describe an instance of a problem is called the *size* of that problem.

An algorithm is said to be in $O(f(n))$ if there exists $c > 0$ and $n_0 \in \mathbb{N}$ such that the number of elementary operations that are necessary to solve an instance of size n is at most $c.f(n)$ for all $n \geq n_0$. If f is a polynomial function, then the algorithm is said to be polynomial. A problem belongs to the *class* P if, for each instance of the problem, there exists an algorithm that is polynomial in the size of the instance, allowing the resolution of the problem. Problems belonging to class P are said to be *easy*.

A *decision problem* is a question concerning the existence, for a given instance, of a configuration such that this configuration satisfies some properties. In other words, the solution to a decision problem can be one of the answers: *yes* or *no*. Let $\mathcal{P}$ be a decision problem and $\mathcal{I}$ the corresponding instances whose answer is yes. $\mathcal{P}$ belongs to the class NP (Non-deterministic Polynomial) if there exists a polynomial algorithm allowing to check if the answer of each instance of $\mathcal{I}$ is yes. It is clear that the class P is contained in the class NP (see Figure 2.7). And, in reality, the difference between P and NP has never been proved, however the conjecture is considered highly probable.

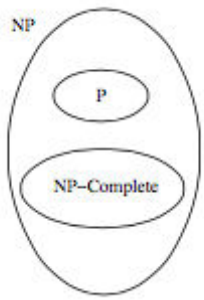


Figure 2.7: Relations between P, NP and NP-Complete

Among the problems that belong to the class NP, some problems are classified in a class called NP-*complete*. The NP-completeness is based on the notion of polynomial reduction. A decision problem P_1 is polynomially reduced to a decision problem P_2 if there exists a polynomial function f, such that for each instance I of P_1, the answer is yes if and only if the answer of $f(I)$ for P_2 is yes as well. This will be denoted by $P_1 \alpha P_2$. A problem P is said to be NP-*complete*, if it belongs to the class NP and if there exists a problem Q, known to be NP-complete, such that $Q \alpha P$. In practice, this theory was first used by Cook [49] who proves that SAT (the Satisfiability Problem) is NP-complete.

With every optimization problem is associated a decision problem. Moreover, every optimization problem whose associated decision problem is NP-complete is called NP-*hard*. Note that most of the combinatorial optimization problems are NP-hard.

Among the methods used to solve them, the polyhedral approach has been shown very efficient. This method is discussed in the following section.

2.4 Polyhedral approach and Branch-and-Cut

2.4.1 Elements of the polyhedral theory

Pioneered by the work of Jack Edmonds [57] for the matching problem, polyhedral approaches have shown to be powerful techniques for formulating, analysing and solving hard combinatorial optimization problems. These techniques consist in reducing the resolution of a combinatorial optimization problem to the resolution of a linear program, and this by describing (completely or partially) the convex hull of its solutions using a linear system of inequalities. This may often lead to polynomial time algorithms providing exact or

approximate solutions, help efficiently solve hard combinatorial problems and provide nice structural min-max relations.

In this section, we present only the basic notions for polyhedral theory. For a deeper study of this approach, the reader is referred to the works of Grötschel et al. [75], Schrijver [121] and Mahjoub [100].

Let $n \in \mathbb{N}$ be a positive integer and $x \in \mathbb{R}^n$. We say that x is a *linear combination* of $x_1, \ldots, x_k \in \mathbb{R}^n$, if there exist k scalar $\lambda_1, \lambda_2, \ldots, \lambda_k$ such that $x = \sum_{i=1}^{k} \lambda_i x_i$. If $\sum_{i=1}^{k} \lambda_i = 1$, then x is said to be an *affine combination* of $x_1, \ldots, x_k$. Moreover, if $\lambda_i \geq 0$ for all $i \in \{1, \ldots, k\}$ with $\sum_{i=1}^{k} \lambda_i = 1$, we say that x is a *convex combination* of $x_1, \ldots, x_k$.

Given a set $S = \{x_1, \ldots, x_k\} \in \mathbb{R}^{n \times k}$, the *convex hull* of S is the set of points $x \in \mathbb{R}^n$ which are convex combination of $x_1, \ldots, x_k$ (see Figure 2.8), that is

$$conv(S) = \{x \in \mathbb{R}^n | x \text{ is a convex combination of } x_1, \ldots, x_k\}.$$

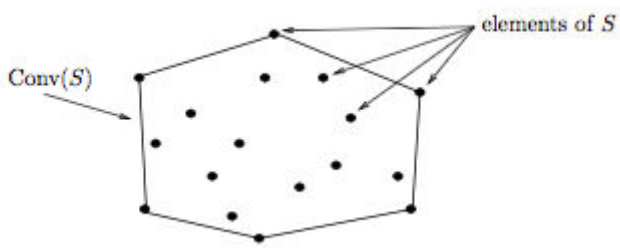


Figure 2.8: A convex hull

The points $x_1, \ldots, x_k \in \mathbb{R}$ are *linearly independent* if the unique solution of the system $x = \sum_{i=1}^{k} \lambda_i x_i = 0$ is $\lambda_i = 0$, $i = 1, \ldots, k$.

They are *affinely independent* if the unique solution of the system

$$x = \sum_{i=1}^{k} \lambda_i x_i = 0, \sum_{i=1}^{k} \lambda_i = 1,$$

is $\lambda_i = 0$, $i = 1, \ldots, k$.

A *polyhedron* P is the set of solutions of a linear system $Ax \leq b$, that is $P = \{x \in \mathbb{R}^n | Ax \leq b\}$, where A is an m-row n-columns matrix and $b \in \mathbb{R}^m$.

A *polytope* is a bounded polyhedron. A point x of P will be also called a *solution* of P.

A polyhedron $P \subseteq \mathbb{R}^n$ is said of *dimension* p if the maximum number of solutions of P that are affinely independent is $p+1$. We denote by $dim(P) = p$. We also have that $dim(P) = p - rank(A^=)$ where $A^=$ is the submatrix of inequalities of A that are satisfied with equality by all the solutions of P (implicit equalities). The polyhedron P is said to be full dimensional if $dim(P) = n$.

An inequality $ax \leq \alpha$ is *valid* for a polyhedron $P \subseteq \mathbb{R}^n$ if for every solution $\overline{x} \in P$, $a\overline{x} \leq \alpha$. This inequality is said to be *tight* for a solution $\overline{x} \in P$ if $a\overline{x} = \alpha$. The inequality $ax \leq \alpha$ is *violated* by $\overline{x} \in P$ if $a\overline{x} > \alpha$. Let $ax \leq \alpha$ be a valid inequality for the polyhedron P. $F = \{x \in P | ax = \alpha\}$ is called a *face* of P. We also say that F is a *face induced by* $ax \leq \alpha$. If $F \neq \emptyset$ and $F \neq P$, we say that F is a *proper* face of P. If F is a proper face and $dim(F) = dim(P) - 1$, then F is called a *facet* of P. We also say that $ax \leq \alpha$ induces a facet of P or is a *facet defining* inequality.

If P is full dimensional, then $ax \leq \alpha$ is a facet of P if and only if F is a proper face and there exists a facet of P induced by $bx \leq \beta$ and a scalar $\rho \neq 0$ such that $F \subseteq \{x \in P | bx = \beta\}$ and $b = \rho a$.

If P is not full dimensional, then $ax \leq \alpha$ is a facet of P if and only if F is a proper face and there exists a facet of P induced by $bx \leq \beta$, a scalar $\rho \neq 0$ and $\lambda \in \mathbb{R}^{q \times n}$ (where q is the number of lines of matrix $A^=$) such that $F \subseteq \{x \in P | bx = \beta\}$ and $b = \rho a + \lambda A^=$.

An inequality $ax \leq \alpha$ is *essential* if it defines a facet of P. It is *redundant* if the system $A'x \leq b'\}$ obtained by removing this inequality from $Ax \leq b$ defines the same polyhedron P. This is the case when $ax \leq \alpha$ can be written as a linear combination of inequalities of the system $A'x \leq b'$. A *complete minimal linear description* of a polyhedron consists of the system given by its facet defining inequalities and its implicit equalities.

A solution is an *extreme point* of a polyhedron P if and only if it cannot be written as the convex combination of two different solutions of P. It is equivalent to say that x induces a face of dimension 0. The polyhedron P can also be described by its extreme points. In fact, every solution of P can be written as a convex combination of some extreme points of P.

Figure 2.9 illustrates the main definitions given is this section.

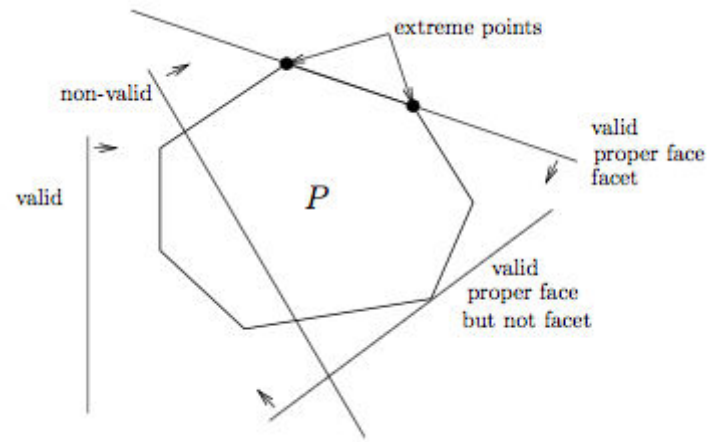


Figure 2.9: Valid inequality, facet and extreme points

Consider a combinatorial optimization problem $\mathcal{P}$. Let E be its basic set, $c(.)$ the weight function associated with its variables and $\mathcal{S}$ the set of its feasible solutions. Suppose that $\mathcal{P}$ consists in finding an element of $\mathcal{S}$ whose weight is maximum. The problem $\mathcal{P}$ can be hence written as $\max\{cx|x \in \mathcal{S}\}$. If $F \subseteq E$, then the 0-1 vector $x^F \in \mathbb{R}^E$ such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise, is called the *incidence vector* of F. The polyhedron $P(\mathcal{S}) = conv\{x^S|S \in \mathcal{S}\}$ is called the *polyhedron of the solutions* of $\mathcal{P}$ or *polyhedron associated with* $\mathcal{P}$. $\mathcal{P}$ is thus equivalent to the linear program $\max\{cx|x \in P(\mathcal{S})\}$. Notice that the polyhedron $P(\mathcal{S})$ can be described by a set of a facet defining inequalities. And when all the inequalities of this set are known, then solving $\mathcal{P}$ is equivalent to the resolution of a linear program.

Recall that the objective of the polyhedral approach for combinatorial optimization problems is to reduce the resolution of $\mathcal{P}$ to that of a linear program. Generally, it is difficult to characterize a polyhedron of a combinatorial optimization problem by a system of linear inequalities. In particular, when the problem is NP-hard there is a very little hope to find such a characterization. In addition, the number of inequalities describing this polyhedron is in general exponential. Therefore, even if we know the complete description of that polyhedron, its resolution remains in practice a hard task because of the large number of inequalities.

Fortunately, a technique called the *cutting plane method* can be used to overcome this difficulty. This method is described in what follows.

2.4.2 Cutting plane method

The cutting plane method is based on a crucial result in combinatorial optimization saying that only a partial description of the polyhedron can be sufficient to solve the problem optimally.

This result comes thanks to the work of Grötschel et al. [75] (1981) who show that the difficulty of solving a linear program does not depend on the number of inequalities of that program, but on the *separation problem* associated with the inequality system of the program. Consider a polyhedron P in $\mathbb{R}^n$ and let $Ax \leq b$ be its system of inequalities. The separation problem associated with P consists in checking if the point $\overline{x} \in \mathbb{R}^n$ satisfies all the inequalities $Ax \leq b$ and, if not, to find an inequality $ax \leq \alpha$ of $Ax \leq b$ violated by $\overline{x}$ (see Figure 2.10).

Grötschel, Lovász and Schrijver [75] prove that an optimization problem (for instance $\max\{cx, Ax \leq b\}$) can be solved in polynomial time if and only if the separation problem associated with $Ax \leq b$ is polynomial as well. This equivalence has permitted an important development of the polyhedral methods in general and the cutting plane method in particular.

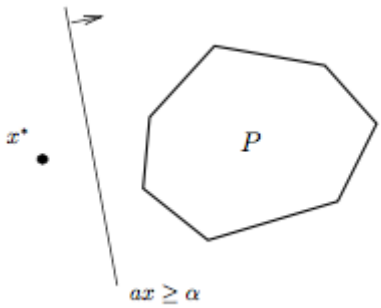


Figure 2.10: A hyperplan separating x^* and P

More precisely, the cutting plane method consists in solving successive linear programs, with possibly a large number of inequalities, by using the following steps. Let $LP = \max\{cx, Ax \leq b\}$ be a linear program and LP' a linear program obtained by considering a small number of inequalities among $Ax \leq b$. Let x^* be the optimal solution of the latter. We solve the separation problem associated with $Ax \leq b$ and x^*. This phase is called the *separation phase*. If every inequality of $Ax \leq b$ is satisfied by x^*, then x^* is also optimal for LP. If not, let $ax \leq \alpha$ be an inequality violated by x^*. Then we add $ax \leq \alpha$ to LP' and repeat this process until an optimal solution is found. Algorithm 1 summarizes the different cutting plane steps.

Algorithm 1: A cutting plane algorithm

Data: A linear program LP and its system of inequalities $Ax \leq b$
Result: Optimal solution x^* of LP

1 Consider a linear program LP' with a small number of inequalities of LP;
2 Solve LP' and let x^* be an optimal solution;
3 Solve the separation problem associated with $Ax \leq b$ and x^*;
4 **if** *an inequality* $ax \leq \alpha$ *of* LP *is violated by* x^* **then**
5 Add $ax \leq \alpha$ to LP';
6 Repeat step 2 ;
7 **else**
8 x^* is optimal for LP;
9 **return** x^*;

Note that at the end, a cutting-plane algorithm may not succeed in providing an optimal solution for the underlying combinatorial optimization problem. In this case a *Branch-and-Bound algorithm* can be used to achieve the resolution of the problem, yielding to the so-called *Branch-and-Cut algorithm*.

2.4.3 Branch-and-Cut algorithm

The Branch-and-Cut method, is a combination of the Branch-and-Bound and cutting-plane methods. The basic idea of branch-and-cut is simple. In each iteration, one solves a linear relaxation of the problem using a cutting plane algorithm. New valid inequalities are then added at each iteration to the current linear program. This permits to obtain increasingly better upper bounds on the value of the optimal solution of the combinatorial optimization problem. Branching occurs only when no violated inequalities are found to cut off infeasible solutions.

Consider again the combinatorial problem $\mathcal{P}$ defined above and assume now that its variables are binary. The polyhedron $P(\mathcal{S})$ is often not completely known because $\mathcal{P}$ may be NP-hard. In this case, it would not be possible to solve $\mathcal{P}$ as a linear program and in general, the solution obtained from the linear relaxation of $P(\mathcal{S})$ is fractional. The resolution of $\mathcal{P}$ can then be done by combining the cutting plane method with a Branch-and-Bound algorithm. Such an algorithm is called a Branch-and-Cut algorithm. Each node of the Branch-and-Bound tree (also called *Branch-and-Cut tree*) corresponds to a linear program solved by the cutting plane method.

Suppose that $\mathcal{P}$ is equivalent to $\max\{cx|Ax \leq b, x \in \{0,1\}^n\}$ and that $Ax \leq b$ has a large number of inequalities. A Branch-and-Cut algorithm

starts by creating a Branch-and-Bound tree whose root node corresponds to a linear program $LP_0 = \max\{cx|A_0x \leq b_0, x \in \mathbb{R}^n\}$, where $A_0x \leq b_0$ is subsystem of $Ax \leq b$ with a small number of inequalities. Then, we solve the linear relaxation of $\mathcal{P}$ that is $LR = \max\{cx|Ax \leq b, x \in \mathbb{R}^n\}$, using a cutting plane algorithm starting from the program LP_0. Let $x_0^* = (x_0^1, x_0^2, \ldots, x_0^k)$ be the optimal solution of LP_0 and $A_0'x \leq b_0'$ the set of inequalities added to LP_0 at the end of the cutting plane phase. If x_0^* is integral, then it is optimal for $\mathcal{P}$. If x_0^* is fractional, then we start the *branching phase*. This consists in choosing a variable, say x_0^1, having a fractional value and adding two nodes P_1 and P_2 in the Branch-and-cut tree. The nodes P_1 and P_2 correspond to the linear programs $LP_1 = \max\{cx|A_0x \leq b_0, A_0'x \leq b_0', x_0^1 = 0, x \in \mathbb{R}^n\}$ and $LP_2 = \max\{cx|A_0x \leq b_0, A_0'x \leq b_0', x_0^1 = 1, x \in \mathbb{R}^n\}$, respectively. We solve the linear program $LR_1 = \max\{cx|Ax \leq b, x_0^1 = 0, x \in \mathbb{R}^n\}$ (resp. $LR_2 = \max\{cx|Ax \leq b, x_0^1 = 1, x \in \mathbb{R}^n\}$) by a cutting plane method starting from LP_1 (LP_2). If the optimal solution of LR_1 (resp. LR_2) is integral then, it is feasible for $\mathcal{P}$. Its value is thus a lower bound of the optimal solution of $\mathcal{P}$ and the node P_1 (resp. P_2) becomes a *leaf* of the Branch-and-Cut tree. If this solution is fractional, then we select a variable with a fractional value and add two children to node P_1 (resp. P_2), and so on.

Remark that at some node of the Branch-and-Cut tree, the addition of a constraint $x^i = 0$ or $x^i = 1$ may make the associated linear program infeasible. Also, even if the corresponding linear program is feasible, its optimal solution may be worse than the best known lower bound of the tree. In both cases, we proceed the *pruning phase* and that node is cut off from the Branch-and-Cut tree. The algorithm ends when all the nodes have been explored and all the leaves of the tree are pruned. At the end of the algorithm, the optimal solution of $\mathcal{P}$ is the best feasible solution among the solutions obtained along the Branch-and-Bound tree.

Figure 2.11 illustrates a Branch-and-Cut tree. That is a Branch-and-Bound tree where in each node $Pi, i = 1, \ldots, 4$, a cutting plane method is used to solve the linear relaxation of node P_i.

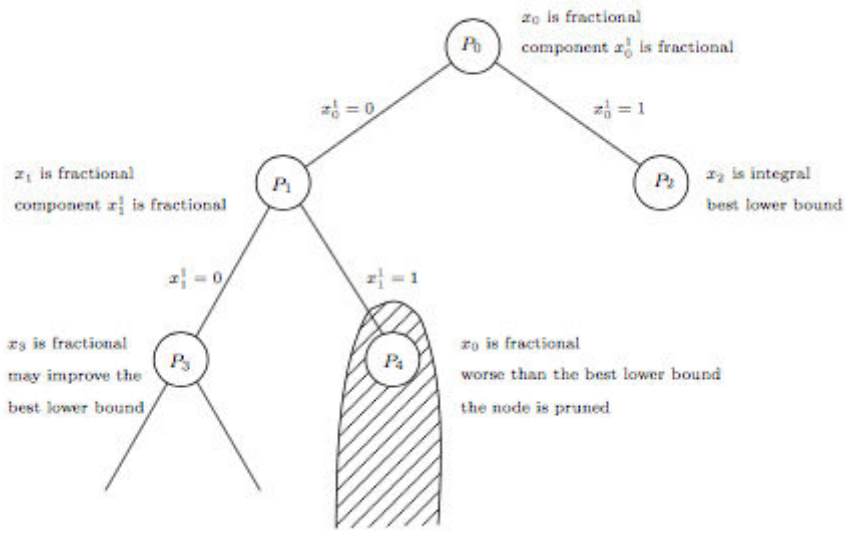


Figure 2.11: A Branch-and-Cut tree

The polyhedral approach and in particular the Branch-and-Cut method have been successfully applied to several combinatorial optimization problem that are considered difficult to solve, such as the Travelling Salesman Problem [33], the Max-Cut problem [43] and the Survivable Network Design Problem [91]. The efficiency of this approach depends on two important theoretical and practical issues. The first one consists in determining a good partial description of the convex hull of the solutions of the problem in terms of linear inequalities. The second issue is to devise efficient separation algorithms (exact or heuristic) for the identified classes of inequalities.

Note that the cutting plane method is effective when the number of variables is polynomial. However, when the number of variables is huge (for example exponential), one should resort to other appropriate methods such as the column generation method that we describe briefly in the following section.

2.4.4 Primal heuristics

The Branch-and-Cut algorithm can be improved by deriving good primal feasible solutions to the combinatorial optimization problem. This can be achieved using the so-called *primal heuristics*, which compute good lower bounds that can be used to prune suboptimal branches of the Branch-and-Cut tree.

Primal heuristics can be used at the root to find early a first feasible solution. They also may be used at a given node of the tree mainly to round fractional solutions and try to get a better bound. As a consequence, they help reducing

considerably the number of generated nodes of the tree as well as the CPU time. Moreover, this guarantees to have an approximation of the optimal solution of the problem for example when a CPU time limit has been reached.

2.5 Bilevel Programming

In this section, we give an overview about bilevel programming. A lot of definitions and notions stems from [51].

2.5.1 History: Stackelberg games

The first bilevel programming problem was introduced in 1934 by H.v.Stackelberg in his book [135] where he presented for the first time a bilevel programming formulation motivated by a market economy example. The model represents the situation where several decision markers, having generally different objectives, try to perform best decisions with respect to their own but they have to make decisions according to a certain hierarchy. If we consider two decision markers, the situation can be presented as the following. One of them will make independent decisions on the market (*the leader*), and the other must act in a dependent way (*the follower*).

On the one hand, the objective of the leader depends not only to his own decision but also on the reaction of the follower. So that, the leader will dictate the selling prices while selecting them but he has to anticipate the reactions of the follower. On the other hand, the follower must react to the decision of the leader. The set of the possible decisions and the objective of the follower are influenced by the leader's decisions.

The *Stackelberg game* is the problem that the leader has to solve. This consists in taking independent decisions, by observing the reactions of the follower on his decisions, and then trying to make good use of this advantage (realizing better objective). This problem can be formulated as following. We refer to the set of feasible strategies of the follower and of the leader by X and Y respectively. Let $F(x, y)$ and $f(x, y)$ be the objective functions of the leader resp. the follower. Given the decision y of the leader, the follower has to choose his decision $x(y)$ such that his objective is maximized on X. This consists in solving the following problem

$$x(y) \in \psi(y) = \underset{x}{Argmax}\{f(x, y) : x \in X\} \tag{2.1}$$

Knowing this selection, the leader solves the Stackelberg game:

$$\text{“}\underset{y}{max}\text{”}\{F(x,y) : y \in Y, x \in \psi(y)\} \tag{2.2}$$

Bilevel programming problems can be generalized with more than one decision maker and more than one level of hierarchy. In that case, we search for an an equilibrium (a.g. Nash equilibria). For more on this general cases the reader is refereed to [71, 126, 128].

2.5.2 Generality

A bilevel programming problem is an optimization problem having a second optimization problem as part of its constraints. The variables are partitioned between two vectors x and y such that x is the optimal solution of a second optimization problem parametrized in y. Let $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p, h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$, and consider the continuous second optimization problem (the follower), defined as follows:

$$\begin{aligned} \underset{x}{Min}\ & f(x,y) \\ & g(x,y) \leq 0, \\ & h(x,y) = 0. \end{aligned} \tag{2.3}$$

Let $\psi(y)$ denote the set of solutions of the problem (2.3). The function $\psi : \mathbb{R}^m \to 2^{\mathbb{R}^n}$ is called *point to set mapping* from $\mathbb{R}^m$ into the power set of $\mathbb{R}^n$. Denote the elements of $\psi(y)$ by $x(y)$. The goal of the bilevel programming problem is to select y^* describing the“data” for the lower level problem which together with the response $x(y^*) \in \psi(y^*)$ satisfies certain equality $H(x(y), y) = 0$ and/or inequality constraints $G(x(y), y) \leq 0$, and an objective function $F(x(y), y)$ is minimized. Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k, H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$, the leader can be formulated as

$$\begin{aligned} \text{“}\underset{y}{Min}\text{”}\ & F(x(y), y) \\ & G(x(y), y) \leq 0, \\ & H(x(y), y) = 0, \\ & x(y) \in \psi(y). \end{aligned} \tag{2.4}$$

The problem (2.4) is the *bilevel programming problem* or the *leader's problem*. The function F is called the *upper level objective* and the functions G and H are called *the upper level constraint functions*. Note that this definition of

the bilevel programming is valid only when the lower level solution is uniquely determined for each possible y.

The bilevel programming problem (2.4) is a generalization of different known optimization problems. If $F(x, y) = -f(x, y)$ for each x, y, it is a *max-min problem*. It is a *decomposition approach* when $F(x, y) = f(x, y)$. If the interdependence of both problems in y is dropped, problem (2.4) is a *bicriteria optimization problem*. What distinguish bilevel problems from biceiteria ones is that in the latter both objectives f and F are considered jointly. An optimal solution of the bicriteria optimization problem is in general not a feasible solution of the bilevel optimization problem where f is minimized over the feasible set and F is then minimized over the resulting set of optimal solutions.

Example 2.1 *(Stephan Dempe [51]) Let the follower problem be given as*

$$\underset{x}{Min}\{-x : x + y \leq 8, 4x + y \geq 8, 2x + y \leq 13\},$$

and consider the bilevel problem

$$\underset{y}{Min}\{3x + y : 1 \leq y \leq 6\}.$$

We refer to the set of all pairs (x, y) such that the constraints of both the follower and the leader are satisfied by M as shown in Figure 2.12. The feasible set of the follower is the intersection of the set M with the set of all points above the point $(0, y)$ on the y-axis. The follower function is minimized on this set, we then reach a point on the union of segments AE and ED which consequently is the optimal solution of the form

$$x(y) = \begin{cases} 6.5 - 0.5y & \text{if } 1 \leq y \leq 3, \\ 8 - y & \text{if } 3 \leq y \leq 6. \end{cases}$$

By varying y between 1 and 6, all points of the union of segments AE and ED are then obtained. This line corresponds to the set of feasible solution of the leader. Consequently, we can see that even for the simplest case, the bilevel programming problem is a nonconvex and a nondifferentiable optimization problem.

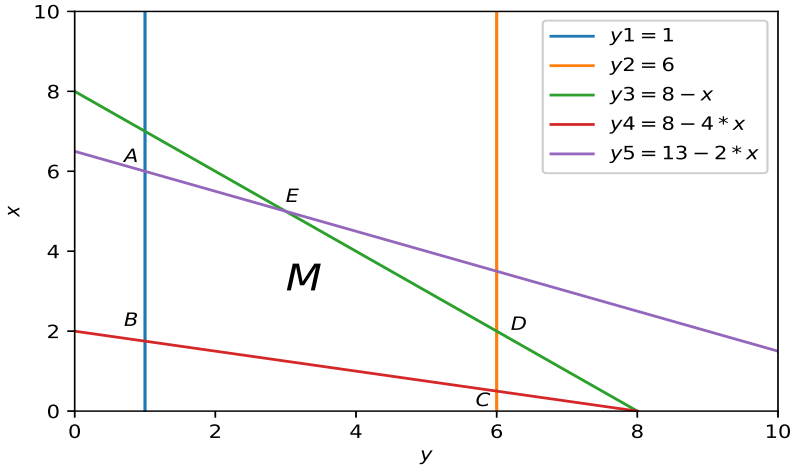


Figure 2.12: The linear bilevel programming problem

In the definition of the bilevel programming problem, the quotation marks are used to express the uncertainty in the definition of the problem in case of non-uniquely determined follower optimal solutions. In other words, if the follower has at most one optimal solution for all values of the parameter y, the quotation marks can be deleted and the familiar notion of an optimization problem arises. In the following, we present an example to highlight the case of nonunique follower optimal solutions.

Example 2.2 *(Lucchetti et al. [98]) Consider the follower problem defined as the following*

$$\psi(y) = \underset{x}{Argmax}\{-xy : 0 \leq x \leq 1\},$$

and let the bilevel program be defined as

$$\text{“}\underset{x}{Min}\text{”}\{x + y : x \in \psi(y), 0 \leq y \leq 1\}.$$

By inserting the optimal solution of the follower into the objective function of the leader, we obtain

$$\psi(y) = \begin{cases} \{0\} & \textit{if } y > 0, \\ \{1\} & \textit{if } y < 0, \\ [0,1] & \textit{if } y = 0, \end{cases}$$

and

$$F(x(y),y) = \begin{cases} = y^2, & y > 0, \\ = 1 + y^2, & y < 0, \\ \in [0,1], & y = 0. \end{cases}$$

It is clear that there is an ambiguity on the value of the function $F(x(y),y)$ at $y = 0$. The infimal function value of $F(x(y),y)$ is equal to 0 but this value is obtained only for $F(x(0),0) = 0$. This case is known as the optimistic position. If this is not the case then the bilevel problem has no solution.

Bilevel programming problems can be converted into one-level optimization problems by replacing the follower problem by its Karush-Kuhn-Tucker conditions [40, 39, 99, 58, 112]. The resulting problem aroused a lot of interest. But it is in general not equivalent to the bilevel problem. It is only possible to use this approach to the optimistic position of bilevel programming and there is no efficient way to use it for the pessimistic one.

With an optimistic position, the leader supposes that the follower is supporting him, which means that the former will select a solution $x(y)$ which is the best for the leader. Denote $\phi(y)$ the optimal solution of the problem

$$\underset{x}{Min}\{F(x,y) : x \in \psi(y)\}, \tag{2.5}$$

then the optimistic position of the bilevel programming problem is to solve

$$\underset{y}{Min}\{\phi(y) : G(x(y),y) \leq 0, H(x(y),y) = 0\}, \tag{2.6}$$

The optimal solution of the problem (2.7) is a pair $(\tilde{y}, \tilde{x(y)})$ such that $\tilde{y}$ is the optimal solution (2.6) and $\tilde{x(y)}$ is the optimal solution (2.5):

$$\underset{x,y}{Min}\{F(x,y) : G(x(y),y) \leq 0, H(x(y),y) = 0, x \in \psi(y)\}. \tag{2.7}$$

Most of works in bilevel programming focus on this problem.

If all functions f, g, h, F, G, H defining the problems (2.3) and (2.4) are assumed to be affine, the problem (2.4) is said to be *a linear bilevel problem.* In case of integrality constraints, we are facing *discrete bilevel problems.* Cutting planes algorithms have found large attention in solving discrete linear bilevel programming.

2.5.3 State of the art: Shortest Path Network Interdiction Problems (SPNIPs)

Bilevel programming is one of the most popular new topics to solve several security problems. For example, in [107] the authors study the electric grid security under disruptive threat problem, and in [65] a bilevel programming model for transmission network expansion planning with security constraints is proposed. The most related works to ours are *Shortest Path Network Interdiction Problems* (SPNIPs) which consist in maximizing the shortest s-t path length either by interdicting arcs [86, 92, 74, 37, 41] or by interdicting nodes [36, 50, 47]. In what follows, we will present the most general works among this references.

In [47] authors study the Minimum Vertex Blocker to Short Paths Problem (MVBP) which is defined as follows. Given a directed graph $G = (V, A)$, a source and a destination $s, t \in V$, a length $l_{ij} \in \mathbb{R}^+$ for $(i, j) \in A$ and an integer d, the MVBP consists in finding a subset $V' \subseteq V$ of minimum cardinality such that the shortest path from s to t in $G \setminus V'$ is at least d.

A generalized version of SPNIPs by interdicting arcs is given in [86], where instead of removing arcs, the leader can pay a given price to increase the length of the arcs. Let $G = (N, A)$, where N is the set of nodes and A is the set of arcs. With each arc $(i, j) \in A$ is associated a weight $c_{ij} \geq 0$. Interdiction increases the arc's weight to $c_{ij} + d_{ij}$ where $d_{ij} > 0$. If the value of d_{ij} is sufficiently large, then interdiction destroys arc (i, j). Consider r_{ij} the resource required to interdict arc (i, j), and r_0 the total amount of interdiction resource available. The problem is to maximize the shortest $s - t$ path length in the directed graph by interdicting arcs, where s is the source and t is the target. Let $\Gamma(i)^+$ (resp. $\Gamma(i)^-$) be the set of outgoing arcs of node i (resp. ongoing arcs of node i). Let x_{ij} be the binary variable indicating if the arc (i, j) is interdicted by the leader, and y_{ij} be the binary variable indicting if the arc (i, j) is traversed by the follower. The SPNIP is equivalent to this formulation:

$$
\begin{aligned}
&\underset{x}{Max}\,\underset{y}{Min} \sum_{(i,j)\in A} (c_{ij} + x_{ij}d_{ij})y_{ij} \\
&r^T x \leq r_0, \\
&\sum_{u\in\Gamma^+(v)} y_{vu} - \sum_{u\in\Gamma^-(v)} y_{uv} = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{if } v \notin \{s, t\} \\ -1 & \text{if } v = t \end{cases} \qquad \forall v \in V, \\
&y_{ij} \geq 0 \qquad \forall ij \in A, \\
&x_{ij} \in \{0, 1\} \qquad \forall ij \in A.
\end{aligned}
$$

In our work, a general description of an instance of the optimization problem that we will study can be given as 1) a graph with a set of source-target nodes representing the attacks (the source is the attacker and the target is a vulnerable asset), and with each arc is associated a positive weight representing the difficulty of propagation of an attacker from the initial end to the final end; 2) security requirements will be given as positive values associated with each attack (a source-target pair) and indicating a difficulty of propagation threshold to be respected by the length of the shortest path between the source and the target; 3) A set of countermeasures that can be installed for each vulnerable asset with a given installation cost and a given effect. The effect of a countermeasure on a node is simulated by increasing the weights of the ongoing arcs of the node by the effect of the countermeasure.

Our work is different from the one of Boros et al. [47], in the sense that instead of removing nodes, the leader can pay a given price to increase the length of its ongoing arcs in order to make it more difficult for an attacker to gain access to that node. This permits us to consider realistic countermeasures allowing to reduce the effect of a risk without completely eliminating it. Now, if increasing the length of the ongoing arcs of a given node yields to a very large value, then the interdiction is equivalent to removing the node and the problem reduces to [47]. Our works is also different from [86], in fact we minimize costs of interdiction while the length of the shortest path is to be increase to at least a positive value. The optimization problem that we will address in this thesis is more general than aforementioned works. We will consider multiple sources and destinations added to the fact that we consider a specific node interdiction technique where the weights of all the ongoing arcs of the interdicted node are to be increased at the same time. To the best of our knowledge, this node interdiction case has never been treated in the literature.

2.6 Concluding remarks

In this chapter we have introduced the mathematical and algorithmic background that will be used to develop our risk treatment approach which involves graph theory, algorithmic theory, combinatorial optimization, polyhedral approaches and bilevel programming. We have also discussed the related work and highlighted our contribution. The next chapter will be devoted to introduce our security risk assessment approach which will be based on graph theory.

Chapter 3

Security risk assessment: models and risk evaluation algorithm

Contents

3.1 Approach overview **46**
3.1.1 Risk analysis 46
3.1.2 Risk evaluation 48
3.2 The Risk Assessment Graphs (RAGs) **49**
3.2.1 Security metrics 49
3.2.2 The RAGs model 51
3.3 Most likely paths-based risk evaluation approach **52**
3.3.1 Risk propagation: the most likely path 52
3.3.2 Risk evaluation algorithm 54
3.4 SDN case study **56**
3.4.1 The Risk Assessment Graphs 56
3.4.2 Risk Evaluation 58
3.5 Simulations **59**
3.5.1 Random systems generation 60
3.5.2 Impact of the number of nodes 61
3.5.3 Impact of the topology and the accessibility changes p and β 61
3.5.4 Impact of the potentiality convergence speed α . . 62
3.6 Concluding remarks **63**

In this chapter, we develop our risk assessment approach. We first present the Risk Assessment Graphs (RAGs) model as a risk assessment tool taking into account the complexity and the evolution of a system over the time. The potentiality and the accessibility are introduced as essential metrics for the definition of the RAGs. Both of them are functions of time and indicate respectively the probability of exploiting a node in the RAG, and the frequency of access between the nodes.

Next, based on this model, we present a quantitative risk evaluation approach which includes the notion of risk propagation by considering the attackers most likely paths in the RAGs. We introduce and compute three security metrics: the propagated risk, the node risk, and the global risk. We further illustrate our approach in a Software Defined Network (SDN) case study. Finally, we conduct numerical simulations and discuss the sensitivity of our metrics to the potentiality, the topology and the accessibility changes.

3.1 Approach overview

In this section, we give some definitions and present an overview of our approach, summarized in Figure 3.1. The risk assessment approach we propose involves two major steps: risk analysis and risk evaluation. For each step we describe the input parameters as well as how the output is generated. More precisely, the topology and the vulnerability databases are used as input of the risk analysis step in order to generate the RAGs. These graphs will be used as input of the risk evaluation process in order to give an evaluation of the proposed security metrics.

3.1.1 Risk analysis

The risk analysis process starts through deriving the input parameters from the topology and the vulnerability databases (e.g. NVD). The identified parameters are considered as essential factors of risk. More precisely, the topology database contains information about the assets, their interconnections as well as the frequency of connexion between each pair of assets which is called *the accessibility*. An asset could be physical (e.g. data center, switch, router, etc...) or virtual (e.g. virtual machine, network function, etc...). A particular subset of assets can be used by an attacker as an entry point to the system. This entry points will be called *the access points* and the risk that can propagate within the system starts from these elements. The vulnerability database contains three kind of information: the vulnerabilities associate

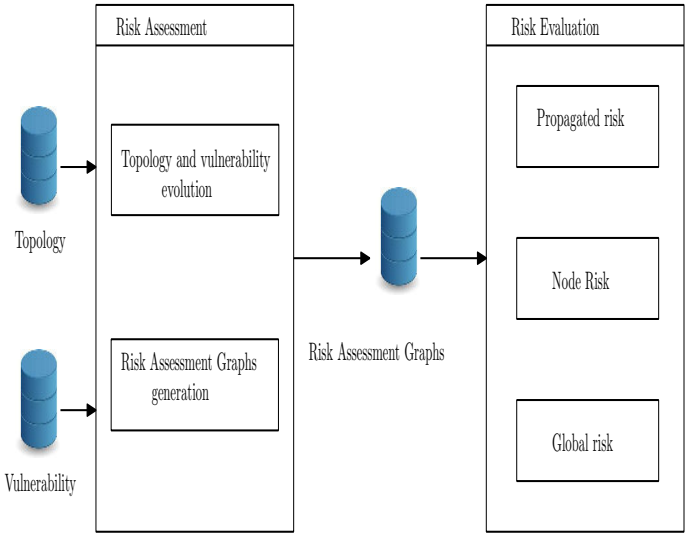


Figure 3.1: Framework Description

with each asset, their likelihood (How easy it is to exploit the vulnerability?), and their impact (what is the level of the damage induced by exploiting a vulnerability?). The risk analysis process we propose includes two steps:

1) Topology and vulnerability evolution over the time: We study the system in a discrete time horizon I. After identifying the system topology and its vulnerabilities, we propose security metrics that take into account the evolution of these two factors over the time. To this end, we introduce *the potentiality* and *the accessibility* as functions of time. At each time slot in the time horizon I, the potentiality evaluates the likelihood of each vulnerability and the accessibility gives the frequency of connection between the system assets.

2) RAGs generation: For each time slot in I, a RAG is generated. The RAG represents the system state at a given time slot as an oriented graph whose nodes are either an asset-vulnerability pair (which represents an asset to be secured with respect to a given vulnerability) or an access point (an attacker). An arc between two nodes in the RAG represents the possibility of exploitation of the vulnerability of the final end from the vulnerability of the initial end. With each arc is associated a positive weight representing *the difficulty of propagation* of an attacker from the initial end to the final end. This metric is evaluated using the accessibility between the assets associated with the ends of the arc and the potentiality associated with the vulnerability of the final end. Let s be an access point and t be an asset-vulnerability node. Every $s - t$ pair corresponds to *a potential attack* and a path from s to t corresponds to *a potential attack path* from the access point s to the

asset-vulnerability node t. The length of this path represents the difficulty of propagation of the attacker s to reach the node t. Note that the length of a path is the sum of the weights of the arcs composing the path.

These RAGs are used in our risk evaluation approach in order to develop security metrics giving a quantitative evaluation of the system risks while taking into account the vulnerability and the system topology features as well as the way they evolve over the time. Added to that, RAGs can be used in the risk treatment step to define the security requirements to be respected. Specifically, given a propagation difficulty threshold associated with each $s-t$ pair, a path from s to t is said to be *secured* if its length is greater than or equal to the $s-t$ threshold.

3.1.2 Risk evaluation

In this step we develop an algorithm which uses the RAGs to define three security metrics.

1) The propagated risk: When propagating in the system, the attacker may be confronted to several paths that can be used to reach its target. From a protection strategy point of view, the highest level of protection requires securing all the paths between each access point and each asset-vulnerability node. For a given attack $s-t$, when the $s-t$ path allowing a maximum risk propagation (the one having the minimum propagation difficulty) is secured, then so it is for all the paths of the RAG (since their length is greater then or equal to the one having the minimum propagation difficulty). Consequently, we need a security metric that is able to indicate if the path of maximum risk propagation in the RAGs is secured or not. We refer to this path of maximum risk propagation by *the most likely path*, and the propagated risk metric will be deduced from its length.

2) The node risk: We evaluate the total risk for a given node as the sum of the propagated risks from each intruder to each asset-vulnerability node.

3) The global risk: We evaluate the global risk as the sum of the risks on each asset-vulnerability node in the RAGs.

The RAGs will be used to evaluate the three security metrics at each time slot in I. In the next section, we formally describe the RAGs and introduce the elementary metrics (potentiality and accessibility) used to define it.

3.2 The Risk Assessment Graphs (RAGs)

In this section, we formally define the RAGs model. To this end, let us first give some definitions and introduce the security metrics that will be useful to define the weights of the arcs in the graph.

3.2.1 Security metrics

Let $I = \{0, 1, \ldots, f\}$ be a discrete time set and $i \in I$. We refer to the set of the assets of the system at time slot i by Λ_i. For each $a \in \Lambda_i$ we denote by V_a^i the set of vulnerabilities associated to the asset a. The pair (a, v) such that $a \in \Lambda_i$ and $v \in V_a^i$ is called an *asset-vulnerability node* and we refer to the set of asset vulnerability nodes at time i by T_i. It is clear that, for a given time slot $i \in I$ and a given asset $a \in \Lambda_t$ there is as many asset-vulnerability nodes as vulnerabilities in V_a^i. For each asset-vulnerability node we define the potentiality function and the impact as follows.

The potentiality function represents the chance for a vulnerability to be exploited by an attacker on a given asset, at least once before a given time slot. This should be an increasing function of time, since the more time passes the easier it is for an attacker to exploit a vulnerability. However, before a given time slot $i \in I$, the number of attacker exploitations is uncertain. One can assume that these numbers are independent random variables defined for each $i \in I$ and for each asset-vulnerability t and denoted by X_t^i, such that each of which yields to an exploitation with probability p_t at each time $i \in I$. Consequently, X_t^i follows a binomial distribution with parameters i and p_t. More formally,

Definition 3.1 *The potentiality function f_t^i of an asset vulnerability node $t = (a, v)$ at time $i \in I$ is the probability of the vulnerability v to be exploited on asset a at least one time before the time slot i, that is*

$$f_t^i = P(X_t^i \geq 1) = 1 - P(X_t^i = 0) = 1 - (1 - p_t)^i. \tag{3.1}$$

Equation (3.1) could be generalized by the function

$$f_t^i(\alpha_t) = 1 - (1 - p_t)^{\alpha_t i}. \tag{3.2}$$

where α_t is a parameter between 0 and 1 controlling how fast the potentiality of the node t converges to 1.

Now, we define the impact metric.

Definition 3.2 *The impact I_t of an asset-vulnerability node $t = (v, a)$ is a positive value representing the level of damage generated by exploiting v on a.*

We assume that the impact is constant over the time. Note that the CVSS scoring method can be used to give an estimation of the impact I_t, and of the exploitation p_t.

We define now the application Δ which for each asset-vulnerability node $t = (a, v)$, gives its associated asset $a \in \Lambda_t$.

$$\Delta : \underset{t=(a,v)}{T_i} \underset{\mapsto}{\rightarrow} \underset{a}{\Lambda_t}$$

Definition 3.3 *Let $i \in I$ and $t_1, t_2 \in T_i$, the accessibility function denoted by $g^i_{(t_1,t_2)}$ is a scalar between 0 and 1 indicating the frequency of access between $\Delta(t_1)$ and $\Delta(t_2)$ during the time from i to $i+1$, $i \in I \setminus \{f\}$.*

It is possible to exploit a node from another only if it is vulnerable and accessible. Formally, at a given time i, an attacker in $u \in T_i$ can damage a node $v \in T_i$ if $g^t_{(\Delta(u),\Delta(v))} \neq 0$, and $f^t_v \neq 0$. In that case, the higher is the potentiality of v, the more likely is the propagation and the same it is for the accessibility. Therefore, we define the propagation function that indicates how it is easy for an attacker to propagate from one node to another:

Definition 3.4 *Let $i \in I$, and $(t_1, t_2) \in T_i$. The propagation function is defined as*

$$h^i_{(t_1,t_2)} = f^i_{t_2} \times g^i_{(t_1,t_2)}. \tag{3.3}$$

We can define the function w that indicates how it is difficult for an attacker to propagate from one node to another at a given time slot.

Definition 3.5 *Let $i \in I$, and let $(t_1, t_2) \in A_i$. The Propagation difficulty function is defined as*

$$w^i_{(t_1,t_2)} = -log(h^i_{(t_1,t_2)}). \tag{3.4}$$

The function w has values in $\mathbb{R}_+$ and is used to evaluate the arcs of the RAGs that is defined in the next section.

3.2.2 The RAGs model

Let $I = \{0, 1, \ldots, f\}$ be a discrete time set, the RAGs are a set of directed graphs $\{G_i = (V_i, A_i) : i \in I\}$. Let $i \in I$, the set of nodes V_i is partitioned into two specified subsets S_i and T_i where $V_i = S_i \cup T_i$ and $S_i \cap T_i = \emptyset$. A node in S_i represents an access point and a node in T_i represents an asset-vulnerability pair. The set of arcs A_i is defined such that for all $u, v \in V_i$, an arc from u to v exists if $v \notin S_i$ and its exploitation from u is possible. The sub-graph of G_i induced by the nodes associated to the same asset $a \in \Lambda_i$ are cliques. With each arc $(u, v) \in A_i$ is associated weights $w^i_{uv} \in \mathbb{R}_+$ representing the propagation difficulty of an attacker from u to v at time slot i.

Remark 3.6 *By definition, an asset-vulnerability node can not damage an attacker, so that arcs from asset vulnerability nodes to access points don't belong to A_i.*

An asset is always accessible from itself over the time which means that for all $i \in I, t_1, t_2 \in T_i$ if $\Delta(t_1) = \Delta(t_2)$, we have $g^i_{(t_1,t_2)} = 1$. That's why, there is an arc between each pair of asset vulnerability nodes having the same asset. Consequently, the sub-graph of G_i induced by the nodes associated to the same asset $a \in \Lambda_i$ are cliques and G_i contains at least $|\Lambda_i|$ cliques.

A simplified representation of the RAG model at a given time slot is given in Figure 3.2.2. The nodes s_1 and s_2 are the access points. The nodes t_i, $i = 1, \ldots, 5$ are the asset-vulnerability nodes. The arcs are labelled by the difficulty of propagation w. A direct exploitation of an asset-vulnerability node from an access point is represented by an arc, e.g., (s_1, t_1) and (s_2, t_5), and an indirect exploitation corresponds to a path, e.g., path $(s_1, t_1, t_2, t_4, t_5)$. Several paths can exist between the same source and destination, for example the paths $(s_1, t_1, t_2, t_4, t_5)$ and $(s_1, t_1, t_3, t_4, t_5)$. We can see that asset a_1 has three vulnerabilities v_2, v_3 and v_4 which induces three nodes $t_2 = (a_2, v_2)$, $t_2 = (a_2, v_3)$ and $t_2 = (a_2, v_3)$ forming a clique in the RAG. Also, some arcs are not bidirectional (e.g., (t_1, t_2)) due to the fact that the access between two assets can be active only in one sense ($g^i_{(t_2,t_1)} = 0$).

To conclude, the RAGs model gives a complete representation of the security system we are studying including the vulnerabilities, the topology and their evolution over the time. RAGs allow analysing the topology of the system (by using the accessibility), and the vulnerabilities (by using the potentiality). The potentiality and the accessibility metrics are used to develop the propagation difficulty metric which is considered as the weight of RAG's arcs. In the next section, our risk evaluation approach use this metric to cope with the risk propagation from each attacker to each asset-vulnerability node.

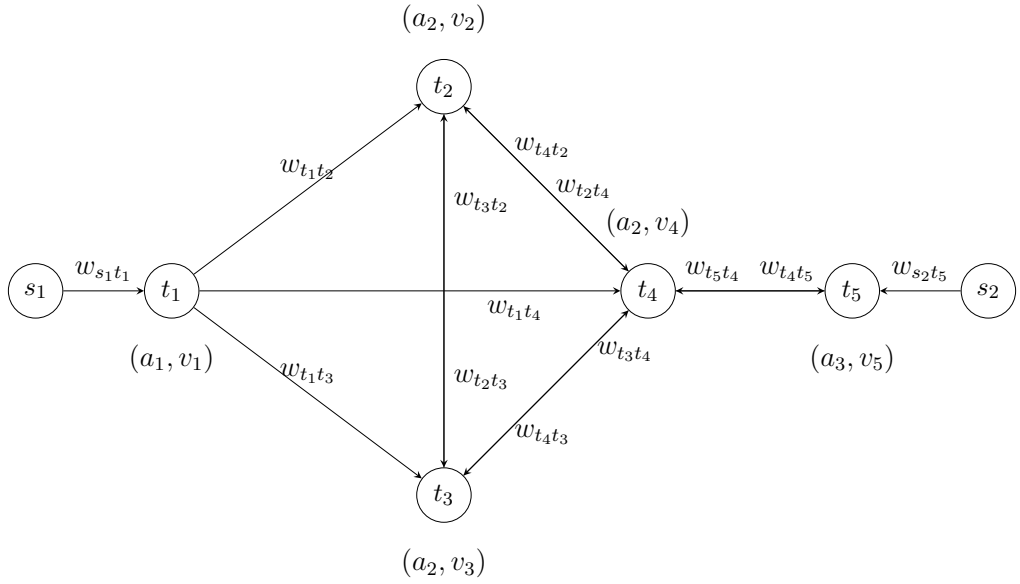


Figure 3.2: Simplified representation of a RAG

3.3 Most likely paths-based risk evaluation approach

We introduce now the most likely path notion and define our security metrics namely the propagated risk, the node risk and the global risk. We then present the risk evaluation algorithm.

3.3.1 Risk propagation: the most likely path

The concept of the propagation difficulty on an arc can be easily generalized for paths between the access points and the asset-vulnerability nodes in the RAG. For a given $i \in I$, $s \in S_i$ and $t \in T_i$, we denote by $P^i_{s,t}$ the set of $s - t$ paths.

Definition 3.7 *Let $i \in I$, $s \in S_i$, $t \in T_i$ and $P = (v_1, \ldots, v_k)$, where $v_1 = s$ and $v_k = t$, be a path of length k in $P^i_{s,t}$. The propagated potentiality of P is a value between 0 and 1 defined as*

$$H^{P,i}_{s,t} = \prod_{j=1}^{k-1} h^i_{(v_j,v_{j+1})}. \tag{3.5}$$

Definition 3.8 *Let $i \in I$, $s \in S_i$, $t \in T_i$ and $P = (v_1, \ldots, v_k)$, where $v_1 = s$ and $v_k = t$, be a path of length k in $P^i_{s,t}$. The propagation difficulty of P is defined as*

$$W^{P,i}_{s,t} = \sum_{j=1}^{k-1} w^i_{(v_j,v_{j+1})}. \tag{3.6}$$

The propagated potentiality of an $s-t$ path P at a given time slot i represents how it is easy for an attacker in s to exploit the node t while propagating on the path P. The most likely $s-t$ path (from an attacker point of view) denoted by P^*, is the path of maximum propagated potentiality and is given by

$$H^{P^*,i}_{s,t} = \underset{P \in P_{s,t}}{max} \{H^{P,i}_{s,t}\}. \tag{3.7}$$

In the literature, the problem (3.7) corresponds to the most reliable path problem [38]. We will see that this problem can be reformulated as a shortest path problem [67] in our RAGs. We have

$$\begin{aligned}
\underset{P \in P_{s,t}}{max} \{H^{P,i}_{s,t}\} &\Leftrightarrow \underset{P \in P^i_{s,t}}{max} \{\prod_{j=1}^{k-1} h^i_{(v_j,v_{j+1})}\} \\
&\Leftrightarrow \underset{P \in P^i_{s,t}}{min} \frac{1}{\prod_{j=1}^{k-1} h^i_{(v_j,v_{j+1})}} \\
&\Leftrightarrow \underset{P \in P^i_{s,t}}{min} log(\frac{1}{\prod_{j=1}^{k-1} h^i_{(v_j,v_{j+1})}}) \\
&\Leftrightarrow \underset{P \in P^i_{s,t}}{min} -log(\prod_{j=1}^{k-1} h^i_{(v_j,v_{j+1})}) \\
&\Leftrightarrow \underset{P \in P^i_{s,t}}{min} \sum_{j=1}^{k-1} -log(h^i_{(v_j,v_{j+1})})
\end{aligned}$$

As $w^i_{(v_j,v_{j+1})} = -log(h^i_{(v_j,v_{j+1})}$, the problem of finding the propagated potentiality with respect to $i \in I, s \in S_i, t \in T_i$ and $P \in P^i_{s,t}$ is consequently equivalent to the this shortest path formulation is G_i:

$$\underset{P \in P^i_{s,t}}{min} \{\sum_{j=1}^{k-1} w^i_{(v_j,v_{j+1})}\}. \tag{3.8}$$

Now, in order to compute the propagated potentiality of the most likely path P^*, we simply run a shortest path algorithm (by using Dijkstra algorithm) between s an t in G_i. The length of the shortest path denoted by $L_G(P^*)$ will give us the propagated potentiality of the most likely path, indeed

$$H_{s,t}^{P^*,i} = \frac{1}{exp(L_G(P^*))}. \tag{3.9}$$

3.3.2 Risk evaluation algorithm

Let $i \in I$, $s \in S_i$ and $t \in T_i$, the propagated risk $R_{s,t}^i$ from an access point s to a node t is the combination of two factors: the *propagated potentiality* $H_{s,t}^i$ and the *impact* I_t.

Definition 3.9 *Let $i \in I$, $s \in S_i$, and $t \in T_i$. The propagated risk from s to t, at the time slot i, is given by*

$$R_{s,t}^i = H_{s,t}^{P^*,i} I_t. \tag{3.10}$$

For each asset-vulnerability node, the summation of the propagated risks from all access points gives the node risk. This is defined as follows.

Definition 3.10 *Let $i \in I$ and $t \in T_i$*

$$R_t^i = \sum_{s \in S_i} R_{s,t}^i. \tag{3.11}$$

Finally,

Definition 3.11 *Let $i \in I$, the global risk at time i is given by the summation of the nodes risks, that is:*

$$R^i = \sum_{t \in T_i} R_t^i. \tag{3.12}$$

Our risk evaluation algorithm is presented in Algorithm 2.

Remark 3.12 If a global risk threshold is given, and the propagation difficulty threshold is supposed to be the same for each pair of access point and asset-vulnerability node, then this threshold can be deduced from the global

Algorithm 2: Risk evaluation algorithm

```
Data: G_i for all t ∈ I
Result: R^i_{s,t}, R^i_t, R^i for all t ∈ I, s ∈ S_i, t ∈ T_i
R^i_{s,t} = 0
R^i_t = 0
R^i = 0
/* Propagated risk                                          */
for i ∈ I do
    for s ∈ S_i do
        for t ∈ T_i do
            L_G(P*) = Dijkstra(s, t, i)
            if L_G(P*)! = ∞ ;                 // an s − t path exists
            then
                H^{P*,i}_{s,t} = 1/exp(L_G(P*))
            else
                H^i_{s,t} = 0
            R^i_{s,t} = H^{P*,i}_{s,t} I_t
/* Node risk                                                */
for i ∈ I do
    for s ∈ S_i do
        for t ∈ T_i do
            R^i_t += R^i_{s,t}
/* Global risk                                              */
for i ∈ I do
    for t ∈ T_i do
        R^i += R^i_t
```

risk threshold as follows. Let $i \in I$ and R^i be the global risk threshold, and assume that for all $(s,t) \in S_i \times T_i$, $H_{s,t}^{P^*,i} = H$ and $W_{s,t}^{P,i} = W$. We have that

$$R^i = \sum_{t \in T_i} R_t^i = \sum_{s \in S_i, t \in T_i} R_{s,t}^i = \sum_{s \in S_i, t \in T_i} H_{s,t}^{P^*,i} I_t = \sum_{s \in S_i, t \in T_i} H I_t = \mid S \mid H \sum_{t \in T_i} I_t$$

Hence, $H = \frac{R^i}{|S| \sum\limits_{t \in T_i} I_t}$.

Therefore, using equation (3.9), the propagation difficulty threshold threshold for all $(s,t) \in S_i \times T_i$ will be given by

$$W = -log(\frac{R^i}{\mid S \mid \sum\limits_{t \in T_i} I_t}). \tag{3.13}$$

3.4 SDN case study

In this section, we illustrate our risk assessment methodology with a Software-Defined Networks (SDN) case study. In Figure 3.3, we give a SDN architecture and we examine *the dynamicity in time* induced by the evolution of the potentialities and the accessibilities, as well as *the dynamicity in space* which is induced by adding a new device and cutting some accessibilities.

Figure 3.3(a) corresponds to the initial state of the system. we have two hosts that will send flow to each other. Host 1 is connected with switch 1, host 2 is connected with switch 3, and switch 2 matches switch 1 with switch 3. The flow transfer is supposed to be bidirectional inside the SDN data plane, as well as between the controller plane and the data plane. The assets of the system (the controller and the switches) are $CISCO$ products, named using the standard Common Platform Enumeration (CPE). The CPE is used as a standardized method of describing and identifying classes of applications, operating systems, and hardware devices present among enterprises computing assets [4]:

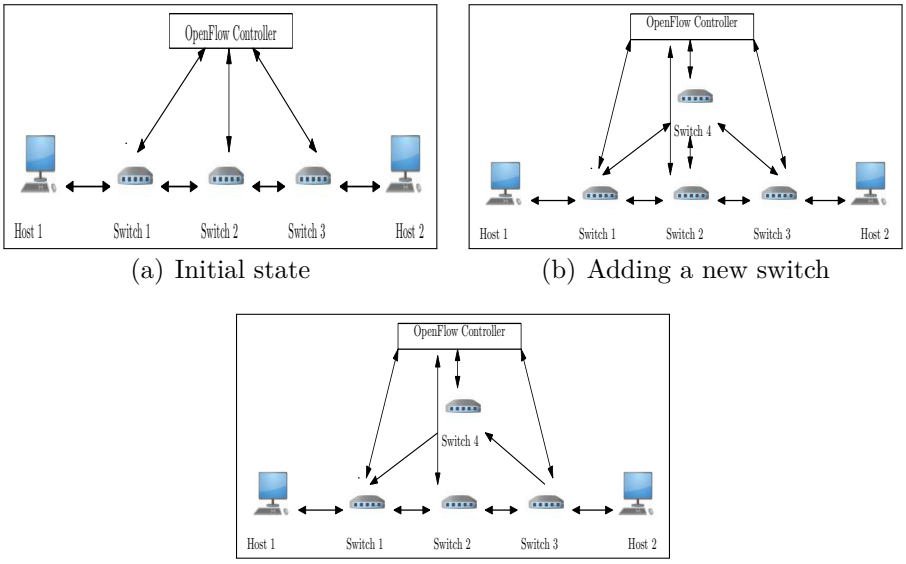
- Controller (denoted by C): $cpe : /h : cisco : 2106_wireless_lan_controller$;
- Switches 1 and 2 (denoted by s_1, s_2 and s_3): $cpe : 2.3 : h : cisco : nexus_5548up$.

In Figure 3.3(b), a new switch s_4 is added in the SDN architecture. It is connected with the controller and the other switches with a bidirectional links. The links between s_4, s_3; s_4, s_3; s_4, s_3 and s_4, s_3 are deleted in Figure 3.3(c). The definition of the RAGs associated to this system and the impact of the dynamicity of the system on the security metrics are examined in the following.

3.4.1 The Risk Assessment Graphs

Now, we show the construction and the visualization of the RAGs afor the SDN case study introduced in Figure 3.3. We study the system in the discrete time set $I = \{1, \ldots, 4\}$. Table 3.1 contains a detailed description of the vulnerabilities and their associated assets at the initial state of the system ($i = 1$). The exploitation and the impact of vulnerabilities are derived from the NVD in 13/10/2016.

The potentiality function (3.1) is used to label the nodes of the RAGs. For this example we assume that the accessibilities are previously determined. As seen in Figure 3.4, they are equal to 1 for all the arcs, at $i = 1, 2$. At $i = 3$,



(a) Initial state (b) Adding a new switch

(c) Cutting access

Figure 3.3: SDN Use Case

the accessibilities between s_2 and all the other assets, as well as those between all the other assets and s_2 become 0.5. At $i = 4$, the accessibilities between (s_4, s_3), (s_4, s_3), (s_4, s_3) and (s_4, s_3) are equal to zero and the corresponding arcs are deleted: $((s_4, v_3), (s_2, v_3))$, $((s_2, v_3), (s_4, v_3))$, $((s_1, v_3), (s_4, v_3))$ and $((s_4, v_3), (s_3, v_3))$.

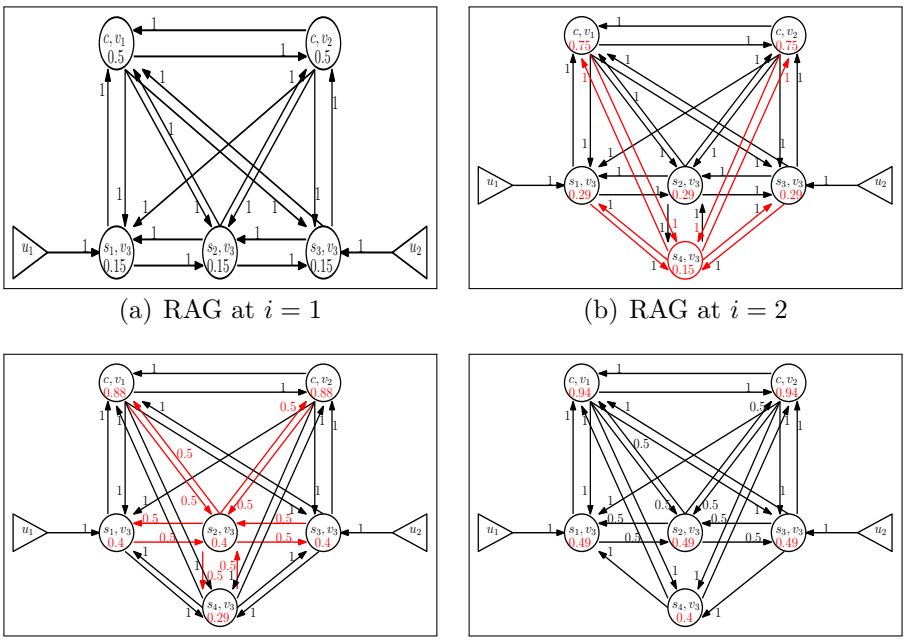
The visualization of the RAGs is illustrated in Figure 3.4, where each asset-vulnerability node is labelled with the potentiality. The arcs are labelled with the accessibility. An arc (n_1, n_2) is drawn if $g^i_{(n_1,n_2)} \neq 0$. The nodes (c, v_1),(c, v_2) correspond to the same asset, and so the accessibility between them is always equal to 1. The red potentialities and links correspond to a change compared to the previous time slot. The nodes u_1, u_2 correspond to the hosts 1 and 2, which play the role of the system access points and they are drawn as triangles.

According to Figure 3.4(a), at $i = 1$ there are 5 asset-vulnerability nodes drawn as circles, and refereed by $(c, v_1), (c, v_2), (s_1, v_3)$, (s_2, v_3) and (s_3, v_3) (see). The corresponding initial potentialities p_w are derived from Table 3.1 which describes the vulnerabilities of the assets and their associated exploitation and impact.

At $i = 2$, the switch s_4 whose associated vulnerability is v_3 as well as the arcs linking (s_4, v_3) with the other nodes are added in Figure 3.4(b). The potentiality of the nodes $(c, v_1), (c, v_2), (s_1, v_3)$, (s_2, v_3) and (s_3, v_3) increases according to function (3.1). Since (s_4, v_3) appears only at $i = 2$, at this time slot the node is labelled by the initial potentiality of v_3 which is 0.155 (see

Table 3.1).

At $i = 3$ the accessibility between (s_2, v_3) and the other assets decreases from 1 to 0.5 according to Figure 3.4(c). Finally at $i = 4$, the arcs $((s_4, v_3), (s_2, v_3))$, $((s_2, v_3), (s_4, v_3))$, $((s_1, v_3), (s_4, v_3))$ and $((s_4, v_3), (s_3, v_3))$ are deleted (see Figure 3.4(d)). We evaluate in the following the risk of this SDN system at each time slot.



(a) RAG at $i = 1$ (b) RAG at $i = 2$

(c) RAG at $i = 3$ (d) RAG at $i = 4$

Figure 3.4: SDN Risk Assessment Graphs ($I = 4$)

3.4.2 Risk Evaluation

We use the RAGs constructed in Section 3.4.1 by Algorithm 2 in order to evaluate our security metrics. The results are presented in Figures 3.5 and 3.6.

Figure 3.5 shows the variation of the nodes risk in function of time. Let us analyse the risks node by node. We see that (c, v_1) node risk is higher than the (c, v_2) one for each time slot. In fact, for the access points u_1 and u_2 the propagated potentiality to (c, v_1) or (c, v_2) has the same value. The factor making the (c, v_1) risk higher than the (c, v_2) risk is actually the impact (6.9 for (c, v_1) comapred to 2.9 for (c, v_2)).

Having the same values of propagated potentiality and the same values of impact for all time slot, the risk of (s_1, v_3) and (s_3, v_3) is the same for all time slot, as seen in Figure 3.5.

The node (s_2, v_3) has a smaller risk than (s_1, v_3) and (s_3, v_3) for each time slot, even if they have the same values of exploitation p_w and impact I_w. This is explained by the fact that the intruder should pass by (s_1, v_3) (if it is u_1) or by (s_3, v_3) (if it is u_2) in order to reach (s_2, v_3). Therefore, the difficulty of propagation increases for the intruder. Consequently, the risk of the node (s_2, v_3) decreases.

The risk of the node (s_4, v_3) at $i = 1$ is equal to 0 because the switch s_4 doesn't exist at this time slot and appears only at $i = 2$. At $i = 3$, the risk of the node (s_4, v_3) becomes bigger than the risk of (s_2, v_3) even if the potentiality of (s_2, v_3) is higher than the one of (s_4, v_3) at this time slot $(0.4 > 0.29)$. Actually, the accessibilities between s_2 and all the other assets, as well as those between all the other assets and s_2 decrease at $i = 3$ to become 0.5. This implies a higher propagated risk to (s_2, v_3).

Finally, Figure 3.6 shows that the global risk of the system is increasing over the time. This arises because all the node risks are increasing in function of time.

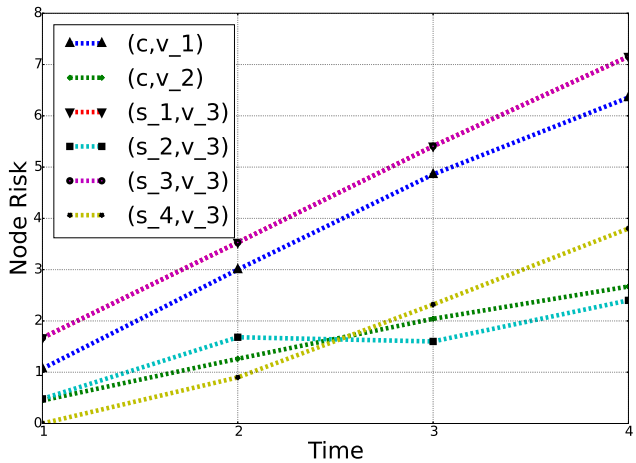


Figure 3.5: Node Risk in Function of Time

3.5 Simulations

We randomly generated systems with a large number of nodes. The aim is to show, for large random systems, the sensitivity of the mean global risk $\sum_{i \in I} R^i$ to the number of nodes, the convergence speed of the potentialities, the topology and the accessibilities. The experiments have been conducted on a computer equipped with an 2x Intel(R) Xeon(R) CPU E5-2650 v2 @

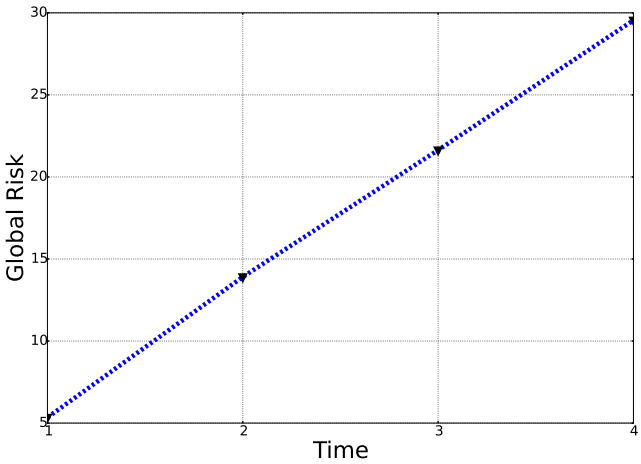


Figure 3.6: Global Risk in Function of Time

2.60GHz machine with 128Go of RAM, running under Linux. We used python 2.7 as programming language and Networkx [16] as a graph library.

3.5.1 Random systems generation

Our random systems are generated as follows. We first set $I = \{1, \ldots, 12\}$. The nodes, the arcs and the parameters of the RAGs for each $i \in I$ are configured as follows.

1) The sets S_i and T_i are generated as follows. The potentiality of the nodes in T_i are computed using equation (3.2). The parameter p_t is randomly generated using a continuous uniform distribution $U(0, 1)$, and the value of α_t varies in the set $\{0.1, 0.2, \ldots, 1\}$. We also set $\alpha_t = \alpha = cst$ for all $t \in T_i$ which implies that the potentiality speed of convergence to one is the same for all the asset-vulnerability nodes.

2) For each time slot in I, two specific subsets of arcs are randomly constructed; the arcs induced by the nodes of T_i, denoted by $A_i(T_i)$, and those connecting the nodes of S_i with those of T_i, denoted by $A_i(S_i, T_i)$. More specifically:

   a) The set $A_i(T_i)$ is randomly generated using Erdös-Renyi graphs [62], in such a way that the sub-graph induced by the nodes of T_i is an Erdös-Renyi random graph of parameters T_i and p. This means that the graph is constructed by randomly connecting $|T_i|$ nodes, while each arc is included with probability p independent from every other arc. We vary $p \in \{0.1, 0.2, \ldots, 0.9\}$.

b) The arcs $A_i(S_i, T_i)$ are constructed by connecting each $s \in S_i$ to a random number of node in T_i.

3) The weights of the arcs are calculated based on equation (3.4). This requires the accessibilities as a parameter, which is simulated as an increasing function of time for these experiments and computed using the equation (3.14) for all $i \in I$ and $v_1, v_2 \in V$:

$$g^i_{(v_1,v_2)} = a_{(v_1,v_2)} + (1 - a_{(v_1,v_2)})\frac{\beta(i-1)}{i}. \tag{3.14}$$

Here $a_{(v_1,v_2)}$ is the accessibility of (t_1, t_2) at the initial state of the system $(i = 1)$. We randomly generate $a_{(v_1,v_2)}$ using a continuous uniform distribution $U(0, 1)$. The parameter β controls how fast the accessibility tends to 1. We vary β in the set $\{0.1, 0.2, \ldots, 1\}$, and we fix the same value for all the arcs.

In the following, we focus on the sensitivity of the mean global risk to the parameters $|V_i|$, p,β and α. Recall that $|V_i|$ is the number of nodes in the RAG at time i. The parameter p gives an indication of the density of the links in the system topology. The speed of convergence of the accessibility is given by β, and the one of the potentiality is given by α

3.5.2 Impact of the number of nodes

Let us now investigate the sensitivity of the mean global risk to the number of nodes. We set $\alpha = \beta = p = 0.5$, and $|S_i| = \frac{1}{2}|T_i|$. We vary $|V_i|$ in $[150, \ldots, 1500]$. The results plotted in Figure 3.7 show a quasi-exponential growth of the mean global risk with the number of nodes.

3.5.3 Impact of the topology and the accessibility changes p and β

Now, we vary the parameters p and β as seen in Figure 3.8. We observe that, for a fixed β, a variation of p from 0.1 to 0.9 implies an increase of the mean global risk by nearly 1000. On the other hand, for p fixed a variation of β from 0.1 to 0.9 yields an increase of the mean global risk by nearly 4000. We can deduce that the parameter β has more influence in the mean global risk than parameter p. In other words, a big change in the accessibilities may have more impact in the system risk than a brutal change in the topology, for this set of random systems.

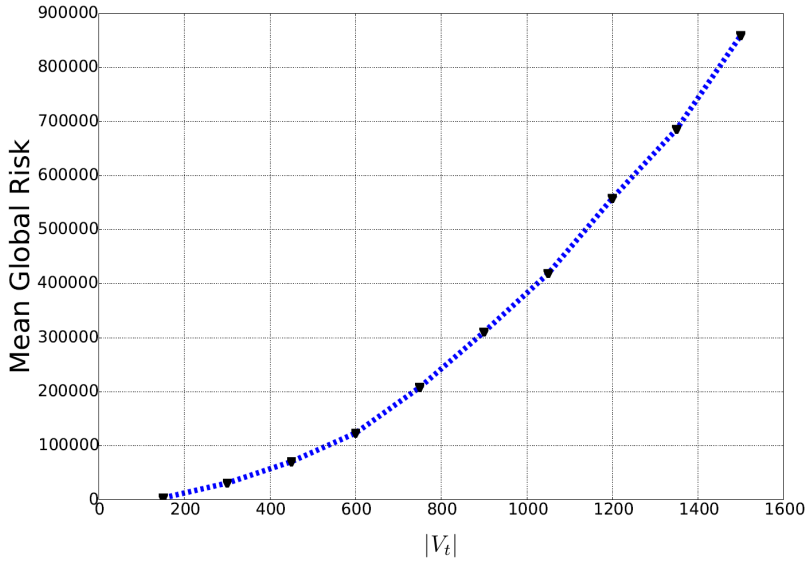


Figure 3.7: Mean Global Risk in Function of the Number of Nodes

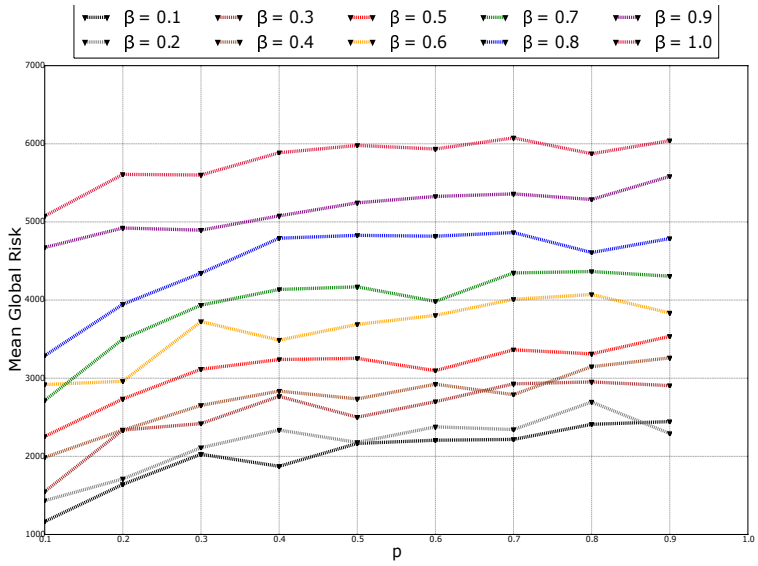


Figure 3.8: Impact of the topology and the accessibility convergence speed (p and β)

3.5.4 Impact of the potentiality convergence speed α

We study the mean global risk evolution in function of the parameter α. We set $|T_i| = 200, |S_i| = 20$ and $\beta = p = 0.5$. The variation of the mean global risk is plotted in Figure 3.9. This risk increases with the increase of α until $\alpha \leq 0.7$. When $\alpha = 0.8, 0.9$, the data are perturbed, and the value of the

mean global risk is slightly reduced.

Actually, this reduction is caused by of the topology change. In that case the probability of existence of a topological link p is constant ($p = 0.5$), but the links remain uncertain, and the realization of the random Erdös-Renyi sub-graph could generate a topology which prevents intruders to have higher propagation in the system.

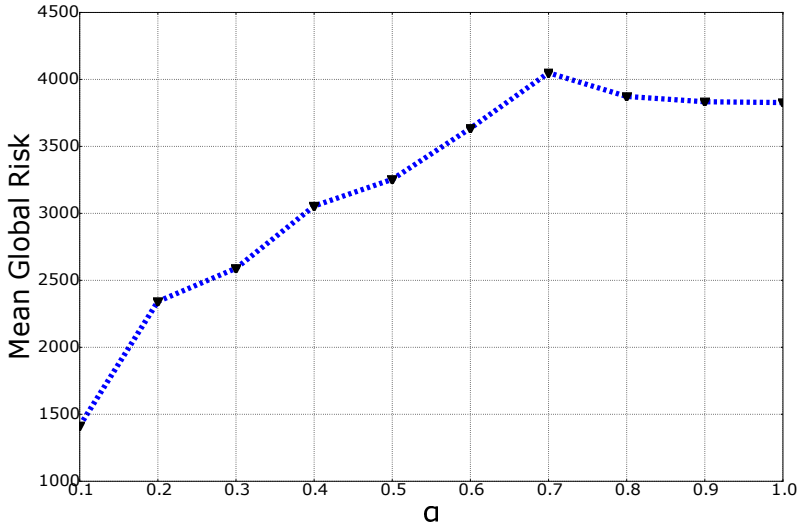


Figure 3.9: Impact of the potentiality convergence speed α

3.6 Concluding remarks

In this chapter, a new risk assessment framework has been proposed. We have introduced the RAGs model which captures the topological information, including the assets of the system, the accessibilities between them, and the vulnerabilities associated with each asset, as well as the way these elements vary over the time.

A risk evaluation approach has been provided based on the propagation of the attackers through the most likely paths. We have defined three security metrics namely the propagated risk, the node risk, and the global risk. Our approach is illustrated by a SDN case study. Several simulations on random systems have been conducted to show the sensitivity of our metrics relatively to the size of the system, the vulnerability convergence properties, the topology and the accessibilities.

The framework we have presented could identify in which time slots the system is not secured. If the risk exceeds a given threshold, an alert could then be sent to start control actions which can consist in the deployment of countermeasures on some system assets in order to reduce the global risk. However, while a countermeasure could reduce the system risks, its deployment might be expensive. This is the motivation to investigate bilevel programming in the next chapter, in order to define a risk treatment optimization model, giving optimal countermeasures placement.

Table 3.1: Topology and Vulnerability Data Basis Mapping

Assets	Vul.	Name : Summary	p_t	I_t
$Controller(C)$	v_1	CVE-2012-0368 : The administrative management interface on Cisco Wireless LAN Controller (WLC) devices with software 4.x, 5.x, 6.0, and 7.0 before 7.0.220.0, 7.1 before 7.1.91.0, and 7.2 before 7.2.103.0 allows remote attackers to cause a denial of service (device crash) via a malformed URL in an HTTP request, aka Bug ID CSCts81997.	0.5	6.9
	v_2	CVE-2013-1235 : Cisco Wireless LAN Controller (WLC) devices do not properly address the resource consumption of terminated TELNET sessions, which allows remote attackers to cause a denial of service (TELNET outage) by making many TELNET connections and improperly ending these connections, aka Bug ID CSCug35507.	0.5	2.9
$Switches1, 2(s_1, s_2, s_3)$	v_3	CVE-2013-5556 : The license-installation module on the Cisco Nexus 1000V switch 4.2(1)SV1(5.2b) and earlier for VMware vSphere, Cisco Nexus 1000V switch 5.2(1)SM1(5.1) for Microsoft Hyper-V, and Cisco Virtual Security Gateway 4.2(1)VSG1(1) for Nexus 1000V switches allows local users to gain privileges and execute arbitrary commands via crafted "install all iso" arguments, aka Bug ID CSCui21340.	0.155	10

Chapter 4

PCSP bilevel programming model, reformulations and optimality conditions

Contents

4.1 Definition and complexity **67**

4.1.1 Problem statement 67

4.1.2 PCSP complexity 68

4.2 Problem examples **70**

4.3 The Bi-level Model **72**

4.3.1 The follower problem 72

4.3.2 The bilevel formulation 73

4.4 Single-Level Reformulations **74**

4.4.1 Compact Single-Level Formulation 74

4.4.2 Path formulation by projection 75

4.5 Optimality conditions: dominance of countermeasures **78**

4.6 Concluding remarks **80**

This chapter is devoted to the introduction of the risk treatment optimization problem which is the final step of the risk management process. The problem we address is called the *Proactive Countermeasure Selection Problem* (PCSP). We will show that the PCSP is NP-Complete and formulate it as a bilevel programming model. Primal-dual optimality conditions will be used in order to convert the bilevel model into a compact single level formulation. We also give a second formulation by projecting the compact formulation

on a subset of variables. Moreover, we introduce some optimality condition inequalities that can improve the algorithmic aspect.

4.1 Definition and complexity

In this section, we state the PCSP and study its complexity.

4.1.1 Problem statement

An instance of the PCSP is given by a triplet (G, K, D) defined as:

- G = (V, A) is the Risk Assessment Graph defined in Chapter 3 defined as follows: the set of nodes V is partitioned into two specified subsets S and T where $V = S \cup T$ and $S \cap T = \emptyset$. A node in S represents an access point and a node in T an asset-vulnerability pair. The set of arcs A is defined such that for all $u, v \in V$, an arc from u to v exists if $v \notin S$ and its exploitation from u is possible. With each arc $(u, v) \in A$ it is associated a weight $w_{uv} \in \mathbb{R}_+$ representing the propagation difficulty of an attacker from u to v.
- $K = \{(t, k) : k \in K_t, t \in T\}$ is a set of available countermeasures such that K_t is the set of countermeasures associated with t. The placement of k on t has a positive cost $c_k^t \in \mathbb{R}+$, and yields an increase of a positive effect $\alpha_t^k \in \mathbb{R}_+$ in the weights of t-ongoing arcs.
- $D = (d_t^s)_{s \in S, t \in T} \in \mathbb{R}_+$ is a positive propagation difficulty threshold vector.

The PCSP consists in selecting a set of countermeasures $K' \subseteq K$ of minimum cost such that the *security constraints* are respected: for each $(s, t) \in S \times T$ the length of the shortest $s - t$ path after placing K' is at least d_s^t.

The decision version of the PCSP can be defined as: Given an instance (G, K, D) of PCSP and $\eta \in \mathbb{R}^+$, does there exists a set $K' \subseteq K$ such that $\sum_{(t,k) \in K'} c_t^k \leq \eta$, and for each $(s, t) \in S \times T$ the length of the shortest $s - t$ path after placing K' is at least d_s^t.

For reasons of simplicity, throughout this manuscript we can graphically represent the PCSP instances (G, K, D). The nodes of the graph G will be represented by circles. The countermeasures will be represented by squares in which we describe the effect flowed by the cost and thresholds will be indicated

by triangles in which we put the $s-t$ threshold. By default, we will assume that for a given $t \in T$, the threshold d_s^t is the same for each $s \in S$.

For the rest of the manuscript, we will use the following notations. We will denote by $L_G(P)$ the length of a path P in G. For each $s \in S$ and $t \in T$, $P_{s,t}$ will denote the set of all $s-t$ paths and $P^*_{s,t}$ will denote the $s-t$ shortest path. A path $P \in P_{s,t}$ is said to be *unsecured* if $L_G(P) < d_s^t$. The couple (s,t) such that $s \in S$, $t \in T$ is called *an attack*. The set $\Gamma = \{(s,t) \in S \times T : \exists P \in P_{st}, L_G(P) < d_s^t\}$ refers to the *unsecured attacks*. Each $s-t$ path such that $(s,t) \in \Gamma$ is called a *potential unsecured path*. Given H a subgraph of G, $K(H) = \{(t,k) : t \in H \cap T, k \in K_t\}$ will denote the set of countermeasures induced by the subgraph H. Let $L \subseteq K$, we denote by G_L the graph G whose edges $(i,j) \in A$ have weights $w_{ij} + \sum_{k \in K_j} \alpha_j^k$. Let $\bar{x} \in \mathbb{R}^K$, we denote by $G[\bar{x}]$ the graph G whose weights are $\bar{w_{ij}} = w_{ij} + \sum_{(j,k) \in K_j} \bar{x}_j^k \alpha_j^k$. Let $v \in T$, we define the set $ST(v) = \{(s,t) \in \Gamma : \exists P \in P_{s,t}, v \in P\}$ as the set of unsecured attacks (s,t) such that there exists a potential unsecured $s-t$ path P that contains v. The set Γ can be obtained by computing the $st-$ shortest paths. Thus, we have the following.

Proposition 4.1 The set of unsecured attacks $\Gamma = \{(s,t) \in S \times T : \exists P \in P_{st}, L_G(P) < d_s^t\}$ can be computed in polynomial time.

Remark 4.2 To verify if a subset of countermeasures is a solution of the PCSP, it suffices to check that this solution respects the security constraints for each $(s,t) \in \Gamma$ instead of verifying them for each $(s,t) \in S \times T$.

Remark 4.3 In order to guarantee the existence of at least one solution, we will assume that for each $s \in S$ and $t \in T$ the length of the $s-t$ shortest path in G_K is at least d_s^t. This means that there exists at least one solution of PCSP which consists in placing all countermeasures.

4.1.2 PCSP complexity

Now, we discuss the complexity of the PCSP. We first prove that the problem is NP-Complete even for only one access point, only one non zero threshold and the same countermeasure per asset-vulnerability node. Then, we show that PCSP is NP-Complete even if G is reduced to one edge: $V = \{s,t\}$ and $A = (s,t)$.

Theorem 4.4 *The PCSP is NP-Complete even for only one access point, only one non zero threshold and the same countermeasure per asset-vulnerability node.*

Proof. First, it is easy to see that the PCSP is in NP. In fact, given a set $K' \subseteq K$ we can in polynomial time verify that K' is a solution of PCSP. We compute a shortest $s-t$ path for each $(s,t) \in S \times T$. Then , if for all $(s,t) \in S \times T$ the length of the shortest $s-t$ path $L_{G_{K'}}(P^*_{st}) \geq d^t_s$, we deduce that K' is a solution of PCSP, if not K' is not a solution.

Let us now reduce the Minimum Vertex Blocker to Short Paths Problem (MVBP) to the PCSP. The MVBP can be stated as follows: given a directed graph $G' = (V', A')$, two nodes $s, t \in V'$, the length $l_{ij} \in \mathbb{R}^+$ of each arc $ij \in A'$, and two integer d and q, the problem consists in finding a subset $V'' \subseteq V'$ such that $\mid V'' \mid \leq q$ and the shortest path from s to t in $G' \setminus V''$ is of length at least d. This problem is NP-Complete [47].

We polynomially construct an instance of the PCSP as follows. We choose $G = G'$ where $S = \{s\}$ and $T = V' \setminus \{s\}$. For all $t \in T$, we fix $K_t = \{k_\infty\}$ where k_∞ is a countermeasure of unit cost $c = 1$ and infinite effect $\alpha = +\infty$ chosen to be the same for all asset-vulnerability nodes. We set $d^t_s = d$, and $d_{s,v} = 0$ for all $v \in T \setminus \{t\}$. Remark that the placement of the countermeasure k_∞ on a node t is the same as deleting t, since the effect of the ongoing arcs of t becomes infinite.

Now consider a solution V'' of MVBP. We have $\mid V'' \mid \leq q$ and $L_{G' \setminus V''}(P^*_{st}) \leq d$. Let $K' = \{(v, k_\infty) : v \in V''\}$. We will show that K' is a solution of PCSP. Remark first that since $d_{s,t} = d$, and $d_{s,v} = 0$ for all $v \in T \setminus \{t\}$, all what we need to check is that the length of the shortest $s-t$ path in $G_{K'}$ exceeds the threshold. As V'' is a solution of MVBP, we have $L_{G' \setminus V''}(P^*_{st}) \geq d$. On the other hand, $d^t_s = d$ and $G' \setminus V'' = G_{K'}$ since removing V'' is nothing but installing K'. Therefore, $L_{G_{K'}}(P^*_{st}) \geq d^t_s$. In addition, since k_∞ has a unit cost, $|V''|$ is equal to the cost of placement of the countermeasures in G. Consequently, $\mid K' \mid = \mid V'' \mid \leq q = \eta$ which together with the fact that $L_{G_{K'}}(P^*_{st}) \geq d^t_s$, implies that K' is a solution of PCSP.

Conversely, consider a solution $K' = \{(v, k_\infty) : v \in T' \subseteq T\}$ of PCSP. Since $d^t_s = d$, $G' \setminus V'' = G_{K'}$ and $\mid K' \mid = \mid V'' \mid \leq q = \eta$, it is easy to see that $V'' = \{v : (v, k_\infty) \in K'\}$ is a solution of MVBP, which completes the proof. □

Theorem 4.5 *The PCSP is NP-Complete even if G is reduced to one edge: $V = \{s, t\}$ and $A = \{(s, t)\}$.*

Proof. The reduction is from the knapsack problem which can be defined as: Given n items with a profit p_i and a weight w_i for every item i, an integer q and a scalar W, the goal is to select a subset of items such that $\sum_{i \in I} p_i \geq q$ and $\sum_{i \in I} w_i \leq W$.

As shown in the proof of theorem 4.4, the PCSP is in NP. Now, let us give a polynomial reduction of the knapsack problem to PCSP by setting $G = (V, A)$ where $V = \{s, t\}$, $A = \{(s, t)\}$ and $w_{st} = 0$. We also fix $d_s^t = q$, $K = \{(t, k_i) : i \in \{1, \ldots, n\}, \alpha_t^{k_i} = p_i, c_t^{k_i} = w_i\}$ and $\eta = W$.

Now suppose that I' is a solution of the knapsack problem which means that $\sum_{i \in I} p_i \geq q$ and $\sum_{i \in I} w_i \leq W$. As $d_s^t = q$ and $\eta = W$, the countermeasures set $K' = \{(t, k_i) : i \in I', \alpha_t^{k_i} = p_i, c_t^{k_i} = w_i\}$ is a solution of PCSP. Conversely, if K' is a solution of PCSP, it is clear that $I' = \{i : k_i \in K'\}$ is a solution of the knapsack problem since $d_s^t = q$ and $\eta = W$. □

4.2 Problem examples

We present now some simple instances of the PCSP and their associated optimal solutions. Examples 4.6 and 4.7 show that it is not always true that if for an instance a countermeasure has a better cost and effect than another, then it will be chosen in the optimal solution.

Example 4.6 Consider the instance presented in Figure 4.1. The graph G is a path graph containing only one access point s and the weights of the graph are equal to zero. with each asset-vulnerability node 1, 2 and 3 is associated one countermeasure. The thresholds are $d_s^1 = d_s^2 = 0$ and $d_s^3 = 1$ which implies that there is only one attack $\Gamma = \{(s, 3)\}$. The optimal solution of that instance consists in placing only the countermeasure $(3, k_3)$.

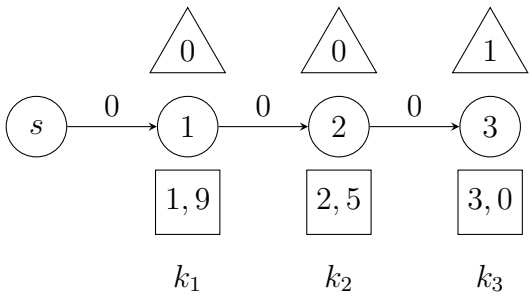


Figure 4.1: A first PCSP example

We can see that the countermeasure $(3, k_3)$ has the best cost and the best effect compared to the other countermeasures. This can be a reason for choosing $(3, k_3)$ in the optimal solution, but this is not always the case as we can see in the next example.

Example 4.7 In the instance represented in Figure 4.2, $(3, k_3)$ is again the best countermeasure in terms of cost and effect. However, The optimal solution of that instance consists in placing only the countermeasure $(2, k_2)$ which allows us to secure the $s - t$ path with the best cost.

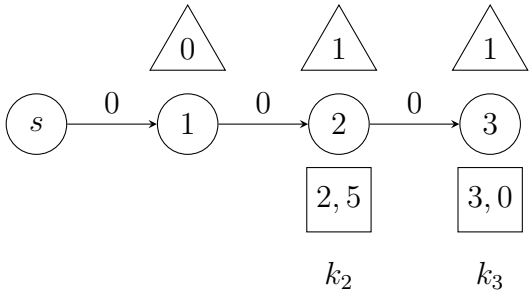


Figure 4.2: A second PCSP example

Example 4.8 Let us now consider the instance presented in Figure 4.3 which has two access points s and s' and one attack $\Gamma = \{(s, 3)\}$. In order to secure the graph with the best cost, the optimal solution consists in placing the countermeasures $(2, k_2)$ and $(3, k_3)$.

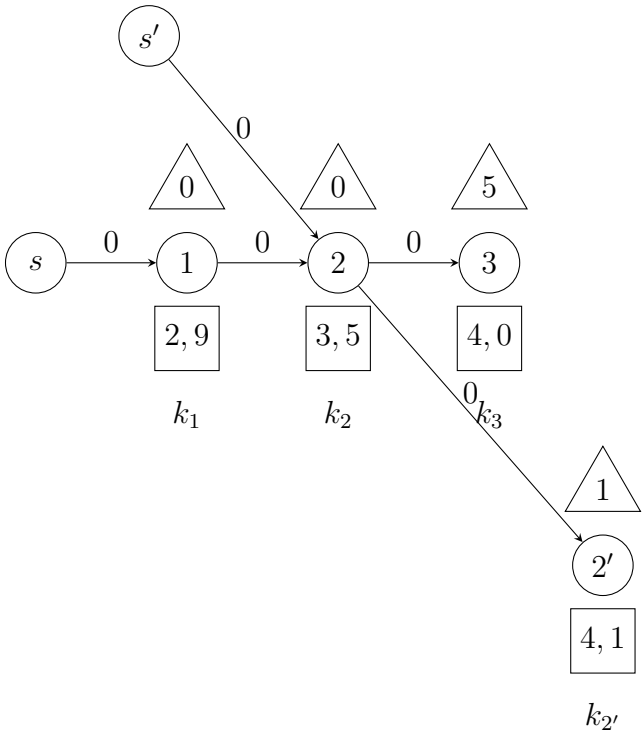


Figure 4.3: A third PCSP example

Example 4.9 Let us now consider again the instance presented in Figure 4.3. We have two unsecured attacks $\Gamma = \{(s,3),(s',2')\}$. In order to secure the graph with the best cost, the optimal solution consists in placing the countermeasures $(1,k_2)$ of effect 3 and cost 2 and $(2,k_2)$.

4.3 The Bi-level Model

In this section, we formulate the PCSP as a bilevel problem in which the leader controls the countermeasure deployment, and forces the shortest paths between each access point $s \in S$, and each asset-vulnerability node $t \in T$ to be at least d_s^t. On the other hand, several followers will play the role of the attackers. For each $s \in S$, and $t \in T$ each follower will compute the $s-t$ shortest path after the leader acts (countermeasures placement). That is the *st-follower problem* denoted by $st-F$.

4.3.1 The follower problem

Let x_t^k, $(t,k) \in K$ be the binary variable used to indicate if the countermeasure k is placed on the node t or not. Let $s \in S$ and $t \in T$. Each st-follower problem aims at computing the $s-t$ shortest path which is its most likely path. The weight of an arc $ij \in A$ after applying a countermeasure k in the node j is $w_{ij}(x) = w_{ij} + \alpha_j^k x_j^k$. Now, let $ij \in A$, we define z_{ij}^{st} as a binary variable indicating whether or not the arc ij belongs to the $s-t$ shortest path. The length of the $s-t$ shortest path is then given by $\sum\limits_{ij \in A} w_{ij}(x) z_{ij}^{st}$.

The st-follower problem is equivalent to the following $0-1$ program:

$$
(\text{st-}F)\begin{cases}
\text{Min} \sum\limits_{ij \in A} (w_{ij} + \sum\limits_{k \in K_j} \alpha_j^k x_j^k) z_{ij}^{st} & \\
\sum\limits_{j \in \Gamma^+(i)} z_{ij}^{st} - \sum\limits_{j \in \Gamma^-(i)} z_{ji}^{st} = \begin{cases} 1 & \text{if } i = s \\ 0 & \text{if } i \notin \{s,t\} \\ -1 & \text{if } i = t \end{cases} & \forall i \in V, \\
z_{ij}^{st} \in \{0,1\} & \forall ij \in A_t.
\end{cases}
$$

As it is well know, the constraint matrix of the shortest path formulation is totally unimodular which guarantees the integrality of the optimal solution of its linear relaxation given by

$$(\text{st-}F)\begin{cases}\text{Min} \sum\limits_{ij\in A}(w_{ij}+\sum\limits_{k\in K_j}\alpha_j^k x_j^k)z_{ij}^{st} \\ \sum\limits_{j\in\Gamma^+(i)} z_{ij}^{st}-\sum\limits_{j\in\Gamma^-(i)} z_{ji}^{st}=\begin{cases}1 & \text{if } i=s\\ 0 & \text{if } i\notin\{s,t\}\\ -1 & \text{if } i=t\end{cases} & \forall i\in V,\\ z_{ij}^{st}\geq 0 & \forall ij\in A.\end{cases}$$

The dual of $st-$F for all $t\in I, s\in S, t\in T$ is

$$(\text{st-}FD)\begin{cases}\text{Max } \lambda_t^{st}-\lambda_s^{st} \\ \lambda_j^{st}-\lambda_i^{st}\leq w_{ij}+\sum\limits_{k\in Kj}\alpha_j^j x_j^k & \forall ij\in A,\\ \lambda_i^{st} \text{ free} & \forall i\in V.\end{cases}$$

In what follows we give the bilevel formulation or the leader formulation of PCSP.

4.3.2 The bilevel formulation

The leader controls the countermeasure deployment respecting the security constraints: given the most likely paths thresholds d_s^t for each $s\in S$ and $t\in T$, the leader forces the shortest paths returned by the st-followers to be at least d_s^t. The objective function is to minimize the total cost of the countermeasure deployment. The PCSP is then equivalent to the following bilevel program:

$$\begin{array}{ll}\text{Min} \sum\limits_{(t,k)\in K} c_t^k x_t^k & \\ \sum\limits_{ij\in A}(w_{ij}+\sum\limits_{(j,k)\in K_j}\alpha_t^k x_j^k)z_{ij}^{st}\geq d_s^t, & \forall s\in S, t\in T,\\ \forall st-F\begin{cases}\text{Min} \sum\limits_{ij\in A}(w_{ij}+\sum\limits_{(j,k)\in K_j}\alpha_j^k x_j^k)z_{ij}^{st}, \\ \sum\limits_{u\in\Gamma^+(v)} z_{vu}^{st}-\sum\limits_{u\in\Gamma^-(v)} z_{uv}^{st}=\begin{cases}1 & \text{if } v=s\\ 0 & \text{if } v\notin\{s,t\}\\ -1 & \text{if } v=t\end{cases} & \forall v\in V,\\ z_{ij}^{st}\in\{0,1\} & \forall ij\in A.\end{cases}\\ x_t^k\in\{0,1\} & \forall (t,k)\in K.\end{array}$$

4.4 Single-Level Reformulations

In this section, we present two different single-level reformulations of our PCSP bilevel model. The first one is obtained by a well know technique [52] using primal-dual optimality conditions to obtain a single level compact formulation of the bilevel problem. The second one, called *path formulation*, is a non compact formulation and obtained by projection of the compact one on the x variables.

4.4.1 Compact Single-Level Formulation

According to the weak and strong duality theorems, every LP problem can be replaced with the primal feasibility constraints, the dual feasibility constraints, and the weak duality equation. Hence, by replacing the follower with its primal-dual optimality conditions, we obtain a single level formulation of the bilevel PCSP model. Note that the primal-dual transformation holds because the linear relaxation of the shortest path problem is integral. The compact single level reformulation is then equivalent to the following program where constraints (4.1) - (4.6) are defined for all $s \in S, t \in T$:

$$\text{Min} \sum_{(t,k)\in K} c_t^k x_t^k$$

$$\lambda_t^{st} - \lambda_s^{st} \geq d_s^t, \tag{4.1}$$

$$\sum_{w\in\Gamma^+(v)} z_{vw}^{st} - \sum_{u\in\Gamma^-(v)} z_{uv}^{st} = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{if } v \notin \{s,t\} \\ -1 & \text{if } v = t \end{cases} \quad \forall v \in V, \tag{4.2}$$

$$\lambda_v^{st} - \lambda_u^{st} \leq w_{uv} + \sum_{k\in K_v} \alpha_v^k x_v^k \quad \forall uv \in A, \tag{4.3}$$

$$\sum_{uv\in A} (w_{uv} z_{uv}^{st} + \sum_{k\in K_v} \alpha_v^k x_v^k z_{uv}^{st}) = \lambda_t^{st} - \lambda_s^{st}, \tag{4.4}$$

$$x_t^k, z_{uv}^{st} \in \{0,1\} \quad \forall uv \in A, (t,k) \in K, \tag{4.5}$$

$$\lambda_v^{st} \text{ free} \quad \forall v \in V. \tag{4.6}$$

After obtaining a single level reformulation, we linearise the term $x_v^k z_{uv}^{st}$. To this end, we introduce a binary variable $y_{k,uv}^{st}$ that takes the value 1 if x_v^k and z_{uv}^{st} are both equal to 1, and 0 otherwise. These operations yield the following compact formulation PCSP1 whose constraints (4.7) - (4.16) are defined for all $s \in S, t \in T$:

$$\text{PCSP1: Min} \sum_{(t,k)\in K} c_t^k x_t^k$$

$$\lambda_t^{st} - \lambda_s^{st} \geq d_s^t, \tag{4.7}$$

$$\sum_{w\in\Gamma^+(v)} z_{vw}^{st} - \sum_{u\in\Gamma^-(v)} z_{uv}^{st} = \begin{cases} 1 & \text{if } v = s \\ 0 & \text{if } v \notin \{s,t\} \\ -1 & \text{if } v = t \end{cases} \quad \forall v \in V, \tag{4.8}$$

$$\lambda_v^{st} - \lambda_u^{st} \leq w_{uv} + \sum_{k\in K_v} \alpha_v^k x_v^k \quad \forall uv \in A, \tag{4.9}$$

$$\sum_{uv\in A} (w_{uv} z_{uv}^{st} + \sum_{k\in K_v} \alpha_v^k y_{k,uv}^{st}) = \lambda_t^{st} - \lambda_s^{st}, \tag{4.10}$$

$$y_{k,uv}^{st} \leq 1/2(x_v^k + z_{uv}^{st}) \quad \forall uv \in A, k \in K_v, , \tag{4.11}$$

$$y_{k,uv}^{st} \geq x_v^k + z_{vu}^{st} - 1 \quad \forall uv \in A, k \in K_v, \tag{4.12}$$

$$x_t^k \in \{0,1\} \quad \forall (t,k) \in K, \tag{4.13}$$

$$z_{uv}^{st} \in \{0,1\} \quad \forall uv \in A, \tag{4.14}$$

$$y_{k,uv}^{st} \in \{0,1\} \quad \forall uv \in A, t \in K_v, \tag{4.15}$$

$$\lambda_v^{st} \text{ free} \quad \forall v \in V. \tag{4.16}$$

We can see that the compact formulation PCSP1 has a polynomial number of variables and constraints. The size of variables is

$$\begin{aligned} &\mid K \mid + \mid S \mid\mid T \mid + \mid A \mid\mid K \mid + \mid S \mid\mid T \mid\mid A \mid + \mid S \mid\mid T \mid\mid V \mid \\ &= \mid K \mid (\mid A \mid +1) + \mid S \mid\mid T \mid (\mid A \mid + \mid V \mid +1). \end{aligned} \tag{4.17}$$

On the other hand, the size of constraints is

$$\mid S \mid\mid T \mid (\mid V \mid + \mid A \mid +2 \mid K \mid\mid A \mid). \tag{4.18}$$

In the next section we introduce an alternative formulation, the path formulation.

4.4.2 Path formulation by projection

A second single-level reformulation of the bilevel model can be obtained by projection of the compact formulation PCSP1 onto the x of variables. Let PCSP2 be this formulation, we have the following proposition.

Proposition 4.10 The projection of formulation PCSP1 on variables x is

$$PCSP2:\ Min \sum_{(t,k)\in K} c_t^k x_t^k$$

$$\sum_{ij\in P} \sum_{k\in K_j} \alpha_j^k x_j^k \geq d_s^t - L_G(P) \quad \forall s \in S, t \in T, P \in P_{s,t}, \tag{4.19}$$

$$x_t^k \in \{0,1\} \qquad \forall (t,k) \in K. \tag{4.20}$$

Proof. It suffices to prove that for each feasible solution (x, z, y, λ) to the formulation $PCSP1$, x is a feasible solution to the formulation $PCSP2$, and for each feasible solution x to the PCSP2 there exists (z, y, λ) such that (x, z, y, λ) is a feasible solution to the PCSP1. Let $s \in S, t \in T$ and denote by $P_{st,i}^* = (v_1, v_2, \ldots, v_p)$ the shortest $s-t$ path where $n_0 = s$ and $n_p = t$.

Consider a solution (x_1, y, z, λ) a of the $PCSP1$ formulation, let x_2 be defined as $x_2 = x_1 = x$. By using inequalities (4.7) and (4.10), we obtain for all $s \in S, t \in T$, $\sum_{uv\in A} (w_{uv} z_{uv}^{st} + \sum_{k\in K_v} \alpha_v^k y_{k,uv}^{st}) \geq d_s^t$. Since

$$z_{uv}^{st} = \begin{cases} 1 & \text{if } uv \in P_{st,i}^* \\ 0 & \text{otherwise.} \end{cases} \quad \text{and } y_{k,uv}^{st} = x_v^k z_{uv}^{st},$$

by replacing z_{uv}^{st} by its value, we obtain for all $s \in S, t \in T$, $\sum_{uv\in P_{st}^*} (w_{uv} + \sum_{k\in K_v} \alpha_v^k x_v^k) \geq d_s^t$. This means that the length of the shortest $s-t$ path is greater than or equal to d_s^t. Therefore, the length of any path between s and t is greater than or equal to d_s^t and inequality (4.19) holds. Consequently, x is a feasible solution of PCSP2.

Conversely, consider a solution x_2 of $PCSP2$, set the decision variables (x_1, z, y, λ) of $PCSP1$ as follows, for all $s \in S, t \in T, k \in K, v \in V, uv \in A$:

- $x_1 = x_2 = x$,
- $z_{uv}^{st} = \begin{cases} 1 & \text{if } uv \in P_{st,i}^* \\ 0 & \text{otherwise.} \end{cases}$,
- $y_{k,uv}^{st} = x_v^k z_{uv}^{st}, and$
- $\begin{cases} \lambda_v^{st} = 0 & \text{if } v = s \\ \lambda_v^{st} = \underset{u\in\Gamma^-(v)}{Min} \{\lambda_v^{st} + w_{uv} + \sum_{k\in K_v} \alpha_v^k x_v^k\} & \text{otherwise.} \end{cases}$

It is obvious that constraints (4.13), (4.14), (4.14) and (4.6) are satisfied. Inequalities (4.11) and (4.12) are also satisfied because y is the product of two binary variables. Now, we will show that constraints (4.7)-(4.10) hold.

Let $s \in S, t \in T, k \in K, v \in V, uv \in A$.

We will first prove that constraints (4.8) are satisfied. As $z_{uv}^{st,i} = \begin{cases} 1 & \text{if } uv \in P_{st}^* \\ 0 & \text{otherwise.} \end{cases}$,

If $v = s$, $\sum_{w \in \Gamma^+(v)} z_{vw}^{st} - \sum_{u \in \Gamma^-(v)} z_{uv}^{st} = \sum_{w \in \Gamma^+(s)} z_{sw}^{st} - \sum_{u \in \Gamma^-(s)} z_{us}^{st} = z_{sv_1}^{st} - 0 = 1.$

If $v \neq s, t$ and $v \notin P_{st,i}^*$, $\sum_{w \in \Gamma^+(v)} z_{vw}^{st} - \sum_{u \in \Gamma^-(v)} z_{uv}^{st} = 0.$

If $v \neq s, t$ and $v \in P_{st,i}^*$, $\sum_{w \in \Gamma^+(v)} z_{vw}^{st} - \sum_{u \in \Gamma^-(v)} z_{uv}^{st} = z_{uv}^{st} - z_{vw}^{st} = 1 - 1 = 0.$

If $v = t$, $\sum_{w \in \Gamma^+(v)} z_{vw}^{st} - \sum_{u \in \Gamma^-(v)} z_{uv}^{st} = \sum_{w \in \Gamma^+(t)} z_{sw}^{st} - \sum_{u \in \Gamma^-(t)} z_{us}^{st} = 0 - z_{sv_p}^{st} = 1.$

It is easy to see that constraints (4.9) are satisfied since

$$\begin{aligned} \lambda_v^{st} &= \underset{u \in \Gamma^-(v)}{Min} \{\lambda_v^{st} + w_{uv} + \sum_{k \in K_v} \alpha_v^k x_v^k\}, \\ &\leq \lambda_u^{st} + w_{uv} + \sum_{k \in K_v} \alpha_v^k x_v^k, \\ \text{Then, } \lambda_v^{st} - \lambda_u^{st} &\leq w_{uv} + \sum_{k \in K_v} \alpha_v^k x_v^k. \end{aligned}$$

Remark now that the variables λ are equal to the weights of nodes used in Bellman algorithm [42] to find the shortest path length. This implies that λ_t^{st} is equal to the shortest $s - t$ length and hence constraints (4.10) are satisfied. Finally, security constraints ensure the validity of constraints (4.7). □

As PCSP2 is the projection of PCSP1 on the x variables and the objective functions of the two problems only depend on x, the following corollary holds.

Corollary 4.11 *The objective functions of PCSP1 and PCSP2 are equal.*

The path formulation PCSP2 could have an exponential number of constraints (4.19) since they are defined for every path between each access point and each asset-vulnerability node. These constraints are called the *security constraints*. They control all the paths from each $s \in S$ to each $t \in T$ and force their length to be greater than or equal to $d_{s,t}$. This ensure the security requirements, since if all the paths are of length at least $d_{s,t}$ then the shortest one is so, and vice versa. One can strengthen this formulation by studying the optimality properties of the problem.

4.5 Optimality conditions: dominance of countermeasures

In this Section, we introduce optimality conditions of the PCSP. These conditions can be modeled as inequalities which can be used when solving the problem in the preprocessing phase as it will be shown in Chapters 6. Let us first define the relation of dominance between countermeasures.

Definition 4.12 Let $(t,k),(v,l) \in K$. We say that (t,k) dominates (v,l) and write $(t,k) \succeq (v,l)$ if $c_t^k \leq c_v^l$ and $\alpha_t^k \geq \alpha_v^l$. The countermeasure (t,k) is called the dominant and (v,l) is the dominated countermeasure.

In other words, $(t,k) \succeq (v,l)$ if (t,k) is better than (v,l) in cost and in effect. Consider now two countermeasures (t,k_1) and (t,k_2) associated to the same asset-vulnerability node t such that $(t,k_1) \succeq (t,k_2)$. Theorem 4.13 shows that the dominance of countermeasures inside the same asset-vulnerability node implies that any optimal solution of PCSP will satisfy an inequality indicating that if the dominated countermeasure is chosen, then so is for the dominant one. In fact, if the dominated countermeasure is chosen but not the dominant one, this will break the optimality.

Theorem 4.13 *Let (G,K,D) be an instance of PCSP, let $t \in T$, and $k_1,k_2 \in K_t$, $k_1 \neq k_2$ such that $(t,k_1) \succeq (t,k_2)$. Then, any optimal solution of PCSP verifies*

$$x_t^{k_2} \leq x_t^{k_1} \tag{4.21}$$

Proof. We will prove the result by contradiction. Let $t \in T$, and $k_1,k_2 \in K_t$, $k_1 \neq k_2$. Suppose that there exists an optimal solution x^* such that $(t,k_1) \succeq (t,k_2)$ but $x_t^{*k_2} > x_t^{*k_1}$. We will show that it is possible to construct a solution $\tilde{x}$ cheaper then x^* and verifying $\tilde{x}_t^{k_2} \leq \tilde{x}_t^{k_1}$, which will end the proof.

Under the assumption that $x_t^{*k_2} > x_t^{*k_1}$, we have necessary $x_t^{*k_2} = 1$ and $x_t^{*k_1} = 0$, because otherwise, since x^* is a binary variable, we obtain either $0 < 0$ or $1 < 1$ which is not possible. Let $\tilde{x}$ be defined as

$$\tilde{x}_v^l = \begin{cases} 1 & \text{if } (v,l) = (t,k_1), \\ 0 & \text{if } (v,l) = (t,k_2), \\ x_v^{*l} & \text{otherwisse.} \end{cases}$$

Claim 4.14 *The incidence vector $\tilde{x}$ is a solution of PCSP.*

Proof. Let $(s,t) \in \Gamma$. We will show that for all $P \in P_{st}$, $\sum_{(v,l)\in K(P)} \tilde{x}_v^l \geq d_s^t - L_G(P)$. We will distinguish two cases.

Case 1 $t \notin P$**:** by construction for all $(v,l) \in K(P) \setminus \{(t,k_1),(t,k_2)\}$ we have $x_v^{*l} = \tilde{x}_v^l$. Then, we can write

$$\begin{aligned}\sum_{(v,l)\in K(P)} \alpha_v^l \tilde{x}_v^l = &\sum_{(v,l)\in K(P)} \alpha_v^l x_v^{*l} \\ &\geq d_s^t - L_G(P), \ \textit{since } x^* \textit{ is a solution.}\end{aligned}$$

Case 2 $v \notin P$**:** By construction of $\tilde{x}$ we also have $x_t^{*k_1} + x_t^{*k_2} = \tilde{x}_t^{k_1} + \tilde{x}_t^{k_2}$, which implies that

$$\begin{aligned}\sum_{(v,l)\in K(P)} \alpha_v^l \tilde{x}_v^l &= \sum_{(v,l)\in K(P)\setminus\{(t,k_1),(t,k_2)\}} \alpha_v^l \tilde{x}_v^l + \tilde{x}_t^{k_1} + \tilde{x}_t^{k_2} \\ &= \sum_{(v,l)\in K(P)\setminus\{(t,k_1),(t,k_2)\}} \alpha_v^l x_v^{*l} + x_t^{*k_1} + x_t^{*k_2}, \\ &\geq d_s^t - L_G(P), \ \textit{since } x^* \textit{ is a solution.}\end{aligned}$$

We conclude that $\tilde{x}$ is a solution of PCSP which ends the proof of the claim.

Now, the cost of the solution $\tilde{x}$ is

$$\begin{aligned}\sum_{(v,l)\in K} c_v^l \tilde{x}_v^l &= \sum_{(v,l)\in K\setminus\{(t,k_1),(t,k_2)\}} c_v^l \tilde{x}_v^l + c_t^{k_1}\tilde{x}_t^{k_1} + c_t^{k_2}\tilde{x}_t^{k_2} \\ &= \sum_{(v,l)\in K\setminus\{(t,k_1),(t,k_2)\}} c_v^l x_v^{*l} + c_t^{k_2}\tilde{x}_t^{k_1} + c_t^{k_2}\tilde{x}_t^{k_2} \\ &= \sum_{(v,l)\in K\setminus\{(t,k_1),(t,k_2)\}} c_v^l x_v^{*l} + c_t^{k_1}, \text{since } \tilde{x}_t^{k_2} = 0.\end{aligned}$$

The one of the solution x^* is

$$\begin{aligned}\sum_{(v,l)\in K} c_v^l x_v^{*l} &= \sum_{(v,l)\in K\setminus\{(t,k_1),(t,k_2)\}} c_v^l x_v^{*l} + c_t^{k_1} x_t^{*k_1} + c_t^{k_2} x_t^{*k_2} \\ &= \sum_{(v,l)\in K\setminus\{(t,k_1),(t,k_2)\}} c_v^l x_v^{*l} + c_t^{k_2}, \text{as } x_t^{*k_2} = 1 \text{ and } x_t^{*k_1} = 0.\end{aligned}$$

Since $(t,k_2) \preceq (t,k_1)$, $c_t^{k_2} \geq c_t^{k_1}$. This implies that $\tilde{x}$ is cheaper than x^*.

□

Remark 4.15 The generalization of the optimality conditions as presented in Theorem 4.13, for countermeasures that are not associated to the same node,

is not longer true. That is to say if (t, k) and (v, l) are two countermeasures such that $(t, k) \succeq (v, l)$ and $t \neq v$, then it is not necessary true that any optimal solution of $PCSP$ verifies $x_t^k \geq x_v^l$. The instance represented by Figure 4.4 shows that claim. In fact, it is clear that $(3, k_3) \succeq (2, k_2)$. However, The optimal solution of that instance consists in placing the countermeasure $(2, k_2)$ which yields $x_2^{k_2} > x_3^{k_3}$ and proves the remark.

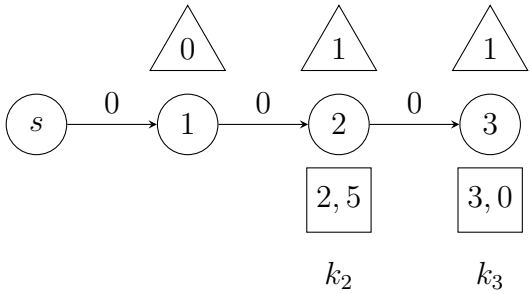


Figure 4.4: Dominance of countermeasures on a path

4.6 Concluding remarks

In this chapter, we have considered a bilevel model for the Proactive Countermeasure Selection Problem (PCSP). We have studied the complexity of the problem and proposed two single-level reformulations of the model. The first one, PCSP1, is compact and based on primal-dual optimality conditions. The second formulation, PCSP2, is a path formulation obtained by projection from PCSP1. We have also introduced optimality conditions for the problem that will be used during the preprocessing in order to improve the algorithmic aspect. Moreover, the compact formulation PCSP1 will be solved for several instances in Chapter 7 and will be compared to PCSP2. Another efficient method which can significantly strengthen the algorithmic aspect is the polyhedral approach. This will be the purpose of the next chapter.

Chapter 5

The PCSP: a polyhedral investigation

Contents

5.1 ILP formulation and the associated polytope . . . 82

5.2 Dimension of $PCSP(G, K, D)$ 83

5.2.1 Essential Countermeasures 83

5.2.2 Dimension 86

5.3 Facial investigation of basic inequalities 89

5.3.1 Trivial Inequalities 89

5.3.2 Security inequalities 90

5.4 Valid inequalities and facial aspect 93

5.4.1 Path Covering Inequalities (PCI) 94

5.4.2 Countermeasures Path Inequalities (CmPI) 97

5.4.3 Essential -by Subsets Removing- Countermeasures (ESRC) inequalities 100

5.5 Concluding remarks 106

In this chapter, we study the path formulation PCSP2 from a polyhedral point of view. We characterize the dimension and describe several classes of valid inequalities. We will also discuss when these inequalities define facets. As it will turn out, this approach will permit to strengthen the formulation and improve the resolution of the problem.

5.1 ILP formulation and the associated polytope

Let us recall the integer programming formulation PCSP2. We use a set of variables called *placement variables* indicating wether or not a countermeasure is placed. Let $x \in \mathbb{R}^{|K|}$ such that for each $(t,k) \in K$

$$x_t^k = \begin{cases} 1 & \text{if } (t,k) \text{ is placed,} \\ 0 & \text{otherwise.} \end{cases}$$

An instance of the PCSP problem corresponds to the triplet (G, K, D). Let $S(G, K, D)$ define the set of feasible solutions of the PCSP associated with the graph G, the countermeasures K and the thresholds D. Note that $S(G, J, D) \subseteq S(G, K, D)$ for all $J \subseteq K$. In fact each $s \in S(G, J, D)$ will satisfy the threshold vector D for G. A vector x^s induced by a solution s of $S(G, K, D)$ satisfies the following constraints:

$$\sum_{ij \in P} \sum_{k \in K_j} \alpha_j^k x_j^k \geq d_s^t - L_G(P) \quad \forall s \in S, t \in T, P \in P_{s,t}, \tag{5.1}$$

$$0 \leq x_t^k \quad \forall (t,k) \in K, \tag{5.2}$$

$$x_t^k \leq 1 \quad \forall (t,k) \in K. \tag{5.3}$$

Recall that inequalities (5.1) are security inequalities. They ensure for each access point $s \in S$ and each asset-vulnerability node $t \in T$, that the length of the shortest $s - t$ path is at least d_s^t. Inequalities (5.2) and (5.3) are *the trivial inequalities.*

The PCSP problem is equivalent to the following integer program.

$$min\{c^T x | x \in \{0,1\}^{|K|} : x \text{ satisfies } (5.1) - (5.3)\} \tag{5.4}$$

Theorem 5.1 *The linear relaxation of (5.4) can be solved in polynomial time.*

Proof. The complexity of the linear relaxation of (5.4) only depends on the one of the separation problem of inequalities (5.1) [76, 77]. Let us denote by $\bar{x}$ a solution to be separated, and let $(s,t) \in S \times T$. We compute a shortest path between s and t in $G[\bar{x}]$ whose arc weights are $\bar{w_{ij}} = w_{ij} + \sum_{(j,k) \in K_j} \bar{x}_j^k \alpha_j^k$. Then, if $L_{G[\bar{x}]}(P_{st}^*) \geq d_s^t$, every $s - t$ path has length greater than or equal

to d_s^t and no violated security inequality is detected. Otherwise, the security inequality associated with the shortest $s - t$ path is violated. Therefore, the separation of constraints (5.1) can be reduced to the calculation of $|S| \times |T|$ shortest paths in $G[\bar{x}]$. Consequently, since this can be done in polynomial time, so it is for the linear relaxation. □

Now, consider an instance (G, K, D) of PCSP. We denote by $PCSP(G, K, D)$, the polytope associated with the PCSP, that is the convex hull of the solutions of formulation (5.4) related to G,K and D, i.e.,

$$PCSP(G, K, D) = conv\{c^T x | x \in \{0, 1\}^{|K|} : \text{x satisfies } (5.1) - (5.3)\}.$$

In the following section, we study the dimension of $PCSP(G, K, D)$.

5.2 Dimension of $PCSP(G, K, D)$

In this section, we characterize the dimension of the polytope $PCSP(G, K, D)$. We first give some definitions which are necessary for the rest of the section.

5.2.1 Essential Countermeasures

We characterize the set of *essential countermeasures* of the polytope $PCSP(G, K, D)$. We suppose that $S(G, K, D) \neq \emptyset$.

Definition 5.2 A countermeasure $(t, k) \in K$ is said to be *essential* for $PCSP(G, K, D)$ if and only if the set $S(G, K \backslash \{(t, k)\}, D) = \emptyset$.

In other words, a countermeasure k is essential if the placement of all the countermeasures except k can not secure the network. We will denote by K^* the set of all essential countermeasures of $PCSP(G, K, D)$. We have $K^* = \{(t, k) : k \in K_t^*, t \in T\}$ such that K_t^* is the set of essential countermeasures associated with t. Hence,

$$x_t^k = 1 \quad \text{for all } (t, k) \in K^*. \tag{5.5}$$

Throughout the rest of the chapter, the first $|K| - |K^*|$ components of any incidence vector of size $|K|$ will be associated to the non-essential countermeasures. The remaining ones will correspond to the essential ones. Let $x \in \mathbb{R}^K$ be an incidence vector, then $(x_i)_{i=1,\ldots,|K|-|K^*|}$ will refer to the non-essential countermeasure decision variables and $(x_i)_{i=|K|-|K^*|+1,\ldots|K|}$ are the essential countermeasure one.

5.2.1.1 Characterization of essential countermeasures

Proposition 5.3 *Let $s_1, s_2 \subseteq K$ such that $s_1 \subseteq s_2$. If $s_1 \in S(G, K, D)$, then $s_2 \in S(G, K, D)$.*

Proof. Consider $s_1, s_2 \subseteq K$ such that $s_1 \subseteq s_2$. We suppose that $s_1 \in S(G, K, D)$. Let $s \in S$, $t \in T$ and $P \in P_{s,t}$, then

$$\sum_{(v,k)\in s_2} \alpha_v^k = \underbrace{\sum_{(v,k)\in s_1} \alpha_v^k}_{\geq d_s^t - L_G(P)} + \underbrace{\sum_{(v,k)\in s_2\setminus s_1} \alpha_v^k}_{\geq 0} \geq d_u^t - L_G(P)$$

Therefore, the incidence vector $x^{s_2} \in \{0,1\}^{|K|}$ satisfies (5.1)-(5.3). Consequently, $s_2 \in S(G, K, D)$. □

An equivalent definition of essential countermeasures is:

Proposition 5.4 A countermeasure $(t, k) \in K$ is essential for $PCSP(G, K, D)$ if and only if there are $s_0 \in S, t_0 \in T$, and $P_0 \in P_{s_0,t_0}$ such that:

$$\sum_{(v,l)\in K(P_0)\setminus\{(t,k)\}} \alpha_v^l < d_{s_0}^{t_0} - L_G(P_0).$$

Proof. By Definition 5.2, if $(t, k) \in K^*$, we have $S(G, K\setminus\{(t, k)\}, D) = \emptyset$. Then, for all $x \in R^{|K|-1}$, there are $s_0 \in S, t_0 \in T$ and $P_0 \in P_{s_0,t_0}$ such that

$$\sum_{(v,l)\in K(P_0)\setminus\{(t,k)\}} \sum \alpha_v^l x_v^l < d_{s_0}^{t_0} - L_G(P_0).$$

If we set $x_v^l = 1$ for all $(v, l) \in K\setminus\{(t, k)\}$, we obtain:

$$\sum_{(v,l)\in K(P_0)\setminus\{(t,k)\}} \alpha_v^l < d_{s_0}^{t_0} - L_G(P_0)$$

.

Conversely, let $(t, k) \in K$ and suppose there are $s_0 \in S, t_0 \in T$, and $P_0 \in P_{s_0,t_0}$ such that:

$$\sum_{(v,l)\in K(P_0)\setminus\{(t,k)\}} \alpha_v^l < d_{s_0}^{t_0} - L_G(P_0).$$

If (t, k) is not essential, then there exists a solution $\tilde{s} \in S(G, K, D)$ such that $\tilde{s} \subseteq K\setminus\{(t, k)\}$. Since $\tilde{s} \subseteq K\setminus\{(t, k)\}$ and $\tilde{s} \in S(G, K, D)$, by proposition 5.3 it is so for $K\setminus\{(t, k)\}$, and for all $s \in S, t \in T$ and $P \in P_{s,t}$, we have

$$\sum_{(v,l)\in K(P_0)\setminus\{(t,k)\}} \alpha_v^l \geq d_{s,t} - L_G(P).$$

A contradiction is obtained by taking $P = P_0$. □

By definition 5.4, a countermeasure $(v, l) \in K$ is essential if and only if there exists an $s-t$ path P such that all the available countermeasures for the nodes composing this path, except (v, l), cannot satisfy the security inequality. The following result shows that P can be (but not the only one) chosen over all the $s-t$ shortest paths $P^*_{s,t}$ such that $v \in P^*_{s,t}$. That is an $s-t$ path P containing v and minimizing $L_G(P) - d^{t_0}_{s_0}$ is minimum over the $s-t$ shortest paths $P^*_{s,t}$ such that $t \in P^*_{s,t}$ for all $(s, t) \in S \times T$.

Proposition 5.5 *$(v, l) \in K^*$ if and only if $\alpha^l_v > \min\limits_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$.*

Proof. Consider $(v, l) \in K$ such that $\alpha^l_v > \min\limits_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$. Let P^*_0 be the shortest $s_0 - t_0$. We have

$$\alpha^l_v > L_{G_K}(P^*_0) - d^{t_0}_{s_0} \Rightarrow \alpha^l_v > \sum_{(t,k)\in K(P_0)} \alpha^k_t + L_G(P_0) - d^{t_0}_{s_0}.$$

By adding $-\alpha^l_v$, we obtain $\sum\limits_{(t,k)\in K(P_0)\setminus\{(v,l)\}} \alpha^k_t + L_G(P_0) < d^{t_0}_{s_0} \Rightarrow (v, l) \in K^*$.

Conversely, suppose that (v, l) is essential. Then, by Definition 5.4, there are $s_0 \in S, t_0 \in T$, and $P_0 \in P_{s_0,t_0}$ such that $v \in P_0$ and $\sum\limits_{(t,k)\in K(P_0)\setminus\{(v,l)\}} \alpha^k_t < d^{t_0}_{s_0} - L_G(P_0)$.

By adding α^l_v, we obtain $\alpha^l_v > L_{G_K}(P_0) - d^{t_0}_{s_0}$. As $L_{G_K}(P_0) - d^{t_0}_{s_0} \geq \min\limits_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$, we obtain $\alpha^l_v > \min\limits_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$ □

Theorem 5.6 *Finding the essential countermeasures for $PCSP(G, K, D)$ can be solved in polynomial time in $\mathcal{O}((\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$.*

Proof. Let (G, K, D) be an instance of PCSP and $(v, l) \in K$. For each $(s, t) \in S \times T$ we compute a shortest path between s and t in the graph G_K. Now, over all the shortest paths containing v we select the one having the minimum value $M = \min\limits_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$. By Proposition 5.5, if, $\alpha^l_v > M$ we deduce that (v, l) is essential. Consequently, finding the essential countermeasures for $PCSP(G, K, D)$ can be reduced to the calculation of $|S| \times |T|$ shortest paths in G_K. □

A procedure for finding the essential countermeasures is given in Algorithm 3.

Algorithm 3: Finding essential countermeasures

Input: An instance (G, K, D) of $PCSP$.
Output: The set K^* of essential countermeasures.
Step 0: $K^* \leftarrow \emptyset$, Compute G_K,
Step 1: Compute a shortest path P^*_{st}, $(s,t) \in S \times T$,
Step 2: For each $(v,l) \in K$:
  Find $M = \min_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\}$
  If $\alpha^l_v > M$:
    $K^* \leftarrow K^* \cup (v,l)$

5.2.1.2 An instance of PCSP with essential countermeasures

In Figure 5.1, we present a simple instance (G, K, D). We set $S = \{s\}$, $T = \{1,2,3,4,5,6\}$, $w_{ij} = 1$ for all $(i,j) \in A\backslash\{(1,5)\}$ and $w_{15} = 2$. We fix one countermeasure per node. The effect of countermeasures as well as the thresholds are described in Figure 5.1. We will use Algorithm 3 to show that $K^* = \{(2,k_2),(4,k_4)\}$. To this end, we construct the graph G_K as shown in Figure 5.2, whose arc weights are $w_{ij} + \sum_{(j,k)\in K_j} \alpha^l_j$ for all $ij \in A$. Next, we compute the shortest $s-t$ paths: $P^*_{s1} = \{s,1\}$, $P^*_{s2} = \{s,1,2\}$, $P^*_{s3} = \{s,1,4,3\}$, $P^*_{s4} = \{s,1,4\}$, $P^*_{s5} = \{s,1,5\}$,$P^*_{s6} = \{s,1,5,6\}$.

We have $M = \min_{(s,t)\in S\times T, v\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\} = \min_{t\in T, 2\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\} = L_{G_K}(P^*_{s2}) - d^2_s = 1$. Since, $\alpha^{k_2}_2 = 2 > 1$, we obtain $(2,k_2) \in K^*$. By using the same method, we can show that $(4,k_4)$ is also essential.

To end this example let us verify that $(5,k_5)$ is not essential. We have $M = \min_{t\in T, 5\in P^*_{st}} \{L_{G_K}(P^*_{st}) - d^t_s\} = L_{G_K}(P^*_{s5}) - d^5_s = 2$. As $\alpha^{k_5}_5 = 2 = M$, we deduce that $(5,k_5)$ is not essential and by the same method it is easy to see that $(1,k_1)$, $(3,k_3)$ and $(6,k_6)$ are so.

5.2.2 Dimension

In this section we characterize of the dimension of $PCSP(G,K,D)$. To this end, we first identify a system of equations of the polytope $PCSP(G,K,D)$. We then prove that every equation of $PCSP(G,K,D)$ is a linear combination of this system.

Proposition 5.7 *Let $(t,k) \in K\backslash K^*$. Then the incidence vector defined by $x^k_t = 0$ and $x^l_v = 1$, for all $(v,l) \in K\backslash\{(t,k)\}$ induces a solution of $PCSP(G,K,D)$.*

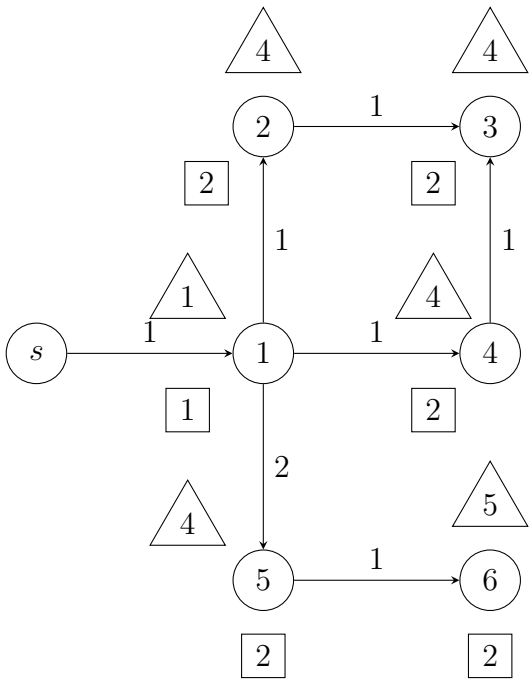


Figure 5.1: An instance with essential countermeasures

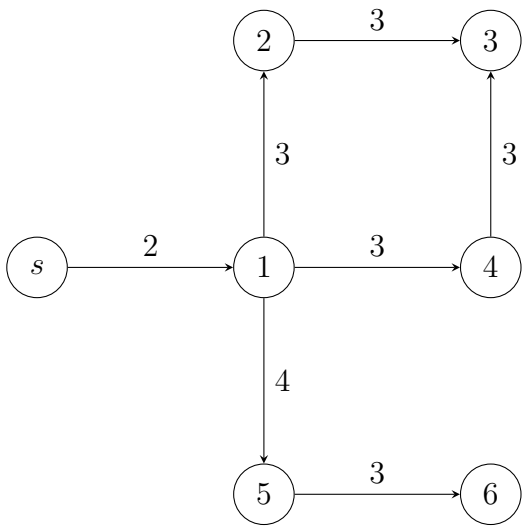


Figure 5.2: The graph G_K

Proof. As $(t, k) \in K \backslash K^*$, then there exists a solution $s_1 \in S(G, K, D)$ such that $(t, k) \notin s_1$. The incidence vector defined by $x_t^k = 0$ and $x_v^l = 1$, for all $(v, l) \in K \backslash \{(t, k)\}$ corresponds to a set $s_2 \in K$ and we have $s_1 \subset s_2$. By Proposition 5.3, we obtain $s_2 \in S(G, K, D)$. $\square$

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} 1 & 2 & \cdots\cdots & |K| & |K^*| & |K| & |K^*|+1 & \cdots\cdots & |K| \end{array} \\
\begin{array}{c} 1 \\ \\ \\ \\ |K^*| \end{array} & \left[\begin{array}{ccc|ccc} & & & & & \\ & 0 & & & I_{|K^*|} & \\ & & & & & \end{array}\right]
\end{array}
$$

$$M =$$

Figure 5.3: Matrix of Equations (5.5)

Proposition 5.8 *Consider $ax = \alpha$ an equation of $PCSP(G, K, D)$. Then $ax = \alpha$ is a linear combination of equations (5.5)*

Proof. Let $ax = \alpha$ be an equation of $PCSP(G, K, D)$ where $x = (x_1, x_2, \ldots, x_{|K|})$ is a solution and $a = (a_1, a_2, \ldots, a_{|K|}) \in \mathbb{R}^{|K|}$. Let M be the matrix of equations (5.5). The latter is given in Figure 5.3. We will prove that there is $\lambda \in \mathbb{R}^{|K|}$ such that $a = \lambda M$.

Consider the solution $x^{s_0} \in \mathbb{R}^{|K|}$ such that $x^{s_0} = (1, 1, \ldots, 1)$. Let $(t, k) \in K \backslash K^*$ and $s_1 = s_0 \backslash \{(t, k)\}$, by proposition 5.7 we have $x^{s_1} \in S(G, K, D)$. Since x^{s_0} and x^{s_1} are both feasible for the PCSP, we have $ax^{s_0} = ax^{s_1} = ax^{s_1} + a_{(t,k)}$. Therefore, $a_{(t,k)} = 0$ for all $(t, k) \in K \backslash K^*$. Consequently, $a = (0, 0, \ldots, 0, a_{|K|-|K^*|+1}, a_{|K^*|}, \ldots, a_{|K|}) = aM$. The result follows by choosing $\lambda = (a_{|K|-|K^*|+1}, a_{|K^*|}, \ldots, a_{|K|})$. □

Equations (5.5) are then the only equations of $PCSP(G, K, D)$. As $rank(M) = |K^*|$ we have the following result.

Theorem 5.9 $dim(PCSP(G, K, T)) = |K| - |K^*|$

Remark 5.10 We can differently prove that $dim(PCSP(G, K, T)) = |K| - |K^*|$: since with the set K^* it is associated $|K^*|$ equations of the form $x_t^k = 1$ for $(t, k) \in K^*$, we have $dim(PCSP(G, K, T)) \leq |K| - |K^*|$. By proposition (5.7), it is easy to see that the incidence vector defined by $x_t^k = 0$ if $(t, k) \in K \backslash K^*$ and $x_t^k = 1$ otherwise, is in $S(G, K, D)$. By varying (t, k) we obtain $|K| - |K^*|$ solutions. Then, we get $|K| - |K^*| + 1$ solutions. Moreover, the incidence vectors of the solutions are affinely independent. Therefore $dim(PCSP(G, K, T)) \geq |K| - |K^*|$. Consequently,

$$dim(PCSP(G, K, T)) = |K| - |K^*|,$$

. As a consequence, we obtain the corollary 5.11.

Corollary 5.11 *PCSP is full dimensional if and only if $K^* = \emptyset$*

5.3 Facial investigation of basic inequalities

We study now the facial structure of the polytope $PCSP(G, K, D)$. In particular, we give necessary and sufficient conditions for inequalities of formulation (5.4) to be facet defining.

5.3.1 Trivial Inequalities

Theorem 5.12 *Inequality (5.3) defines a facet of $PCSP(G, K, D)$ if and only if $(t, k) \in K \setminus K^*$.*

Proof. Let $(t, k) \in K$ and denote by $F_t^k = \{x \in PSCP(G, K, D) : x_t^k = 1\}$ the face induced by inequality (5.3).

Necessity condition: If $(t, k) \in K^*$, then $F_t^k = PSCP(G, K, D)$. Therefore, inequality $x_t^k \leq 1$ doesn't define a facet.

Sufficient condition: Let $(v, l) \in K \setminus (K^* \cup \{(t, k)\})$. By proposition 5.7, the set of countermeasures $S = K \setminus \{(v, l)\}$ corresponds to a solutions of PCSP. The incidence vector x^S is given by $x_v^l = 0$ and $x_v^l = 1$ otherwise. Clearly, $x \in F_t^k$. By varying $(v, l) \in K \setminus \{K^*, (t, k)\}$, we obtain $|K| - |K^*| - 1$ solutions of F_t^k. If we add the solution given by $x_w^m = 1$ for all $(w, m) \in K$, we get $|K| - |K^*|$ solutions of F_t^k whose incidence vectors are affinely independent. Consequently, F_t^k is a facet. □

Lemma 5.13 *Let $L, M \subseteq K$ such that $L \subseteq M$. Then, $M^* \subseteq L^*$.*

Proof. We will prove the lemma by contradiction. Suppose that $L \subseteq M$ but $M^* \subsetneq L^*$. Then, there is $(v, l) \in M^*$ and $(v, l) \in L^*$.

Since $(v, l) \in M^*$, $S(G, M \setminus \{(v, l)\}, D) = \emptyset$. Now, let $s \in S, t \in T$ and $P \in P_{st}$. As $L \subseteq M$, we have $L \setminus \{(v, l)\} \subseteq M \setminus \{(v, l)\}$. Then,

$$\sum_{(w,m) \in L \setminus \{(v,l)\}} \alpha_w^m \leq \sum_{(w,m) \in M \setminus \{(v,l)\}} \alpha_w^m < d_s^t - L_G(P).$$

Consequently, $S(G, M \setminus \{(v, l)\}, D) = \emptyset$ and $(v, l) \in L^*$. □

Proposition 5.14 *Let (G, K, D) be an instance of $PCSP$, and $I, J \subseteq K$ such that $I \subseteq J$. If $S(G, K \setminus J, D) \neq \emptyset$, then $S(G, K \setminus I, D) \neq \emptyset$*

Proof. Consider (G, K, D) an instance of $PCSP$, and let $I, J \subseteq K$ such that $I \subseteq J$. Assume $S(G, K \setminus J, D) \neq \emptyset$ and let $S \in S(G, K \setminus J, D)$. Since $K \setminus J \subseteq K \setminus I$, we have that $S \subseteq K \setminus I$ and it verifies the threshold vector D in the graph G. Hence, $S \in S(G, K \setminus I, D)$. Consequently, $S(G, K \setminus I, D) \neq \emptyset$. □

Theorem 5.15 *Inequality (5.2) defines a facet of $PCSP(G, K, D)$ if and only if $(t, k) \in K \setminus K^*$ and $(K \setminus \{(t, k)\})^* = K^*$*

Proof. Let $(t, k) \in K$ and denote by $F_t^k = \{x \in PSCP(G, K, D) : x_t^k = 0\}$ the face induced by inequality (5.2).

Necessity condition: If $(t, k) \in K^*$, then it is clear that $F_t^k = \emptyset$. Therefore, inequality $0 \leq x_t^k$ doesn't define a facet. Now, suppose that $(K \setminus \{(t, k)\})^* \neq K^*$ and $(t, k) \notin K^*$. By Lemma 5.13, we know that $K^* \subseteq (K \setminus \{(t, k)\})^*$. This implies that there exists a countermeasure $(v, l) \in (K \setminus \{(t, k)\})^*$ and $(v, l) \notin K^*$. Hence, $\mid (K \setminus \{(t, k)\})^* \mid \geq \mid K^* \mid +1$.

Moreover, $dim(F_t^k) = |K| - 1 - \mid (K \setminus \{(t, k)\})^* \mid \leq |K| - |K^*| - 2$. Consequently, F_t^k doesn't define a facet.

Sufficient condition: let $(t, k) \in K \setminus K^*$ and suppose that $(K \setminus \{(t, k)\})^* = K^*$. Let $(v, l) \in K \setminus (K^* \cup \{(t, k)\})$ and $s = K \setminus \{(t, k), (v, l)\}$. Since $(K \setminus \{(t, k)\})^* = K^*$, we have $S(G, K \setminus \{(t, k), (v, l)\}, D) \neq \emptyset$. By Proposition 5.14, we have $S(G, K \setminus \{(t, k), (v, l)\}, D) \subseteq S(G, K, D)$. Then, $s \in S(G, K, D)$. The incidence vector of s is given by $x_v^l = x_t^k = 0$ and $x_w^m = 1$ for $(w, m) \in K \setminus \{(t, k), (v, l)\}$. Clearly, $x \in F_t^k$. Hence, we obtain $|K| - |K^*| - 1$ solutions of F_t^k.

If we add the solution given by $x_w^m = 1$ for all $(w, m) \in K$, we get $|K| - |K^*|$ solutions of F_t^k whose incidence vectors are affinely independent. Therefore, $dim(F_w^k) \geq |K| - |K^*| - 1$. In addition, $dim(F_t^k) \leq |K| - |K^*| - 1$. Consequently, $dim(F_w^k) = |K| - |K^*| - 1$ and F_t^k is a facet. □

5.3.2 Security inequalities

Let us now study the facial aspect of the security inequalities. The following proposition will be useful in this section.

Theorem 5.16 *Let $s \in S$, $t \in T$ and $P \in P_{s,t}$. Inequality (5.1) defines a facet of $PCSP(G, K, D)$ if*

1) For all $(v,l) \in K(P)$, $(v,l) \in K \setminus K^*$ and $\alpha_v^l = \xi$ for some scalar ξ,

2) there exists $r \in \mathbb{N}$ such that $1 \leq r \leq |K(P)|$ and $r\xi = d_s^t - V(P)$,

3) for all $(v,l) \in K \setminus K(P)$ and $J \subseteq K(P)$ such that $(v,l) \notin J$ and $|J| = |K(P)| - r$, we have $S(G, K \setminus \{J \cup \{(v,l)\}\}, D) \neq \emptyset$.

Proof. Denote inequality (5.1) by $ax \leq \alpha$ and let $F_{s,t}^P = \{x \in PCSP(G,K,D) : ax = \alpha\}$. We will prove the result by maximality. Let then $bx \leq \beta$ be a valid inequality defining a facet F of $PCSP(G,K,D)$ such that $F_{s,t}^P \subseteq F$. We will show that there exists $\rho \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{|K|}$ such that $b = \rho a + \lambda M$ where M is the matrix of equations given in Figure 5.3.

Let us start by proving that for all $(v,l) \in K(P)$, $b_v^l = \rho$. We have the following claim.

Claim 5.17 *Let $J_0 \subseteq K(P)$ where $|J_0| = |K(P)| - r$ such that $1 \leq r \leq |K(P)|$. The subset of countermeasures $S_0 = K \setminus J_0$ is in $F_{s,t}^P$.*

Proof. First, we will show that $S_0 = K \setminus J_0 \in S(G,K,D)$. By condition 3, for all $(v,l) \in K \setminus \{K^* \cup K(P)\}$ and $J \subseteq K(P)$ such that $|J| = |K(P)| - r$ we have $S(G, K \setminus \{J \cup \{(v,l)\}\}, D) \neq \emptyset$. Let $(v_0, l_0) \in K \setminus \{K^* \cup K(P)\}$, as $S(G, K \setminus \{J_0 \cup \{(v_0,l_0)\}\}, D) \neq \emptyset$ and $J_0 \subseteq J_0 \cup \{(v_0,l_0)\}\}$. By Proposition 5.14 we have $S(G, K \setminus J_0, D) \neq \emptyset$. Therefore, $S_0 = K \setminus J_0 \in S(G, K \setminus J_0, D) \subseteq S(G,K,D)$. On the other hand, by conditions 1 and 2 we have that for all $(v,l) \in K(P)$ $\alpha_v^l = \xi$, and $r\xi = d_s^t - V(P)$. Hence,

$$\sum_{(v,l) \in K(P) \setminus J_0} \alpha_v^l = \xi(|K(P)| - |J_0|) = \xi(|K(P)| - |K(P) + r|) = \xi r = d_s^t - L_G(P).$$

Consequently, $S_0 = K \setminus J_0 \in F_{s,t}^P$.

Now, we will use Claim 5.17 to show that for all $(v,l) \in K(P)$, $b_v^l = \rho$. Let $(v_1,l_1) \in J_0$ and $(v_2,l_2) \in K(P) \setminus J_0$. Consider $S_1 = (S_0 \cup \{(v_1,l_1)\}) \setminus \{(v_2,l_2)\}$. Along the same line as in Claim 5.17 we can see that $S_1 \in F_{s,t}^P$. In fact, we have that $S_1 \in S(G,K,D)$, and $S_1 = K \setminus J_1$ where $J_1 = (J_0 \cup \{(v_1,l_1)\}) \setminus (v_2,l_2)$. Hence, $J_1 \subset K(P)$ and $|J_1| = |K(P)| - r$, where r is given by Condition 2.

As $S_0, S_1 \in F_{s,t}^P$, we have that $b^{S_0} = b^{S_1} = b^{S_0} + b_{v_1}^{l_1} - b_{v_2}^{l_2}$. This yields $b_{v_1}^{l_1} = b_{v_2}^{l_2}$ for all $(v_1,l_1) \in J_0$ and $(v_2,l_2) \in K(P) \setminus J_0$. Hence, for all $(v,l) \in K(P)$ $b_v^l = \rho$.

Now, we show that for all $(v,l) \in K \setminus (K(P) \cup K^*)$, $b_v^l = 0$. Let $(v,l) \in K \setminus (K(P) \cup K^*)$ and consider $S_2 = S_0 \setminus \{(v,l)\}$. By Condition 3, we have $S(G, K \setminus \{J_0 \cup \{(v,l)\}\}, D) \neq \emptyset$. On the other hand, $S_2 = K \setminus \{J_0 \cup \{(v,l)\}\}$.

By proposition 5.3, we obtain $S_2 \in S(G, K \setminus \{J_0 \cup \{(v,l)\}\}, D) \subset S(G, K, D)$. Since $(v,l) \notin K(P)$ and S_0 satisfies inequality (5.1) with equality, it follows $S_2 \in F^P_{s,t}$.

Hence, $b^{S_0} = b^{S_2} = b^{S_0} - b^l_v$. Then $b^l_v = 0$ for all $(v,l) \in J_0$.

Consequently we have that

$$b^l_v = \begin{cases} \rho & \text{if } (v,l) \in K(P), \\ 0 & \text{if } K \setminus (K(P) \cup K^*). \end{cases}$$

Let $\lambda = (0, \ldots, 0, b_{|K|-|K^*|+1}, \ldots, b_{|K|})$. Hence, the vector b can be written as

$$b = \rho a + \lambda M$$

□

Theorem 5.18 *Let $s \in S$, $t \in T$ and $P \in P_{s,t}$. Inequality (5.1) defines a facet of $PCSP(G, K, D)$ only if*

1) There exists $(v,l) \in K(P)$ such that $\alpha^l_v \leq d^t_s - V(P)$,

2) for all $J \subseteq K^* \cap K(P)$, $\sum_{(v,l)\in J} \alpha^l_v \leq d^t_s - L_G(P)$,

3) there exists $(v,l) \in K(P)$ such that $(v,l) \in K \setminus K^*$ and $\alpha^l_v \neq \frac{1}{|K(P)|}(d^t_s - L_G(P))$.

Proof. Assume that for all $(v,l) \in K(P)$, $\alpha^l_v > d^t_s - L_G(P)$. Then the following holds for any solution of $PCSP(G, K, D)$

$$\sum_{(v,l)\in K(P)} x^l_v \alpha^l_v > d^t_s - L_G(P).$$

This implies that any solution of $PCSP(G, K, D)$ does not belong to $F^P_{s,t}$. As a consequence, inequality (5.1) is not facet defining.

Suppose that there is $J_0 \subseteq K^* \cap K(P)$ such that $\sum_{(v_0,l_0)\in J_0} \alpha^{l_0}_{v_0} > d^t_s - L_G(P)$. Since $J_0 \subseteq K^*$, the following inequalities are valid for $PCSP(G, K, D)$.

$$x^{l_0}_{v_0} \geq 1 \quad \text{for all } (v_0, l_0) \in J_0.$$

By multiplying by $\alpha_{v_0}^{l_0}$ for all $(v_0, l_0) \in J_0$, we obtain the following valid inequalities.

$$\alpha_{v_0}^{l_0} x_{v_0}^{l_0} \geq \alpha_{v_0}^{l_0} \quad \text{for all } (v_0, l_0) \in J_0. \tag{5.6}$$

As $\sum_{(v_0,l_0)\in J_0} \alpha_{v_0}^{l_0} > d_s^t - L_G(P)$, there exists $r \in \mathbb{R}_+^*$ such that $\sum_{(v_0,l_0)\in J_0} \alpha_{v_0}^{l_0} \geq d_s^t - L_G(P) + r$. Then, by summing inequalities (5.6), we obtain

$$\sum_{(v_0,l_0)\in J_0} x_{v_0}^{l_0} \alpha_{v_0}^{l_0} \geq d_s^t - L_G(P) + r$$

Since $J_0 \subseteq K(P)$,it also follows that

$$\sum_{(v_0,l_0)\in J_0} x_{v_0}^{l_0} \alpha_{v_0}^{l_0} \geq d_s^t - L_G(P) + r > d_s^t - L_G(P). \tag{5.7}$$

Inequality (5.1) is then dominated by inequality (5.7). It cannot consequently, be facet defining.

Assume now that for all $(v, l) \in K(P)$ we have $(v, l) \in K^*$ and $\alpha_v^l = \frac{1}{|K(P)|}(d_s^t - L_G(P))$. We will prove that $PCSP(G, K, D) = F_{s,t}^P$. It suffices to show that $PCSP(G, K, D) \subseteq F_{s,t}^P$. Let $x \in PCSP(G, K, D)$, as for all $(v, l) \in K(P)$ we have $(v, l) \in K^*$, we obtain $x_v^l = 1$ for all $(v, l) \in K(P)$. Then,

$$\sum_{(v,l)\in K(P)} \alpha_v^l x_v^l = \sum_{(v,l)\in K(P)} \alpha_v^l.$$

Since, $\alpha_v^l = \frac{1}{|K(P)|}(d_s^t - L_G(P))$ for all $(v, l) \in K(P)$, we obtain

$$\sum_{(v,l)\in K(P)} \alpha_v^l x_v^l = d_s^t - L_G(P)$$

. Hence, every solution in $PCSP(G, K, D)$ belongs to $F_{s,t}^P$. Consequently, $PCSP(G, K, D) = F_{s,t}^P$ and $F_{s,t}^P$ is not facet defining. □

5.4 Valid inequalities and facial aspect

Here, we introduce three families of valid inequalities and provide a facial investigation for each of them.

5.4.1 Path Covering Inequalities (PCI)

The first inequalities are called the Path Covering inequalities and they come from the property that for a given (s,t) unsecured path P, if a set of countermeasures $C_P^{s,t}$ induced by a subset of nodes in P doesn't allow securing P then at least one countermeasure in $P \setminus C_P^{s,t}$ must be placed. More formally, this can be defined as follows.

Definition 5.19 Let $(s,t) \in \Gamma$ and $P \in P_{st}$. A set $C_P^{s,t} \subseteq K(P)$ is said to be a sufficient countermeasure set (resp. a non sufficient countermeasure set) with respect to s,t and P if $C_P^{s,t} \neq \emptyset$ and $\sum_{(v,l)\in C_P^{s,t}} \alpha_v^l \geq d_s^t - L_G(P)$ (resp. if $C_P^{s,t} \neq \emptyset$ and $\sum_{(v,l)\in C_P^{s,t}} \alpha_v^l < d_s^t - L_G(P)$).

Definition 5.20 A set $C_P^{*s,t} \subseteq K(P)$ is a maximal non sufficient countermeasure set with respect to s,t and P if $C_P^{*s,t}$ is a non sufficient countermeasures set and for all $(v,l) \in K(P) \setminus C_P^{s,t}$, the set $C_P^{*s,t} \cup \{(v,l)\}$ is sufficient for s,t and P.

Theorem 5.21 *Let $(s,t) \in \Gamma$, $P \in P_{st}$ and $C_P^{s,t}$ a corresponding non sufficient countermeasures set. Then, the following inequality is valid for $PCSP(G,K,D)$*

$$\sum_{(v,l)\in K(P)\setminus C_P^{s,t}} x_v^l \geq 1. \tag{5.8}$$

Proof. It is easy to see that by definition a non sufficient countermeasure set $C_P^{s,t}$ is a set of countermeasures induced by the path P whose total effect does not satisfy the security inequality of P. Therefore, additional countermeasures in $K(P) \setminus C_P^{s,t}$ are necessary in order to satisfy the security inequality. Hence (5.8). □

Example 5.22 Let (P,K,D) be the instance represented in Figure 5.4, where P is the path $(s,1,2,3,4,5)$.

It is clear that $C^* = \{(1,k_1),(2,k_2)\}$ is a non sufficient countermeasures w.r.t. s, 5 and P. In fact, $\alpha_1^{k_1} + \alpha_2^{k_2} = 3 < 9 - 5 = 4$. C^* is a maximal non sufficient countermeasures set because $C^* \cup \{(3,k_3)\}$, $C^* \cup \{(4,k_4)\}$, and $C^* \cup \{(5,k_5)\}$ are sufficient w.r.t. s, 5 and P.

For this instance, the path covering inequality $x_3^{k_3} + x_4^{k_4} + x_5^{k_5} \geq 1$ is valid.

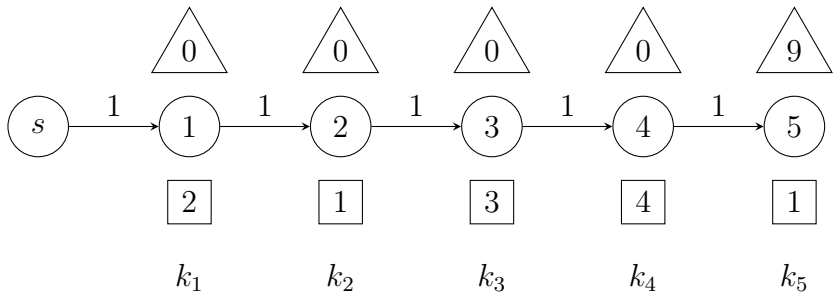


Figure 5.4: Path covering inequalities

In what follow we study the facial aspect of Path Covering Inequalities. Recall that for a given $v \in T$ the set $ST(v) = \{(s,t) \in \Gamma : \exists P \in P_{s,t}, v \in P\}$ is the set of unsecured attacks (s,t) such that there exists a potential unsecured $s-t$ path P that contains v.

Theorem 5.23 *Let $(s,t) \in \Gamma$, $P \in P_{st}$ and $C_P^{s,t}$ the corresponding non sufficient countermeasure set. Inequality (5.8) defines a facet of $PCSP(G,K,D)$ if*

1) $C_P^{s,t}$ is maximal with respect to s,t and P,

2) $K(P) \cap K^* = \emptyset$,

3) for all $v \in K(P) \setminus C_P^{s,t}$, $ST(v) = \emptyset$,

4) for all $(t,k) \in C_P^{s,t}, (v,l) \in K(P) \setminus C_P^{s,t}$, we have $C_P^{s,t} \setminus \{(t,k)\} \cup \{(v,l)\}$ is sufficient with respect to s,t and P.

Proof. Denote inequality (5.8) by $ax \geq \alpha$ and $F_P^{s,t} = \{x \in PSCP(G,K,D) : \sum_{(v,l)\in K(P)\setminus C_P^{s,t}} x_v^l = 1\}$ its associated face. Let $bx \geq \beta$ be a valid inequality defining a facet F. Assume that $F_P^{s,t} \subseteq F$. To prove that $F_P^{s,t}$ is a facet of $PCSP(G,K,D)$, we will show that there exist $\rho \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{|K^*|}$ such that $b = \rho a + \lambda M$, for some $\rho \in \mathbb{R}$ where M is the matrix of equations defined in Figure 5.3.

To this end, we will use the following claim:

Claim 5.24 *Let $(v,l),(w,m) \in K(P)\setminus C_P^{s,t}$ and $(x,n) \in K\setminus K(P) \cup C_P^{s,t}$ such that $(w,m) \neq (v,l)$. The following subsets of countermeasures are solutions of $PCSP(G,K,D)$ and belong to $F_P^{s,t}$:*

- $S_0 = (K \setminus K(P)) \cup (C_P^{s,t} \cup \{(v,l)\})$,
- $S_1 = S_0 \setminus \{(v,l)\} \cup \{(w,m)\}$,
- $S_2 = S_0 \setminus \{(x,n)\}$.

Proof. Let us prove that S_0 is in $F_P^{s,t}$. We start by showing that S_0 of incidence vector x is a solution of $PCSP(G,K,D)$. Let $(s',t') \in \Gamma$ and $P' \in P_{st}$, we will prove that $\sum_{(v,l)\in K(P')} x_v^l \alpha_v^l \geq d_s^t - L_G(P')$. We will distinguish three cases.

Case 1: $P' = P$, as $C_P^{s,t}$ is maximal with respect to s,t and P', the set $C_P^{s,t} \cup \{(v,l)\}$ is sufficient for s',t' and P', that is to say $\sum_{(v,l)\in C_P^{s,t}\cup\{(v,l)\}} x_v^l \alpha_v^l \geq d_{s'}^{t'} - L_G(P')$. Then $\sum_{(v,l)\in K(P')} x_v^l \alpha_v^l \geq d_{s'}^{t'} - L_G(P')$.

Case 2: $P' \neq P$ and there exists $v \in P \cap P'$: Since for all $(v,l) \in K(P') \setminus C_P^{s,t}$, $ST(v) = \emptyset$, then by definition of $ST(v)$, P' is secured which means $L_G(P') \geq d_{s'}^{t'}$. Hence $\sum_{(v,l)\in K(P')} x_v^l \alpha_v^l \geq d_{s'}^{t'} - L_G(P')$.

Case 3: $P' \neq P$ and there doesn't exists $v \in P \cap P'$. Then, all the countermeasures $K(P')$ induced by P' are chosen in S_0. This yields the security inequality associated to P' is satisfied: $\sum_{(v,l)\in K(P')} x_v^l \alpha_v^l \geq d_{s'}^{t'} - L_G(P')$. This is true, because if not even the placement of the set K will not satisfy the security inequality associated to P', which is to say $S(G,K,D) = \emptyset$ and a contradiction holds.

As a consequence, we obtain $S_0 \in S(G,K,D)$. In addition, by construction of S_0, the unique element of $K(P) \setminus C_P^{s,t}$ that belongs to S_0 is (v,l). Therefore, $\sum_{(w,m)\in K(P)\setminus C_P^{s,t}} x_w^m = 1$ which implies that $S_0 \in F_P^{s,t}$.

By the same reasoning used for S_0 we can easily show that $S_1 \in F_P^{s,t}$.

Now consider $S_2 = S_0 \setminus \{(x,n)\}$ and let us show that S_2 is in $S(G,K,D)$. Let $(s',t') \in \Gamma$ and $P' \in P_{st}$, we will prove that $\sum_{(v,l)\in K(P')} x_v^l \alpha_v^l \geq d_s^t - L_G(P')$. We distinguish two cases. If $(x,n) \in C_P^{s,t}$, then by condition 4 we obtain S_2 is a solution. Now, if $(x,n) \notin C_P^{s,t}$ by using condition 4 we can ensure that S_2 remains a solution of $PCSP(G,K,D)$. On the other hand, by construction of S_2, the unique element of $K(P) \setminus C_P^{s,t}$ that is in S_2 is (v,l). Therefore, $\sum_{(w,m)\in K(P)\setminus C_P^{s,t}} x_w^m = 1$ which implies that $S_2 \in F_P^{s,t}$.

By claim 5.24, S_0, S_1 and S_2 are in $F_P^{s,t}$, for all $(v,l),(w,m) \in K(P)\setminus C_P^{s,t}$ and $(x,n) \in K\setminus K(P)\cup C_P^{s,t}$. As S_0, S_1 are in $F_P^{s,t}$, $b^{S_0} = b^{S_1} = b^{S_0} - b_v^l + b_w^m$. Then, $b_v^l = b_w^m$. Therefore, for all $(v,l) \in K(P)\setminus C_P^{s,t}$ we obtain $b_v^l = \rho$. On the other hand, since $b^{S_0} = b^{S_2} = b^{S_0} - b_x^n$, $b_x^n = 0$ for all $(x,n) \in (K\setminus K^*)\setminus(K(P)\setminus C_P^{s,t})$. Consequently, we have that

$$b_v^l = \begin{cases} \rho & \text{if } (v,l) \in K(P)\setminus C_P^{s,t}, \\ 0 & \text{if } (v,l) \in (K\setminus K^*)\setminus(K(P)\setminus C_P^{s,t}). \end{cases}$$

Let $\lambda = (0,\ldots,0,b_{|K|-|K^*|+1},\ldots,b_{|K|})$. Hence, the vector b can be written as $b = \rho a + \lambda M$ which ends the proof.

□

Theorem 5.25 *Let $(s,t) \in \Gamma$, $P \in P_{st}$ and $C_P^{s,t}$ be the corresponding non sufficient countermeasure set. Inequality ((5.8)) defines a facet of $PCSP(G,K,D)$ only if*

1) $(K(P)\setminus C_P^{s,t}) \cap K^* = \emptyset$,

2) $C_P^{s,t}$ is maximal w.r.t. s,t and P, or $\mid K(P)\setminus C_P^{s,t} \mid \geq 2$.

Proof. Consider $(s,t) \in \Gamma$, $P \in P_{st}$ and let$C_P^{s,t}$ be the corresponding non sufficient countermeasures set. Let us denote by $F_P^{s,t} = \{x \in PSCP(G,K,D) : \sum_{(v,l)\in K(P)\setminus C_P^{s,t}} x_v^l = 1\}$ the face associated with inequality ((5.8)).

Suppose that $(K(P)\setminus C_P^{s,t}) \cap K^* \neq \emptyset$. Then, there exists $(v,l) \in (K(P)\setminus C_P^{s,t}) \cap K^*$. If $\mid (K(P)\setminus C_P^{s,t}) \cap K^* \mid = 1$, then it is clear that $PCSP(G,K,D) = F_P^{s,t}$. If $\mid (K(P)\setminus C_P^{s,t}) \cap K^* \mid \geq 2$, then $F_P^{s,t} = \emptyset$. In both cases $F_P^{s,t}$ cannot be facet defining.

Assume now that $C_P^{s,t}$ is not maximal w.r.t. s,t and P, and that $\mid K(P)\setminus C_P^{s,t} \mid = 1$. As $C_P^{s,t}$ is not maximal, there exists $(v,l) \in K(P)\setminus C_P^{s,t}$ such that $C_P^{s,t} \cup \{(v,l)\}$ is non sufficient. In addition, since $\mid K(P)\setminus C_P^{s,t} \mid = 1$, we obtain $K(P)\setminus C_P^{s,t} = \{(v,l)\}$. Hence, $C_P^{s,t} \cup \{(v,l)\} = K(P)$ which is non sufficient. Consequently, $F_P^{s,t} = \emptyset$, and hence it is not facet defining. □

5.4.2 Countermeasures Path Inequalities (CmPI)

Let us first define what is a *Countermeasure Path* (CmP).

Definition 5.26 *Let $J = \{(t_1, k_1), (t_1, k_1), \ldots, (t_n, k_n)\} \subseteq K \backslash K^*$. If $(t_{i+1}, k_{i+1}) \in (K \setminus \{(t_i, k_i)\})^*$ for $i = 1, \ldots, n-1$, then the set J is said to be Countermeasures Path set. We refer to such set as CmP.*

Note that the set J exists only if $S(G, K \backslash \{(t_i, k_i)\}) \neq \emptyset$ for all $i \in \{1, \ldots, n-1\}$, since $K \setminus \{(t_i, k_i)\})^*$ is only defined in this case.

Theorem 5.27 *Let $J = \{(t_1, k_1), (t_1, k_1), \ldots, (t_n, k_n)\}$ be a CmP. The following inequality is valid for $PCSP(G, K, D)$*

$$\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \lceil \frac{n-1}{2} \rceil. \tag{5.9}$$

Proof. Let $i \in \{1, \ldots, n-1\}$, we will first prove that $x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} \geq 1$ is valid for $PCSP(G, K, D)$. We then show the validity of inequality (5.9) by using Chvátal Gomory procedure.

Let $x \in R^{|K|}$. If $x_{t_i}^{k_i} = 1$, whatever the value of $x_{t_{i+1}}^{k_{i+1}}$ inequality $x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} \geq 1$ is verified. Now if $x_{t_i}^{k_i} = 0$, since $(t_{i+1}, k_{i+1}) \in (K \setminus \{(t_i, k_i)\})^*$ we should have $x_{t_i}^{k_i} = 1$ and $x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} \geq 1$ is valid.

The following inequalities are valid for $PCSP(G, K, D)$.

$$\begin{aligned}
&x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} \geq 1 \quad \text{for all } i \in \{1, \ldots, n-1\} \\
&x_{t_1}^{k_1} \geq 0, \\
&x_{t_n}^{k_n} \geq 0.
\end{aligned}$$

By summing these inequalities, we obtain

$$2 \sum_{i=1}^{n} x_{t_i}^{k_i} \geq n - 1.$$

By dividing by 2 and rounding up the right hand side, we obtain

$$\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \lceil \frac{n-1}{2} \rceil$$

□

Now, we will study the facet aspect of these inequalities. We will give necessary and sufficient conditions to these inequalities to be facet defining.

Theorem 5.28 *Let $J = \{(t_1, k_1), (t_1, k_1), \ldots, (t_n, k_n)\}$ be a CmP. Inequality (5.9) defines a facet of $PCSP(G, K, D)$ if for all $I \subseteq K \setminus K^*$ such that $|I| = n - \lceil \frac{n-1}{2} \rceil + 1$, $S(G, K \setminus I, D) \neq \emptyset$.*

Proof. Denote inequality (5.9) by $ax \leq \alpha$ and $F_n = \{x \in PSCP(G, K, D) : \sum_{i=1}^{n} x_{t_i}^{k_i} = \lceil \frac{n-1}{2} \rceil\}$ its associated face. Let $bx \leq \beta$ be a valid inequality defining a facet F and assume that $F_n \subseteq F$. We will show that there exist $\rho \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{|K^*|}$ such that $b = \rho a + \lambda M$, where M is the matrix of equations defined in Figure 5.3. Fir this we show the following.

Claim 5.29 *There is $I_0 \subset J$ with $|I_0| = n - \lceil \frac{n-1}{2} \rceil$ such that $S_0 = K \setminus I_0 \in F_n$.*

Proof. First, we will show that there exists $I_0 \subseteq J$ such that $|I_0| = n - \lceil \frac{n-1}{2} \rceil$ and $S_0 = K \setminus I_0 \in S(G, K, D)$. Let $I_0 \subseteq J$ such that $|I_0| = n - \lceil \frac{n-1}{|} 2$. Let $(v, l) \in K \setminus K^*$ and $(v, l) \notin K \setminus I_0$. The set $I_0 \cup \{(v, l)\}\} \subseteq K \setminus K^*$ and it is of size $n - \lceil \frac{n-1}{2} + 1 \rceil$. By the condition of Theorem 5.28, $S(G, K \setminus I_0 \cup \{(v, l)\}\}, D) \neq \emptyset$. Now, by proposition 5.14 we obtain $S(G, K \setminus I_0, D) \neq \emptyset$. Therefore, $S_0 = K \setminus I_0 \in S(G, K \setminus I_0, D) \subseteq S(G, K, D)$.

In addition, x^{S_0} satisfies $\sum_{i=1}^{n} x_{v_i}^{l_i} = \sum_{(v,l) \in J \setminus I_0} x_v^l = \lceil \frac{n-1}{2} \rceil$. Consequently, $S_0 = K \setminus I_0 \in F_n$.

Let now $(v, l) \in J \setminus I_0$ and $(w, m) \in I_0$ and consider $S_1 = (S_0 \cup \{(w, m)\}) \setminus \{(v, l)\}$. We can write $S_1 = K \setminus I_1$ where $I_1 = (I_0 \cup \{(w, m)\}) \setminus \{(u, l)\} \subseteq J$ and $|I_1| = n - \lceil \frac{n-1}{2} \rceil$. Along the same line as in Claim 5.29, it is easy to see that $S_1 \in F_n$.

As $S_0, S_1 \in F_n$, we obtain $b^{S_0} = b^{S_1} = b^{S_0} + b_w^m - b_v^l$. This yields, $b_w^m = b_v^l$ for all $(v, l) \in J \setminus I_0$ and $(w, m) \in I_0$. We then deduce that for all $(v, l) \in J$ $b_v^l = \rho$, for some $\rho \in \mathbb{R}$.

Consider now $(v, l) \in K \setminus (K^* \cup J)$ and $S_2 = K \setminus \{I_0 \cup \{(v, l)\}\}$. We have for all $I \subseteq K \setminus K^*$ such that $|I| = n - \lceil \frac{n-1}{2} \rceil + 1$, $S(G, K \setminus I, D) \neq \emptyset$. In particular, for $I = I_0 \cup \{(v, l)\}$ we obtain $S(G, K \setminus I_0 \cup \{(v, l) D) \neq \emptyset$. Then, $S_2 = K \setminus I_0 \cup \{(v, l)\} \subseteq S(G, K \setminus I_0 \cup \{(v, l) D) \subseteq S(G, K, D)$. In addition, the incidence vector of S_2 satisfies inequality (5.28) at equality. Hence, $S_2 \in F_n$.

As $S_0, S_2 \in F_n$, we obtain $b^{S_0} = b^{S_2} = b^{S_0} - b_v^l$. This yields, $b_v^l = 0$ for all $(v, l) \in K \setminus (K^* \cup J)$. As a consequence,

$$b_v^l = \begin{cases} \rho & \text{if } (v, l) \in J, \\ 0 & \text{if } (v, l) \in (K \setminus K^*) \setminus J. \end{cases}$$

Let $\lambda = (0, \ldots, 0, b_{|K|-|K^*|+1}, \ldots, b_{|K|})$. Hence, the vector b can be written as $b = \rho a + \lambda M$ which ends the proof. $\square$

Theorem 5.30 *Inequality (5.9) defines a facet of $PCSP(G, K, D)$ only if*

1) n is even,

2) There exists $I \subseteq J$, $|I| \geq n - \lceil \frac{n-1}{2} \rceil$ such that $S(G, K \setminus I, D) \neq \emptyset$.

Proof. Assume that n is odd. Then, $\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \frac{n-1}{2}$ can be obtained by linear combination of $x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} \geq 1$ for all $i \in \{1, \ldots, n-1\}$. As a consequence, F_n is not a facet of $PCSP(G, K, D)$.

Suppose now that for all $I \subseteq J$, $|I| \geq n - \lceil \frac{n-1}{2} \rceil$, the set of solutions $S(G, K \setminus I, D) = \emptyset$. We will prove that inequality (5.9) is dominated by $\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \lceil \frac{n-1}{2} \rceil + 1$ which implies that (5.9) is not facet defining. For this, it suffices to show that $\sum\limits_{i=1}^{n} x_{t_i}^{k_i} \neq \lceil \frac{n-1}{2} \rceil$.

Suppose that $\sum\limits_{i=1}^{n} x_{t_i}^{k_i} = \lceil \frac{n-1}{2} \rceil$. Then, there exists a solution s in $S(G, K, D)$ where x is its associated incidence vector and $\sum_{i=1}^{n} x_{t_i}^{k_i} = \lceil \frac{n-1}{2} \rceil$. Let $M \subseteq \{1, \ldots, n\}$ for which $x_{t_m}^{k_m} = 1$ for all $m \in M$, and $N = \{1, \ldots, n\} \setminus M$ for which $x_{t_n}^{k_n} = 0$ for all $n \in N$. Consider then $S_0 = \cup_{n \in N} \{(t_n, k_n)\}$. We have that $s \in S(G, K, D)$ and $s \subseteq K \setminus S_0$ which means that $s \in S(G, K \setminus S_0, D)$. On the other hand, $S_0 \subseteq J$ and $|S_0| = n - \lceil \frac{n-1}{2} \rceil$. We obtain a contradiction by the fact that $S(G, K \setminus S_0, D) = \emptyset$. As a result, inequality (5.9) is not facet defining. $\square$

5.4.3 Essential -by Subsets Removing- Countermeasures (ESRC) inequalities

We introduce now the Essential-by Subsets Removing- Countermeasure (ESRC) inequalities. Given a subset of non essential countermeasures K' of size n and an integer $p \leq n$, an ESRC w.r.t. K' and p is a countermeasure that becomes essential at each time we remove a subset of size p from K'. If such countermeasure exists, we must ensure that either we have chosen it and, if not we have not removed from the solution a subset of K' of size greater than or equal to p. This is the intuition behind this family of valid inequality we introduce in the following.

Definition 5.31 *Let $K' \subseteq K \setminus K^*$ of size n. Let $p \in \{2, \ldots, n\}$. Consider $(t_0, k_0) \in K \setminus K^* \cup \{(t_0, k_0)\}$. The countermeasure (t_0, k_0) is said to be Essential by Subsets Removing Countermeasure (ESRC) w.r.t K' and p iff for all $L \in K', |L| = p$ $(t_0, k_0) \in (K \setminus L)^*$.*

We consider the Essential by Subsets Removing Countermeasure Problem (ESRCP) defined as:

Given an instance (G, K, D) of the PCSP, a non-essential countermeasure (t_0, k_0), an integer k, is there $K' \subseteq K \setminus K^*$ such that $p \leq \mid K \mid$ and $p \leq k$ for which (t_0, k_0) is an ESRC.

Theorem 5.32 *The ESRCP is NP-Complete, even if G is a path.*

Proof. It is easy to see that ESRCP is in NP. In fact, given $L \in K', |L| = p$, checking if $(t_0, k_0) \in (K \setminus L)^*$ can be done in $O(n^3 log(n))$ using theorem 5.6. We then repeat this $C_n^p = O(n^p)$ times. Hence, we can check in $O(n^p n^3 log(n))$ if a couple (K', p) is a solution of ESRCP w.r.t a given (t_0, k_0).

We transform SUBSET SUM to ESRCP. The Subset Sum problem is the problem defined as follows. Given a finite set A with size $s(a) \in \mathbb{N}$ for each $a \in A$ and an integer B, find a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = B$.

Let an arbitrary instance of SUBSET SUM be given by a set A of cardinality m, with size $s(a) \in \mathbb{N}$ for each $a \in A$, and an integer B. We must construct an instance (G, K, D), $(t_0, k_0) \in K \setminus K^*$ and $k \geq 0$ of ESRCP, such that A has a subset of size B if and only if there exists $K' \subseteq K \setminus K^*$ and $p \leq \mid K' \mid, p \leq k$ such that (t_0, k_0) is an ESRC w.r.t (K', p).

For this, we first construct a path graph $G = (V, X)$, where $V = S \cup T$, and X is the set of arcs. We choose only one access point $S = \{s\}$. The set of asset-vulnerability nodes will be composed of the set A and two additional nodes i and t_0, that is $T = A \cup \{i\} \cup \{t_0\}$. We construct the set of arcs $X = \{(s, a_1)\} \cup \{(a_i, a_{i+1}), i = 1, \ldots, m\} \cup \{(a_m, t_0)\}$, with $w_{ij} = 0$ for all $(i, j) \in X$.

Second, for each node a in A, we construct exactly one countermeasure $K_a = \{(a, k_a)\}$ with an effect $\alpha_a^{k_a} = s(a)$. We also construct $K_{t_0} = \{(t_0, k_0)\}$ with an effect $\alpha_{t_0}^{k_0} = B + \epsilon$, where $0 < \epsilon < 1$. We then set $K_i = \emptyset$.

Finally, we set $d_s^a = 0$ for all $a \in A$, $d_s^i = B$, $d_s^{t_0} = B + \epsilon$ and $k = m$. The reduction is illustrated in Figure 5.5.

It is not hard to see that (G, K, D), (t_0, k_0), and k can be constructed from A with size $s(a)$ for each $a \in A$, and B in polynomial time. All that we have

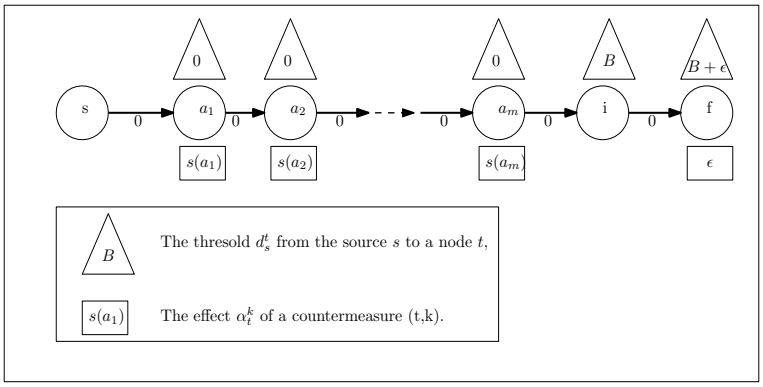


Figure 5.5: Reduction

to show now is that A has a subset of size B iff there exists $K' \subseteq K \setminus K^*$ and $p \leq \mid K' \mid, p \leq k$ such that (t_0, k_0) is an ESRC w.r.t (K', p).

To this end, suppose that $A' \subseteq A$ is a subset sum for A, s and B. We can use A' to obtain a couple (K', p) for which (t_0, k_0) is an ESRC w.r.t K' and p. It suffices to consider $K' = A \setminus A'$ and $p = |A \setminus A'|$. In that case, we have $p \leq m = |A| = k$. In addition, since $\sum\limits_{a \in A'} s(a) = B$ and $d_s^{t_0} = B + \epsilon$, By Theorem 5.6 we simply obtain $(t_0, k_0) \in (K \setminus K')^*$.

Conversely, suppose that (t_0, k_0) is an ESRC w.r.t K' and p. We have the following claim.

Claim 5.33 *For all* $L \in K', |L| = p, \sum\limits_{a \in K \setminus L} s(a) = B$

Proof. Let (t_0, k_0) be an ESRC w.r.t K' and p. We have for all $L \subseteq K', |L| = p$, $(t_0, k_0) \in (K \setminus K')^*$. By theorem 5.6 this is equivalent to say that for all $L \subseteq K'$, $|L| = p$, $\alpha_{t_0}^{k_0} > \min\limits_{(s,t) \in S \times T, t_0 \in P_{st}^*} \{L_{G_{K \setminus L}}(P_{st}^*) - d_s^t\}$. Since there is only one path passing through t_0, $\alpha_{t_0}^{k_0} > L_{G_{K \setminus L}}(P_{st_0}^*) - d_s^{t_0}$. Then, $\sum\limits_{a \in K \setminus L} s(a) < B + \epsilon$. Since for all $a \in A$, $s(a) \in \mathbb{N}$ and $B \in \mathbb{N}$, we obtain $\sum\limits_{a \in K \setminus L} s(a) \leq B$. On the other hand, it is easy to see that $\sum\limits_{a \in K \setminus L} s(a) \geq B$. In fact, we know that $S(G, K \setminus L, D) \neq \emptyset$ and $d_s^i = B$. Then, for all $P \in P_{si}$, $L_G(P) \geq B$. Therefore, $\sum\limits_{a \in K \setminus L} s(a) \geq B$. As a result, $\sum\limits_{a \in K \setminus L} s(a) = B$.

Let $L_0 \subseteq K', |L_0| = p$, by Claim 5.33, $\sum\limits_{a \in K \setminus L_0} s(a) = B$. We use $K \setminus L_0$ to construct the set $A' = K \setminus L_0$. A' is a subset sum for A, s, and B. $\square$

Theorem 5.34 *Let $K' \subseteq K \setminus K^*$ of size n and $p \in \{2, \ldots, n\}$ and denote by I_n^q the set of subsets of K' of size q. Let (t_0, k_0) be an ESRC w.r.t. K' and p. Then, the following inequalities are valid for $PCSP(G, K, D)$ for all $q \in \{1, \ldots, n-p+1\}$,*

$$qx_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l \geq q \quad \forall I \in I_n^{p+q-1}. \tag{5.10}$$

Proof. We will prove by recurrence that inequality (5.10) is valid for all $q \in \{1, \ldots, n-p+1\}$.

Let us prove this for $q = 1$ which means

$$x_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l \geq 1 \text{ for all } I \in I_n^p$$

. Let $I \in I_n^p$. Two cases are possible. 1) If $x_{t_0}^{k_0} = 1$, then $x_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l \geq 1$ is satisfied. If $x_{t_0}^{k_0} = 0$, $\sum_{(u,l)\in I} x_u^l \geq 1$. Suppose that $\sum_{(u,l)\in I} x_u^l = 0$. As, $x_u^l \geq 0$ for all $(u,l) \in I$, we obtain $x_u^l = 0$ for all $(u,l) \in I$. On the other hand, since K' is an ESRC, we have $(t_0, k_0) \in (K \setminus I)^*$. This means that (t_0, k_0) must be chosen. This yields $x_{t_0}^{k_0} = 1$ which contradicts $x_{t_0}^{k_0} = 0$. Consequently, inequality (5.10) is valid for $q = 1$.

Now, suppose that inequality (5.10) is valid for an integer $q \in \{1, \ldots, n-p\}$, by using chvátal Gomory proceeder and using the following claim, we will show that this remains true for $q+1$.

Let us denote by $(I_i)_{\{i=1,\ldots,p+q\}}$ the elements of I_n^{p+q-1}. By summing the inequalities (5.10) over the elements of I_n^{p+q-1} we obtain

$$\sum_{i=1}^{p+q} qx_{t_0}^{k_0} + \sum_{i=1}^{p+q} \sum_{(u,l)\in I_i} x_u^l \geq \sum_{i=1}^{p+q} q.$$

Moreover, for all $I_i \in I_n^{p+q-1}, i = 1, \ldots, p+q$, there exists one and only one $\{(t_i, k_i)\} \in I$ such that $I_i = I \setminus \{(t_i, k_i)\}$. Then,

$$q(p+q)x_{t_0}^{k_0} + \sum_{i=1}^{p+q} \sum_{(u,l)\in I\setminus\{(t_i,k_i)\}} x_u^l \geq q(p+q).$$

Let $i \in \{1, \ldots, p+q\}$. So $I_i = I \setminus \{(t_i, k_i)\}$, $\{(t_i, k_i)\} \notin I_i$ for all $i \in \{1, \ldots, p+q\}$ and $\{(t_i, k_i)\} \in I_j$ for all $j \neq i$. Therefore, each $(t_i, k_i) \in I$ is compted $|I_n^{p+q-1}| - 1 = p+q-1$ times in the sum $\sum_{i=1}^{p+q} \sum_{(u,l)\in I\setminus\{(t_i,k_i)\}} x_u^l$. Hence,

$\sum_{i=1}^{p+q} \sum_{(u,l)\in I\setminus\{(t_i,k_i)\}} x_u^l = (p+q-1) \sum_{(u,l)\in I} x_u^l$. As a consequence, the following inequality is valid for $PCSP(G,K,D)$.

$$q(p+q)x_{t_0}^{k_0} + (p+q-1) \sum_{(u,l)\in I} x_u^l \geq q(p+q) \tag{5.11}$$

As $(p-1)x_{t_0}^{k_0} \geq 0$, by summing this inequality together with inequality 5.11, we obtain

$$(q(p+q)+(p-1))x_{t_0}^{k_0} + (p+q-1) \sum_{(u,l)\in I} x_u^l \geq q(p+q).$$

Note that $q(p+q)+(p-1) = (q+1)(p+q-1)$. By dividing the former inequality by $(p+q-1)$, we obtain

$$(q+1)x_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l \geq \frac{q(p+q)}{p+q-1}.$$

By rounding up, we obtain

$$(q+1)x_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l \geq (q+1)$$

□

Theorem 5.35 *Let $K' \subseteq K \setminus K^*$ of size n and $p \in \{2,\ldots,n\}$. Let (t_0,k_0) be an ESRC w.r.t. K' and p. Let $q \in \{1,\ldots,n-p+1\}$, inequality (5.10) defines a facet of $PCSP(G,K,D)$ if for all $I \in I_n^{p+q-1}$ and $(t,k) \in K \setminus (K^*, I, \{(t_0,k_0))\})$, we have $S(G, K\setminus\{I,(t,k)\}, D) \neq \emptyset$.*

Proof. Denote inequality (5.10) by $ax \leq \alpha$ and $F_{n,p}^q = \{x \in PSCP(G,K,D) : qx_{t_0}^{k_0} + \sum_{(u,l)\in I} x_u^l = q\}$ its associated face. Let $bx \leq \beta$ be a valid inequality defining a facet F. Assume that $F_{n,p}^q \subseteq F$. We will show that there exist $\rho \in \mathbb{R}$ and $\lambda \in \mathbb{R}^{|K^*|}$ such that $b = \rho a + \lambda M$, where M the matrix of equations defined in Figure 5.3.

Claim 5.36 *Denote by $I = \{(t_1,t_1),(t_2,t_2),\ldots,(t_{p+q-1},t_{p+q-1})\}$. The following subset of countermeasures are solutions of $PCSP(G,K,D)$ and their incidence vectors belong to $F_{n,p}^q$*

- $S_0 = K \setminus I$,
- $S_1 = K \setminus \cup_{i=0}^{p-1}\{(t_i, k_i)\}$,
- $S_3 = S_0 \setminus \{(t,k)\}$ *for all* $(t,k) \in K \setminus (K^* \cup I \cup \{(t_0,k_0)\})$,
- $S_{ij} = S_1 \setminus \{(t_i,k_i)\} \cup \{(t_j,k_j)\}$ *for all* $i \in \{1,\dots,p-1\}, j \in \{p,\dots,p+q-1\}$.

Proof. By assumption for all $I \in I_n^{p+q-1}$ and $(t,k) \in K \setminus \{K^*, I, (t_0,k_0)\}$, we have $S(G, K \setminus \{I,(t,k)\}, D) \neq \emptyset$. By Proposition 5.14, $S(G, K \setminus I, D) \neq \emptyset$.

Let $S_0 = K \setminus I \in S(G,K,D)$. We have $S_0 \in S(G, K \setminus I, D) \neq \emptyset$. Then, $S(G, K \setminus I, D) \subseteq S(G,K,D)$ and we obtain $S_0 \in S(G,K,D)$. In addition $(t_0,k_0) \in S_0$. Therefore, S_0 satisfies inequality (5.10) with equality. Consequently, $S_0 = K \setminus I \in F_{n,p}^q$.

Note that $S_1 = S_0 \cup_{i=p}^{p+q-1} \{(t_i,k_i)\}$. We have $S_0 \subseteq S_1 \in S(G,K,D)$. By proposition 5.3, $S_1 \in S(G,K,D)$. On the other hand, S_1 satisfies inequality (5.10) at equality. Therefore, $S_1 = K \setminus \cup_{i=0}^{p-1}\{(t_i,k_i)\} \in F_{n,p}^q$.

Now, let $i \in \{1,\dots,p-1\}, j \in \{p,\dots,p+q-1\}$. As we have proved that $S_1 \in F_{n,p}^q$, we can show that $S_{ij} = S_1 \setminus \{(t_i,k_i)\} \cup \{(t_j,k_j)\} \in F_{n,p}^q$.

Finally, by assumption we have $S_3 \in S(G,K,D)$. Since S_3 satisfies inequality (5.10) at equality, $S_3 = S_0 \setminus \{(t,k)\} \in F_{n,p}^q$ which ends the proof of the claim.

As $S_{ij}, S_1 \in F_{n,p}^q$, we have that $b^{S_{ij}} = b^{S_1} = b^{S_1} - b_{t_i}^{k_i} + b_{t_j}^{k_j}$. Then, $b_{t_i}^{k_i} = b_{t_j}^{k_j}$. As a consequence, for all $(v,l) \in I$, we have $b_v^l = \rho$, for some scalar ρ.

Since $S_0, S_1 \in F_{n,p}^q$, we have $b^{S_1} = b^{S_0}$. Then, $\sum_{i=1}^{p+q-1} b_{t_i}^{k_i} = \sum_{i=0}^{p-1} b_{t_i}^{k_i}$. Therefore, $b_{t_0}^{k_0} = \sum_{i=p}^{p+q-1} b_{t_i}^{k_i} = \rho q$.

Now since $S_0, S_3 \in F_{n,p}^q$, $b^{S_3} = b^{S_0} = b^{S_0} - b_v^l$. Therefore, for all $(v,l) \in K \setminus (K^* \cup I \cup \{(t_0,k_0)\})$, we get $b_v^l = 0$.

As a consequence, we obtain

$$b_v^l = \begin{cases} \rho q & \text{if } (v,l) = (t_0,k_0), \\ \rho & \text{if } (v,l) \in I, \\ 0 & \text{if } (v,l) \in K \setminus (K^* \cup I \cup \{(t_0,k_0)\}), \end{cases}$$

Let $\lambda = (0, \ldots, 0, b_{|K|-|K^*|+1}, \ldots, b_{|K|})$. Hence, the vector b can be written as $b = \rho a + \lambda M$. $\square$

Note that for $p = n$, the Inequality (5.10) will be written as

$$\sum_{(v,l) \in K'} x_v^l \geq 1. \tag{5.12}$$

5.5 Concluding remarks

In this chapter, we have investigated the polytope associated with the PCSP path formulation. We have characterized its dimension and studied the facial aspect of its basic constraints. We have further introduced three families of valid inequalities. We have also discussed necessary conditions as well as sufficient conditions for these inequalities to be facet defining. Based on these results, a Branch-and-Cut algorithm will be developed in the next chapter.

Chapter 6

Branch-and-Cut algorithm and computational study

Contents

6.1 Branch-and-Cut algorithm 108

6.1.1 Preprocessing 108

6.1.2 Algorithm description 109

6.1.3 Feasibility test 110

6.1.4 Separation problems and algorithms 110

6.1.5 Implementation's features 117

6.1.6 Branching strategy 118

6.1.7 Primal heuristic 118

6.2 Computational study 119

6.2.1 Random instances 121

6.2.2 Realistic instances 129

6.3 Concluding remarks 133

In this chapter, we develop a Branch-and-Cut algorithm for the PCSP. The algorithm will be based on the formulation PCSP2. The aim is to perform algorithmic applications of the polyhedral results described in the previous chapter. First, we present the preprocessing phase including the essential countermeasure equations and the optimality conditions presented in Chapter 4. Next, we give an overview of the algorithm. Then, we describe separation routines for the basic and valid inequalities. We also provide some implementation's features, explain our branching strategy and propose a primal heuristic.

Furthermore, we present numerical tests of the compact formulation PCSP1 and the path formulation PCSP2. PCSP1 is directly solved using Cplex with its cuts and pre-solve features. PCSP2 is solved using the Branch and Cut algorithm described in this chapter. The aim of the computational study is to examine, from an algorithmic point of view, the efficiency of the polyhedral study given in Chapters 4 and 5. To this end, we investigate the impact of optimality condition inequalities and valid inequalities in the resolution of the problem. The tests are executed on random and realistic instances.

6.1 Branch-and-Cut algorithm

The Branch-and-Cut algorithm starts with preprocessing phase in which we consider a restricted version of the problem. This includes the essential countermeasures equations as well as optimality condition inequalities.

6.1.1 Preprocessing

Since the path formulation (5.4) is given with a huge number of security inequalities, we first consider a restricted version of the corresponding linear program. A restricted number of security inequalities (4.19) are then generated in the first LP. For each $s \in S, t \in T$, we generate only the security inequality associated with the shortest $s-t$ path. We also use the essential countermeasures equations (5.5) as well as the optimality condition inequalities (4.21).

Therefore, the initial linear program LP_0 that we solve in the first step is as follows

$$\text{Min} \sum_{(t,k)\in K} \alpha_t^k x_t^k$$

$$\sum_{(v,l)\in K(P_{st}^*)} \alpha_v^k x_v^l \geq d_s^t - L_G(P_{st}^*) \quad s \in S, t \in T, \tag{6.1}$$

$$x_t^k \geq x_t^l \quad t \in T, (t,k), (t,l) \in K_t : (t,k) \succeq (t,l), \tag{6.2}$$

$$x_t^k = 1 \quad (t,k) \in K^*, \tag{6.3}$$

$$0 \leq x_t^k \quad (t,k) \in K, \tag{6.4}$$

$$x_t^k \leq 1 \quad (t,k) \in K, \tag{6.5}$$

where P_{st}^* is the shortest $s-t$ path.

6.1.2 Algorithm description

Denote by $\overline{x} \in \mathbb{R}^K$ the solution of the current linear relaxation (5.4) of the path formulation. The solution $\overline{x}$ is optimal for the linear relaxation of the problem if and only if it satisfies all the security inequalities (4.19). In general, this is not the case. Therefore, violated security inequalities and valid inequalities are added to the restricted LP, by solving a subproblem called *separation problem*. The process is repeated until no more violated inequality is found. The final solution, is hence optimal for the linear relaxation of (5.4). If the solution is integral then it is optimal for the PCSP problem. If not, then we create new subproblems by branching on a fractional variable. The separation routine is then considered at each node of the tree and the process continue. Algorithm 4 gives the main phases of our Branch-and-Cut algorithm.

This algorithm uses the security inequalities and valid inequalities described in Chapter 5, whose separations are performed in the following order:

1) Security inequalities,
2) Path Covering inequalities,
3) Countermeasure Path inequalities,
4) Essential by Subsets Removing Countermeasure inequalities.

One can here remark that the inequalities to be separated are all global, that is they are valid in the whole Branch-and-Cut tree. In our Branch-and-Cut algorithm, we choose the following strategy of separation. At each separation procedure, we can add more than one violated inequality if there is any. Moreover, when separating the valid inequalities given above, we move to the separation of a new class of inequality only if no more violated inequalities of the current one is detected. We also choose to apply the cutting plane process for all the nodes of the Branch-and-Cut tree in order to get the best possible lower bound, and then limit the number of generated nodes in the tree.

In what follows, we describe the separation routines used for the inequalities mentioned above. Depending on the class of the valid inequality, we devise exact or heuristic procedures of separation. Before giving the separation procedures, we first present the feasibility test of a solution $x \in \mathbb{R}^K$ described in the following section.

Algorithm 4: Branch-And-Cut Algorithm

Data: An instance (G, K, D) of PCSP
Result: Optimal solution of PCSP

1 $LP \leftarrow LP_0$;
2 Solve the linear program LP ;
3 **if** *no (Security,Path Covering inequalities, Path Countermeasures inequalities,Essential by Countermeasures Subsets Removing) inequality is violated by* $\overline{x}$ **then**
4 go to 8;
5 **else**
6 Add all possible violated inequalities by $\overline{x}$;
7 go to 2;
8 **if** $\overline{x}$ *is integer* **then**
9 $\overline{x}$ is an optimal solution for PCSP. Stop ;
10 **else**
11 Create two sub-problems by branching on a fractional variable.
12 **forall the** *open sub-problem* **do**
13 go to 2;
14 **return** the best optimal solution of all the sub-problems.

6.1.3 Feasibility test

The path formulation of the PCSP problem is given with an exponential number of inequalities. In practice, these inequalities are not enumerated and are not all present in the initial LP (LP_0). As a consequence, an optimal solution of LP_0, although it is integer, may not necessary be feasible for the problem. This solution should, in fact, satisfy all the security inequalities. To check this, one should solve the separation problem for the basic security inequalities (6.1), which is detailed in the following section.

6.1.4 Separation problems and algorithms

6.1.4.1 Separation of security inequalities

Let $\bar{x}$ be the current solution. The separation problem of the security inequalities (4.19) reduces to a shortest path problem and can then be solved in polynomial time. In fact, if the shortest $s-t$ path $\tilde{P}$ in $G[\bar{x}]$ for $s \in S, t \in T$ with respect to $\bar{x}$ has a length strictly less than the d_s^t, then the security inequality associated with s, t, and $\tilde{P}$ is violated, otherwise there is no violated

security inequalities. This can be done using Dijkstra algorithm[54]. The separation algorithm for the security inequalities is detailed in Algorithm 5.

Algorithm 5: Separation of security inequalities

Data: Fractional Solution $\overline{x}$, an instance (G, K, D)
Result: Violated Security inequalities

```
Let I ← ∅ ;
/* denotes the set of Security inequalities violated by x̄ */
forall the s ∈ S do
    forall the t ∈ T do
        Calculate the value of L_{G_{K_x̄}}(P*_{st}) ;
        if L_{G_{K_x̄}}(P*_{st}) < d^t_s then
            /* there is a violated Security inequality */
            Denote I(P*_{st}) the violated inequality;
            I ← I ∪ I(P*_{st})
return the detected violated SNST inequalities I ;
/* I = ∅ if no violated Security inequalities are detected */
```

Recall that, Dijkstra algorithm has a complexity of $\mathcal{O}((\mid A \mid + \mid V \mid) log(\mid V \mid))$. As the total separation algorithm is carried out for each $s \in S$ and $t \in T$, this implies that we compute $\mid S \mid \times \mid T \mid$ shortest paths. As a consequence, the whole separation routine is in $\mathcal{O}((\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$ time.

In what follows we discuss the separation algorithms for the valid inequalities.

6.1.4.2 Separation of Path Covering inequalities

Let $(s, t) \in \Gamma$, $P \in P_{st}$ and $C_P^{s,t}$ a corresponding non sufficient countermeasure set, the corresponding path covering inequality is

$$\sum_{(v,l) \in K(P) \setminus C_P^{s,t}} x_v^l \geq 1.$$

Let now $\bar{x}$ be the current solution. The separation problem for the path covering inequality consists in finding $C_P^{s,t} \subseteq K(P)$ such that $\sum_{(v,l) \in C_P^{s,t}} \alpha_v^l < d_s^t - L_G(P)$ and $\sum_{(v,l) \in K(P) \setminus C_P^{s,t}} \bar{x}_v^l < 1$. For this, it suffices to find the set $C_P^{*s,t} \subseteq K(P)$ that verifies $\sum_{(v,l) \in C_P^{*s,t}} \alpha_v^l < d_s^t - L_G(P)$ and minimizes $\sum_{(v,l) \in K(P) \setminus C_P^{s,t}} \bar{x}_v^l$.

As minimizing $\sum_{(v,l)\in K(P)\setminus C_P^{s,t}} \bar{x}_v^l$ is equivalent to maximizing $\sum_{(v,l)\in C_P^{s,t}} \bar{x}_v^l$, the separation problem can be reformulated as: given $\bar{x}$ as a current solution,$(s,t) \in \Gamma$ and $P \in P_{st}$, find $C_P^{*s,t} \subseteq K(P)$ that verifies $\sum_{(v,l)\in C_P^{*s,t}} \alpha_v^l < d_s^t - L_G(P)$ and maximizes $\sum_{(v,l)\in C_P^{s,t}} \bar{x}_v^l$.

Theorem 6.1 *The separation problem of path covering inequalities is NP-Complete.*

Proof. We reduce the Knapsack problem to the separation problem. The knapsack problem is defined as: given a set I of n items with a profit p_i and a weight w_i for every item i, and a scalar W, the goal is to select a subset of items such that $\sum_{i\in I} w_i \leq W$ and maximizes $\sum_{i\in I} p_i$.

We set $P = (v_1, v_2, \ldots, v_n)$ and $K(P) = \{(v_i, l_i) : i \in I\}$, $d_s^t - L_G(P) = W$, $\bar{x}_{v_i}^{l_i} = p_i$, $\alpha_{v_i}^{l_i} = w_i$, the separation problem is nothing but a Knapsack problem. $\square$

As the problem is NP-Complete, we perform an heuristic to separate the path covering inequalities. The idea consists in sorting the countermeasures induced by the path P according to an increasing order of $\frac{\bar{x}_t^k}{\alpha_t^k}$. Next, we select the countermeasures in this order until a maximal non sufficient countermeasures set is found. Finally, we check if the corresponding inequality is violated. The heuristic is given in Algorithm 6.

We apply the heuristic described in Algorithm 6 for each shortest $s-t$ path P_{st}^*. Consequently, the separation problem complexity is in $\mathcal{O}(\mid S \mid \times \mid T \mid \times \mid V \mid)$

6.1.4.3 Separation of Countermeasures Path inequalities

In this section, we discuss the separation of countermeasures path inequalities (5.9). Let $J = \{(t_1, k_1), (t_1, k_1), \ldots, (t_n, k_n)\} \subseteq K \setminus K^*$ be a countermeasures path. The countermeasures path inequality associated to J is

$$\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \lceil \frac{n-1}{2} \rceil.$$

The separation problem for countermeasures path inequalities can be formulated as: given $\bar{x}$ as a current solution, find a countermeasures path $J =$

Algorithm 6: Separation heuristic of Path Covering Inequalities

Data: Fractional solution $\overline{x}$, an instance (G, K, D), $(s,t) \in \Gamma$ and $P \in P_{st}$

Result: Violated Path Covering Inequalities

Let $I \leftarrow \emptyset$;

`/* denotes the set of Path Covering Inequalities violated by` $\overline{x}$ `*/`

Sort the elements of $K(P)$ such that $\frac{\bar{x}_t^k}{\alpha_t^k} \geq \frac{\bar{x}_v^l}{\alpha_v^l}$;

while *No maximal not sufficient countermeasures set* $C_P^{s,t}$ *found* **do**

  Select a new countermeasure in the established order ;

if $\sum_{(v,l) \in K(P) \setminus C_P^{s,t}} \bar{x}_v^l < 1$ **then**

  `/* there is a violated Path Covering inequality */`

  Denote $I(C_P^{s,t})$ the violated inequality;

  $I \leftarrow I \cup I(C_P^{s,t})$

return the detected violated Path Covering Inequalities I ;

`/*` $I = \emptyset$ `if no Path Covering Inequalities are detected */`

$\{(t_1, k_1), (t_1, k_1), \ldots, (t_n, k_n)\}$ that minimizes $\sum_{i=1}^{n} \bar{x}_{t_i}^{k_i}$. We have the following result.

Theorem 6.2 *Inequalities* (5.9) *can be separated in polynomial time.*

Proof. An exact method to separate inequalities (5.9) can be described as follows.

As a first step and by using Algorithm 3 presented in Chapter 5 for finding essential countermeasures, we can show that we can find each countermeasures path J of size 2. The set J can be found by computing $(K \setminus \{(t,k)\})^*$ for each $(t,k) \in K \setminus K^*$. Each $(v,l) \in K \setminus K^*$ such that $(v,l) \in (K \setminus \{(t,k)\})^*$ forms with (t,k) a countermeasures path $J = \{(t,k),(v,l)\}$ of size 2. This can be obtained in $\mathcal{O}((\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$ time, by using Algorithm 3 with the instance $(G, K \setminus \{(t,k)\}, D)$. Next if $x_t^k + x_v^l < 1$, a violated countermeasures path inequality w.r.t. J is detected.

Now, consider the bipartite graph $\tilde{K} = (K' \cup K'', \tilde{A})$, as shown in Figure 6.1 as follows: For each $(t,k) \in K \setminus K^*$, we consider two vertices $(t',k') \in K'$ and $(t'',k'') \in K''$. For simplicity we refer to (t',k') by v' and (t'',k'') by v''. For each countermeasures path set $J = \{(t_1,k_1),(t_2,k_2) \in K \setminus K^* : (t_2,k_2) \in (K \setminus \{(t_1,k_1)\})^*\}$, we consider the arcs (v_1', v_2'') and (v_1'', v_2') with the same weight $z_{v_1 v_2} = x_{t_1}^{k_2} + x_{t_1}^{k_2} - 1$. Since $x_{t_1}^{k_2} + x_{t_1}^{k_2} \geq 1$, we have $z_{v_1 v_2} \geq 0$.

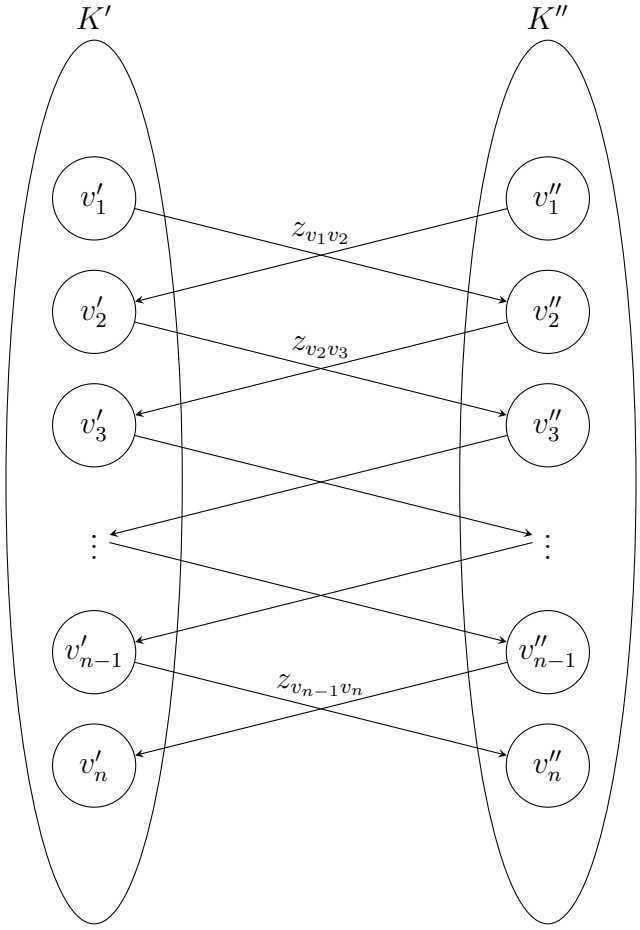


Figure 6.1: The bipartite graph $\tilde{K} = (K' \cup K'', \tilde{A})$ used for separating the Countermeasures Path inequalities

We can write

$$\begin{aligned}
&\sum_{i=1}^{n-1} z_{v_iv_{i+1}} = \sum_{i=1}^{n-1}(x_{t_i}^{k_i} + x_{t_{i+1}}^{k_{i+1}} - 1)\\
\Rightarrow &\sum_{i=1}^{n-1} z_{v_iv_{i+1}} = x_{t_1}^{k_1} + x_{t_n}^{k_n} + 2\sum_{i=2}^{n-1} x_{t_i}^{k_i} - (n-1)\\
\Rightarrow &\sum_{i=1}^{n-1} z_{v_iv_{i+1}} + x_{t_1}^{k_1} + x_{t_n}^{k_n} = 2\sum_{i=1}^{n} x_{t_i}^{k_i} - (n-1)\\
\Rightarrow &\sum_{i=1}^{n-1} z_{v_iv_{i+1}} + x_{t_1}^{k_1} + x_{t_n}^{k_n} + (n-1) = 2\sum_{i=1}^{n} x_{t_i}^{k_i}\\
\Rightarrow &\frac{1}{2}(\sum_{i=1}^{n-1} z_{v_iv_{i+1}} + x_{t_1}^{k_1} + x_{t_n}^{k_n} + (n-1)) = \sum_{i=1}^{n} x_{t_i}^{k_i}
\end{aligned}$$

As, $\sum_{i=1}^{n} x_{t_i}^{k_i} \geq \lceil \frac{n-1}{2} \rceil$, if n is even, we obtain

$$\frac{1}{2}(\sum_{i=1}^{n-1} z_{v_i v_{i+1}} + x_{t_1}^{k_1} + x_{t_n}^{k_n} + (n-1)) \geq \frac{n}{2}$$
$$\Rightarrow \sum_{i=1}^{n-1} z_{v_i v_{i+1}} + x_{t_1}^{k_1} + x_{t_n}^{k_n} \geq 1. \tag{6.6}$$

Therefore, separating inequalities (5.9) is equivalent to separating (6.6) .

Moreover, as $\tilde{K}$ is bipartite, every path from a vertex v_1' to a vertex v_2'' corresponds to a countermeasures path set of even size n.

For each pair of vertices $((t', k'), (v'', l'')) \in K' \times K''$, we compute a shortest path $P^*_{(t',k'),(v'',l'')}$ with respect to the weights z_{uv}. If $L_{G_{\bar{K}}}(P^*_{(t',k'),(v'',l'')}) \geq 1 - x_t^k - x_v^l$, then there is no violated inequality related to a countermeasure path between (t', k') and (v'', l''), otherwise the countermeasures path inequality associated to the path $P^*_{(t',k'),(v'',l'')}$ is violated. We use this for each pair $((t', k'), (v'', l'')) \in K' \times K''$. This algorithm is polynomial which ends the proof. □

Exact separation of the inequalities (5.9) can be done in $\mathcal{O}(\mid K \setminus K^* \mid^2 (\mid \tilde{A} \mid + 2 \mid K \setminus K^* \mid) log(2 \mid K \setminus K^* \mid)) + \mathcal{O}((\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$ time.

The separation algorithm is described in Algorithm 7.

6.1.4.4 Separation of essential – by subsets removing – countermeasure inequalities

We study in this section the separation of Essential – by Subsets Removing – Countermeasure inequalities (ESRC) (5.10). Let $(t, k) \in K \setminus K^*$, $K' \subseteq K \setminus (K^* \cup \{(t, k)\}$ of size n, and $2 \leq p \leq n$. For $q \in \{1, \ldots, n - p + 1\}$, the ESRC inequalities can be written as

$$q x_t^k + \sum_{(v,l) \in L} x_v^l \geq q \quad \text{for all } L \subseteq K', \mid L \mid = p + q - 1.$$

Let us refer to the current solution by $\overline{x}$. The separation problem associated to the ESRC inequalities can be defined as follows. find a non essential countermeasure $(t, k) \in K \setminus K^*$, a subset $K' \subseteq K \setminus (K^* \cup \{(t, k)\})$ of size n,

Algorithm 7: Separation of Countermeasures Path inequalities

Data: Fractional Solution $\overline{x}$, an instance (G, K, D)
Result: Violated Countermeasures Path inequalities

Let $I \leftarrow \emptyset$;
/* denotes the set of Countermeasures Path inequalities violated by $\overline{x}$ */
forall the $(t, k) \in K \setminus K^*$ **do**
  Calculate the set $(K \setminus \{(t, k)\})^*$ using Algorithm 5 ;
  forall the $(v, l) \in (K \setminus \{(t, k)\})^* \setminus K^*$ **do**
    if $x_t^k + x_v^l < 1$ **then**
      /* there is a violated Countermeasures Path inequality of size 2 */
      Denote $I_{(t,k)}^{(v,l)}$ the violated inequality;
      $I \leftarrow I \cup I_{(t,k)}^{(v,l)}$
Construct the graph $\tilde{K} = (K' \cup K'', \tilde{A})$;
forall the $(t', k') \in K'$ **do**
  forall the $(v'', l'') \in K''$ **do**
    if $L_{G_{\tilde{K}}}(P^*_{(t',k'),(v'',l'')}) < 1 - x_t^k - x_v^l$ **then**
      /* there is a violated Countermeasures Path inequality of size $n > 2$ */
      Denote $I_{(t,k)}^{(v,l)}$ the violated inequality;
      $I \leftarrow I \cup I_{(t,k)}^{(v,l)}$
return the detected violated Countermeasures Path inequalities I ;
/* $I = \emptyset$ if no Countermeasures Path inequalities are detected */

and an integer $2 \leq p \leq n$ such that $(t, k) \in (K \setminus L)^*$ for all $L \subseteq K', \mid L \mid = p$, and

$$(n - p + 1)x_t^k + \sum_{(v,l) \in K'} x_v^l < (n - p + 1).$$

We know by Theorem 5.32 that the separation problem for the ESRC inequalities is NP-Complete. In order to separate these inequalities, we use the heuristic described in Algorithm 8. The idea of the heuristic is to try to find an ESRC among the non essential countermeasures induced by the shortest $s - t$ paths P^*_{st} and for $p = 2, 3$. For each shortest $s - t$ path, one can choose the countermeasure $(t_0, k_0) \in K(P^*_{st}) \cap (K \setminus K^*)$ which has the greatest effect and fix $K' = (K(P^*_{st}) \cap (K \setminus K^*)) \setminus \{(t_0, k_0)\}$. Next, we check if (t_0, k_0) is ESRC w.r.t. K' and p. To this end, we start by setting $p = 2$. Then, we test

if the condition $(t_0, k_0) \in (K \setminus L)^*$ is verified for all $L \subseteq K'$ such that $\mid L \mid= 2$. If this is the case, then (t_0, k_0) is ESRC w.r.t. K' and 2. If not we repeat this process for $p = 3$. When an ESRC is found, we check if the associated inequality is violated for $q = n - p + 1$.

The separation algorithm of ESRC inequalities can be carried out in $\mathcal{O}((\binom{2}{|K'|} + \binom{3}{|K'|}) \mid K' \mid (\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$ time.

Algorithm 8: Separation of Countermeasures Path inequalities

Data: Fractional Solution $\overline{x}$, an instance (G, K, D), $p = 2$
Result: Violated ESRC inequalities

Let $I \leftarrow \emptyset$;
`/* denotes the set of ESRC inequalities violated by` $\overline{x}$ `*/`
forall the $(s, t) \in S \times T$ **do**
  Calculate the shortest $s - t$ path P^*_{st} ;
forall the $(s, t) \in S \times T$ **do**
  Essential $\leftarrow$ *True* ;
  while $p = 2, 3$ *and Essential and not Finish* **do**
    $L \leftarrow \{I \subseteq K' :\mid I \mid= p\}$;
    while $L \neq \emptyset$ **do**
      if $(t_0, k_0) \in (K \setminus L[0])^*$ **then**
        $L \leftarrow L \setminus L[0]$;
      else
        Essential $\leftarrow$ *False*;
    if $L = \emptyset$ *or not Essential* **then**
      Finish $\leftarrow$ *True*;
if *Finish and* $(n - p + 1)x^k_t + \sum_{(v,l)\in K'} x^l_v < (n - p + 1)$ **then**
  `/* there is a violated ESRC inequality */`
  Denote $I^p_{K'}$ the violated inequality;
  $I \leftarrow I \cup I^p_{K'}$
return the detected violated ESRC inequalities I ;
`/*` $I = \emptyset$ `if no Countermeasures Path inequalities are detected */`

6.1.5 Implementation's features

During the separation procedures, in order to efficiently deal with the violated inequalities that are added, we create particular data structure called *pools* whose size increases dynamically. All the generated inequalities are dynamic

and stored in a specific pool, that is to say they are removed from the current linear program if they are not active. At each iteration, the separation procedure begins first by detecting violated inequalities in the pool. If no such inequality exits, then we carry out our separation procedure on the valid inequalities in the order given before.

6.1.6 Branching strategy

Let $(\mathcal{P})$ denote the linear program of a given node in the Branch-and-Cut tree. Suppose that the optimal solution of the linear relaxation of $(\mathcal{P})$ is fractional. Denote by $\overline{x}$ this fractional solution. The branching phase consists in choosing a fractional variable $\overline{x}_t^k$, $t \in T, k \in K$, and then create two subproblems $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ by adding respectively the constraints $x_t^k \leq \lfloor \overline{x}_t^k \rfloor$ and $x_t^k \geq \lceil \overline{x}_t^k \rceil$. As the decision variables for the PCSP problem are binary, this reduces to fix the variable x_t^k either to 0 or to 1.

There are several strategies used to select the fractional variable on which we choose to branch. In our case, we have chosen the strategy introduced by Padberg and Rinaldi [113] for the Symmetric Travelling Salesman Problem. This strategy consists in choosing the most fractional variable, that is the fractional variable which is the nearest to 0.5. If there exist many variables having the same fractional value, and satisfying this condition, then we choose the most weighted one in the objective function.

6.1.7 Primal heuristic

To accelerate the Branch-and-Cut algorithm and enable a fast pruning of uninteresting branches of the tree, we propose a primal heuristic. Given a fractional solution, we try to obtain a feasible solution for the PCSP problem by rounding as shown in Algorithm 9. To this end, we use the following result.

Proposition 6.3 *Let $\bar{x}$ be a fractional solution. Then, $x = \lceil \bar{x} \rceil$ is a feasible solution of PCSP.*

Proof. Let $(s,t) \in \Gamma$ and $P \in P_{st}$, it suffices to prove that $\sum_{(v,l)\in K(P)} \alpha_v^l x_v^l \geq d_s^t - L_G(P)$. As $\lceil \bar{x}_v^l \rceil \geq x_v^l$ for all $(v,l) \in K$, we can write

$$\begin{aligned}\sum_{(v,l)\in K(P)} \alpha_v^l x_v^l &\geq \sum_{(v,l)\in K(P)} \alpha_v^l \bar{x}_v^l, \\ &\geq d_s^t.\end{aligned}$$

□

Algorithm 9: Primal Heuristic

Data: Fractional Solution $\bar{x}$
Result: Integer Feasible Solution x
forall the $i \in \{1, \dots \mid K \mid\}$ **do**
  $x_i = \lceil \bar{x}_i \rceil$
return the integer feasible solution x ;

Based on these results, we devise a Branch-and-Cut algorithm that we have tested on random and realistic instances. The results we obtained are discussed in the next sections.

6.2 Computational study

Before discussing the experimental results, we present the tools that we have used for the implementation.

The Branch-and-Cut algorithm described in the previous chapter has been implemented in Python 2.7 [22] using the solver CPLEX 12.8 [2]. The Branch-and-Cut algorithm have been tested on Intel(R) Xeon(R) CPU E5-2603 v3 1.60GHz with 126Go of RAM, running under Linux. The Python package Networkx [16] has been used for the creation and the manipulation of graphs. Python Pandas library [18] has been used for results data manipulation and analysis and matplotlib [15] for results data visualization.

The maximum CPU time is fixed to 5 hours. The results are reported in the tables that will be presented in the sequel. The entries of the various tables are the following:

$\|V\|$	:	number of node in the graph G;
$\|S\|$	:	number of access points;
$\|T\|$	:	number of asset-vulnerability nodes;
$\|A\|$	:	number of arcs;
$\|K\|$	:	number of countermeasures;
$\|\Gamma\|$	:	number of attacks;
p	:	the probability of edges of the Erdős - Rényi graph induced by T;
I_x	:	name of the random instance, where x is either p or (S, T);
$\|K^*\|$	:	number of essential countermeasures;
N	:	number of nodes in the Branch & Cut tree;
OI	:	number of generated optimality conditions inequalities;
Sec	:	number of generated security inequalities;
PCI	:	number of generated Path Covering inequalities;
$CmPI$	:	number of generated Countermeasures Path inequalities;
Opt	:	the value of the optimal solution;
$NOpt$	:	the number of instances solved to optimality / total number of instances;
Gap	:	the relative error between the best upper bound (the optimal solution if the problem has been solved to optimality) and the lower bound obtained at the root,
CPU	:	total CPU time (in the format hh:mm:ss).

In order to discuss the efficiency of the optimality conditions and valid inequalities, we will conduct comparison study of the Branch-and-Cut algorithm with and without these inequalities. We will refer by *the basic formulation*, the path formulation PCSP2 (5.4) without including the optimality condition inequalities (4.21) and without considering the valid inequalities (5.8) and (5.9).

Note that the ESRC inequalities (5.10) have numerically shown not very efficient in strengthening the formulation and improving the execution time. For this reason, we will not consider these inequalities in the Branch-and-Cut algorithm.

We have conducted experimentations on random and realistic instances described in the next sections.

6.2.1 Random instances

Before presenting our numerical results, we first describe the random instances we used.

6.2.1.1 Description

In order to vary the type of random instances, we generate instances with different densities and sizes of S and T. Then, for each fixed $\mid S \mid$ and $\mid T \mid$, we generate a PCSP instance (G, K, D) as follows:

- The sub-graph induced by the set of nodes T is an Erdős - Rényi random graph [59] of parameters $|T|$ and p, where p is the probability of existence of an arc in the graph.
- We connect each access point in S to a number of nodes in T chosen uniformly between 1 and $\mid T \mid$.
- We uniformly generate a positive weight for each arc in the interval $[0, 100]$.
- We set the same threshold d_s^t chosen randomly in the interval $[0, 100]$ for all $(s, t) \in S \times T$ which is randomly.
- Each node has a number of countermeasures between 0 and 10 chosen uniformly in the set of countermeasures described in Table 6.1.

Name	effect	cost
k_1	10	5
k_2	20	10
k_3	30	20
k_4	40	30
k_5	50	40
k_6	60	90
k_7	70	80
k_8	80	70
k_9	90	60
k_{10}	100	50

Table 6.1: The set of countermeasures

This random procedure for generating instances will be used to construct two families of instances denoted F_p and $F_{S,T}$:

- F_p: we fix $\mid S \mid = 50$, $\mid T \mid = 100$ and vary the parameter p in $\{0.1, 0.2, \ldots, 1\}$. Then, for each p we randomly generate as described above a family of five instances that will be denoted by I_p.
- $F_{S,T}$: we fix $p = 0.3$ and vary at the same time the number of nodes $\mid S \mid$ and $\mid T \mid$. For each $\mid S \mid$ and $\mid T \mid$, a family of five instances denoted by $I_{S,T}$ is randomly generated as previously described.

For each family of five instances. , we will report the average results.

6.2.1.2 F_p random instances

Let us now discuss the numerical results for the F_p family.

In Table 6.2 we present numerical results obtained by the Branch-and-Cut using the basic formulation and numerical results obtained by the Branch-and-Cut including the optimality conditions in the preprocessing phase.

We can observe that with the basic formulation none of the instance of the families are solved to optimality within the time limit, except $I_0.2$ and $I_0.3$ for which only one instance out of five has been solved to optimality.

On the other hand, the numerical tests of the path formulation with the optimality condition inequalities show that 34 instances are solved to optimality compared to 2 instances with the basic formulation which is PCSP2 without including the new inequality. For example by considering the basic formulation, the algorithm could not solve any of the instances of the family $I_0.7$. However these have been all solved to optimality within 3 hours when the optimality condition are added.

We also observe that the Gap and the number of nodes in the Branch-and-Cut tree have significantly decreased by adding the optimality condition inequalities in the preprocessing phase. This is the case of the families $I_0.2$ and $I_0.3$. This shows the efficiency of the optimality condition inequalities in our Branch-and-Cut algorithm.

Now let us discuss the impact of the valid inequalities without the optimality conditions. In Table 6.3, we present the numerical results obtained by the Branch-and-Cut algorithm with and without the valid inequalities.

As we can see, by adding the valid inequalities, we could solve many instances to optimality compared to the case with only optimality conditions.

					basic formulation					Branch-and-Cut with optimality conditions					
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*OI*	*N*	*Gap*	*CPU*	*NOpt*
I_0.1	3401.2	1315.4	534	7.2	12	3976.8	0.67	-	0/5	21.4	2114	669.8	0.34	4:31:7	4/5
I_0.2	4293	988.6	614.6	12.2	21.2	2928	0.46	4:01:3	1/5	11.8	2344.2	603	0.34	3:16:9	4/5
I_0.3	5007.6	1428.2	452	6.2	15.2	1763	0.33	3:22:4	1/5	13.4	2198.4	587.8	0.16	2:39:1	4/5
I_0.4	5798	1253	983	7.4	14.4	4002.6	0.68	-	0/5	15	2009.2	554.6	0.26	2:57:1	4/5
I_0.5	6566.8	1007.8	547.2	17.6	22	4555	0.59	-	0/5	17.2	2206	706.4	0.45	3:03:4	4/5
I_0.6	7343.4	2309	477	6.4	11.6	4230.6	0.78	-	0/5	17.4	2148.2	664.4	0.37	2:45:6	3/5
I_0.7	8015.4	450.6	876.4	2	21.2	4265.2	0.80	-	0/5	15.4	2101	802.4	0.22	3:09:4	5/5
I_0.8	8764.6	910.8	681.2	10.2	32.8	4884.4	0.89	-	0/5	22.8	2534.2	763.8	0.38	2:13:6	3/5
I_0.9	9310.2	1322.8	622.8	6.6	10.2	4660.8	0.63	-	0/5	19.6	2005.4	859.2	0.29	3:02:8	3/5
I_1.0	11546.2	1479.2	596.2	5	10	4256.2	0.73	-	0/5	9	1998	666	0.44	3:15:9	2/5

Table 6.2: Efficiency of optimality conditions in solving F_p

In fact for eight instance families, four out of five instances are solved to optimality. However, by considering the basic formulation, we could solve four out five instances only for five instance families. We can also observe the five instances of the family $I_0.4$ are solved to optimality by considering the valid inequalities which is not the case when considering only the optimality conditions. Moreover, the valid inequalities have a stronger impact than the optimality condition inequalities in improving the average lower bound, the average number of nodes in the Branch-and-Cut tree and the average CPU time.

We can also remark that, the average number of path covering inequalities PCI is greater than the average number of countermeasures path inequalities CmPI.

Moreover, the average number of security inequalities by considering valid inequalities is less than the average number of security inequalities generated by only considering optimality condition. This is caused by the fact that some security inequalities are dominated by valid inequalities and in particular by the path covering inequalities.

In Table 6.4, we report the results obtained by the Branch-and-Cut algorithm using optimality condition inequalities and valid inequalities together. It appears that this is more efficient than considering each of them separately. Indeed, by considering only valid inequalities, the fact of solving five instances out of five to optimality has been obtained only for two instance families $I_0.4$ and $I_0.7$ within an average time of 2:09:2 and 2:59:3, respectively. However, by considering both optimality condition inequalities and valid inequalities, this has been obtained for seven instance families within an average time be-

					basic formulation					Branch and Cut with valid inequalities						
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_0.1	3401.2	1315.4	534	7.2	12	3976.8	0.67	-	0/5	2.8	136.8	12.2	269.4	0.31	4:01:2	4/5
I_0.2	4293	988.6	614.6	12.2	21.2	2928	0.46	4:01:3	1/5	9.4	146	3.3	303.4	0.31	2:55:3	4/5
I_0.3	5007.6	1428.8	452	6.2	15	1763	0.33	3:22:4	1/5	3.6	198.2	10.6	387	0.16	2:02:4	4/5
I_0.4	5798	1253	983	7.4	14.4	4002.6	0.68	-	0/5	7	239.4	8.2	254	0.20	2:09:2	5/5
I_0.5	6566.8	1007.8	547.2	17.6	22	4555	0.59	-	0/5	9.4	206.4	1.2	306.4	0.38	2:33:6	4/5
I_0.6	7343.4	2309	477	6.4	11.6	4230.6	0.78	-	0/5	5.8	448.2	5.2	598.6	0.37	2:35:7	4/5
I_0.7	8015.4	450.6	876.4	2	21.2	4265.2	0.80	-	0/5	5.8	387.8	6.6	302	0.20	2:59:3	5/5
I_0.8	8764.6	910.8	681.2	10.2	32.8	4884.4	0.89	-	0/5	2.6	589.6	2.2	389.2	0.37	2:09:3	4/5
I_0.9	9310.2	1322.8	622.8	6.6	10.2	4660.8	0.63	-	0/5	9.2	405.4	3.6	372.4	0.23	2:55:1	4/5
I_1.0	11546.2	1479.2	596.2	5	10	4256.2	0.73	-	0/5	9.6	998	7.6	467.2	0.39	3:05:9	4/5

Table 6.3: Efficiency of valid inequalities in solving f F_p

tween 0:38:4 and 1:22:5. In addition, as we can see in Table 6.4 and Table 6.3 the average Gap and the average number of nodes are better for all the instance families when considering valid inequalities and optimality condition inequalities together. For instance, the maximum Gap obtained only with valid inequalities is 0.39. On the other hand, the one obtained by considering the optimality condition inequalities with the valid inequalities is 0.17.

					Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_0.1	3401.2	1315.4	534	7.2	2.8	2114.2	55	12.2	103.6	0.12	1:21:2	4/5
I_0.2	4293	988.6	614.6	12.2	5.4	2344.6	76.4	3.3	189	0.11	0:43:7	5/5
I_0.3	5007.6	1428.8	452	6.2	2.4	2198.6	88.4	10.6	148.2	0.11	1:22:5	5/5
I_0.4	5798	1253	983	7.4	5.4	2009	134.2	8.2	84.4	0.09	1:02:5	5/5
I_0.5	6566.8	1007.8	547.2	17.6	9.2	2206.6	133.6	1.2	96.4	0.14	0:44:8	5/5
I_0.6	7343.4	2309	477	6.4	5.2	2148.2	248.8	5	138.4	0.17	1:05:4	4/5
I_0.7	8015.4	450.6	876.4	2	4	2101.6	387.2	6.6	104.2	0.08	1:04:5	5/5
I_0.8	8764.6	910.8	681.2	10.2	1.6	2534.4	344.6	2.2	143.4	0.14	0:49:5	4/5
I_0.9	9310.2	1322.8	622.8	6.6	7.4	2005.6	305.2	3.6	72.2	0.06	1:09:3	5/5
I_1.0	11546.2	1479.2	596.2	5	6.4	1998.4	812.2	7.6	107	0.12	0:38:4	5/5

Table 6.4: Efficiency of the Branch and Cut algorithm in solving F_p

In Table 6.5, we can see that solving the path formulation using the Branch-and-Cut algorithm is better than solving the compact formulation directly using Cplex with its cuts and its presolve features. In fact, our Branch-and-Cut algorithm solves five instances out of five to optimality for seven instance families, and four out of five instances for the renaming three families. However, with the compact formulation solving five instances out of five is obtained only for two instance families ($I_0.4$ and $I_0.3$). In addition, none of

the instances of the families $I_0.9$, $I_1.0$ are solved to optimality. Moreover, with the path formulation, we can also notice a better average CPU time and a better average Gap.

We can remark that though the performances of the path formulation is better than the compact one, the average number of nodes obtained by solving the compact formulation using Cplex is less than the average number of nodes obtained by solving the path formulation using the Branch-and-Cut algorithm. This is caused by the huge size of variables of the compact formulation in comparison with the path formulation. As a consequence, in the resolution tree, the time of solving a node by Cplex with the compact formulation becomes greater than the time of solving a node by the Branch-and-Cut algorithm.

					Compact formulation				Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_0.1	3401.2	1315.4	534	7.2	84.4	0.19	2:33:4	4/5	2.8	2114.2	55	12.2	103.6	0.12	1:21:2	4/5
I_0.2	4293	988.6	614.6	12.2	128.8	0.46	4:01:3	4/5	5.4	2344.6	76.4	3.3	189	0.11	0:43:7	5/5
I_0.3	5007.6	1428.8	452	6.2	76.8	0.13	1:57:9	5/5	2.4	2198.6	88.4	10.6	148.2	0.11	1:22:5	5/5
I_0.4	5798	1253	983	7.4	101.2	0.14	1:58:3	5/5	5.4	2009	134.2	8.2	84.4	0.09	1:02:5	5/5
I_0.5	6566.8	1007.8	547.2	17.6	94	0.23	2:01:1	3/5	9.2	2206.6	133.6	1.2	96.4	0.14	0:44:8	5/5
I_0.6	7343.4	2309	477	6.4	99 .6	0.28	2:23:4	3/5	5.2	2148.2	248.8	5	138.4	0.17	1:05:4	4/5
I_0.7	8015.4	450.6	876.4	2	45.4	0.15	1:43:7	2/5	4	2101.6	387.2	6.6	104.2	0.08	1:04:5	5/5
I_0.8	8764.6	910.8	681.2	10.2	82	0.14	0:56:3	1/5	1.6	2534.4	344.6	2.2	143.4	0.14	0:49:5	4/5
I_0.9	9310.2	1322.8	622.8	6.6	161	0.58	-	0/5	7.4	2005.6	305.2	3.6	72.2	0.06	1:09:3	5/5
I_1.0	11546.2	1479.2	596.2	5	-	-	-	0/5	6.4	1998.4	812.2	7.6	107	0.12	0:38:4	5/5

Table 6.5: Comparison with the compact formulation for F_p

We also observe that when p increases, which implies that the number of arcs increases, the problem becomes harder to solve using the compact formulation. However, with the Branch-and-Cut algorithm, we could solve almost all the instances within a CPU time not exceeding one hour and half. In fact, we can observe in Table 6.5 that with the compact formulation, as p increases, the fewer is the number of instances that are solved to optimality. On the other hand, the path formulation's performances remain good the same whatever the value of p. This is due to the size of the two formulations. Indeed, the number of variables of the compact formulation depends on the number of arcs which increases when p increases. However, the number of variables of the path formulation only depend on the number of countermeasures.

It is clear that the Branch-and-Cut algorithm is very efficient in solving the family of instances F_p. In the next section, we study the efficiency of our algorithm to solve the family $F_{S,T}$ and the sensitivity to the size of nodes.

6.2.1.3 $F_{S,T}$ random instances

We discuss now the numerical results for the $F_{S,T}$ random instances.

Table 6.6 illustrates the numerical results obtained by the basic formulation and the numerical results obtained by path formulation including optimality condition inequalities. As we can see the problem is hard to solve with the basic formulation. In fact, only the first three families $I_10,100$, $I_20,200$ and $I_30,300$ contain some instances that are solved to optimality within the CPU time limit. On the other hand, none of the instances in the remaining seven instance families have been solved to optimality within the time limit. This can be explained by the separation time of security inequalities and the size of the formulation. In fact, for these instances $\mid S \mid$, $\mid T \mid$ and $\mid \Gamma \mid$ are large ($\mid \Gamma \mid$ between 1009.6 and 21907.6). As a consequence, the separation security inequalities is time consuming as it reduces to compute $\mid \Gamma \mid$ shortest paths in $\mathcal{O}((\mid A \mid + \mid V \mid) \mid S \mid \times \mid T \mid \times log(\mid V \mid))$ time. Moreover, $\mid K \mid$ (which is the number of variables of the path formulation) is quiet large (between 1613 and 5503.6). This together with the large separation time implies that no instance for the last six families has been solved to optimality.

We can see in Table 6.6 that introducing the optimality condition inequalities has a positive impact in solving the problem. By considering these inequalities, we could solve 24 instances to optimality including some instances of large size such that $I_80,800$ and $I_90,900$. However, the number of instances solved to optimality without optimality condition inequalities is 14.

					basic formulation					Branch-and-Cut with optimality conditions					
Name	$\lvert A\rvert$	$\lvert\Gamma\rvert$	$\lvert K\rvert$	$\lvert K^*\rvert$	*Sec*	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*OI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	15.4	163.8	0.14	0:21:5	5/5	32	694.4	82	0.09	0:07:2	5/5
I_20,200	500.2	627.2	667.2	6.4	12.4	252	0.13	0:43:7	5/5	28.6	932.2	203.6	0.11	0:29:3	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	44.4	1320	0.17	1:43:7	3/5	12.8	1027.8	449.6	0.10	1:09:7	4/5
I_40,400	1864.8	11910.4	981.8	37.4	23	833	0.17	3:53:9	1/5	23.8	2555.6	372.8	0.14	3:24:5	2/5
I_50,500	2676.4	1009.6	1674.4	30.8	25	1530.6	0.42	-	0/5	62.6	4170.4	563.6	0.18	3:59:3	2/5
I_60,600	3276.2	2988.2	1613	40.2	42	1299.8	0.57	-	0/5	35.4	3987.4	789	0.17	4:02:4	2/5
I_70,700	3988.4	2534.8	1876.2	27.8	17.2	1076.8	0.44	-	0/5	25.8	4456.6	858.2	0.12	4:43:7	2/5
I_80,800	4694	2987.2	2854.4	63.2	31.4	1288.2	0.49	-	0/5	45	4483	1152	0.18	3:52:7	1/5
I_90,900	6204.6	1910.4	2765.4	52.2	13.6	1150.8	0.48	-	0/5	33	12019	1008	0.17	4:09:3	1/5
I_100,1000	10866.4	7479.4	3596	53.8	38.6	877.8	0.62	-	0/5	12	15888.8	830.2	0.55	-	0/5
I_110,1100	14444.6	15986.8	4987.2	67.2	10	480	0.59	-	0/5	22.8	14327.2	502.2	0.52	-	0/5
I_120,1200	19546.6	21907.6	5503.6	55	7.2	340.2	0.61	-	0/5	13.8	14230.6	290.8	0.47	-	0/5

Table 6.6: Efficiency of optimality conditions in solving $F_{S,T}$

Now let us discuss the impact of considering the valid inequalities without adding the optimality condition inequalities. Table 6.7 gives the numerical results obtained by solving the basic formulation and the numerical results

obtained by solving the path formulation with valid inequalities. It is clear that the valid inequalities are stronger than the optimality conditions in improving the problem resolution. In fact, as seen in Table 6.7, considering the valid inequalities allows us to solve five more instance including two instances of large size in the families $I_80, 800$ and $I_90, 900$.

Note that generally considering the valid inequalities in the resolution of the path formulation is better than adding the optimality condition inequalities. However, for some specific cases such that the instance in $I_80, 800$, optimality condition inequalities can be more efficient (but not significantly) than valid inequalities. This instance is solved within 3:52:7 when optimality condition inequalities are added and within 4:02:7 when valid inequalities are considered.

					basic formulation					Branch and Cut with valid inequalities						
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	15.4	163.8	0.14	0:21:5	5/5	2.2	14.6	0	43	0.07	0:03:3	5/5
I_20,200	500.2	627.2	667.2	6.4	12.4	252	0.13	0:43:7	5/5	22.8	98.6	12	193.8	0.09	0:22:3	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	44.4	1320	0.17	1:43:7	3/5	28.2	643.8	12.6	277.2	0.11	2:09:7	5/5
I_40,400	1864.8	11910.4	981.8	37.4	23	833	0.17	3:53:9	1/5	21.2	944.2	22.2	173.8	0.13	4:24:5	3/5
I_50,500	2676.4	1009.6	1674.4	30.8	25	1530.6	0.42	-	0/5	36.2	365.4	7	187.2	0.12	3:52:3	3/5
I_60,600	3276.2	2988.2	1613	40.2	42	1299.8	0.57	-	0/5	25	1287.6	8.6	189.2	0.10	3:42:7	3/5
I_70,700	3988.4	2534.8	1876.2	27.8	17.2	1076.8	0.44	-	0/5	35.2	434.4	18.8	156	0.10	4:12:6	3/5
I_80,800	4694	2987.2	2854.4	63.2	31.4	1288.2	0.49	-	0/5	33	112	7.4	1237	0.18	4:02:7	1/5
I_90,900	6204.6	1910.4	2765.4	52.2	13.6	1150.8	0.48	-	0/5	31	745	6	169.6	0.13	4:29:5	2/5
I_100,1000	10866.4	7479.4	3596	53.8	38.6	877.8	0.62	-	0/5	42.8	367.6	7	430.6	0.46	-	0/5
I_110,1100	14444.6	15986.8	4987.2	67.2	10	480	0.59	-	0/5	27.4	450	13	396.6	0.43	-	0/5
I_120,1200	19546.6	21907.6	5503.6	55	7.2	340.2	0.61	-	0/5	34.8	340.2	7.2	230.8	0.41	-	0/5

Table 6.7: Efficiency of valid inequalities in solving $F_{S,T}$

Table 6.8 shows that considering both the optimality condition inequalities and the valid inequalities is more efficient than considering each of them separately. In fact as it appears, we could solve five more instances than the case where we only consider valid inequalities. In particular, we could solve two instances of large size in $I_100, 1000$ and $I_110, 1110$ that have not been solved with the valid inequalities and the optimality condition inequalities together. In addition, as we can see, by including both valid inequalities and optimality condition inequalities we improve the CPU time, the Gap and the number of nodes in the Branch-and-Cut tree for all the instance families.

We present in Table 6.9 the numerical results obtained by solving the compact formulation using Cplex and the numerical results obtained by solving the path formulation using the Branch-and-Cut algorithm with optimality condition inequalities and valid inequalities. Clearly, the Branch-and-Cut algorithm for the path formulation is more performant than Cplew for the

					Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	2.2	694.4	24.6	0	43.6	0.06	0:01:2	5/5
I_20,200	500.2	627.2	667.2	6.4	2.8	932.2	98.6	12	153	0.08	0:12:2	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	8.2	1027.8	643.8	12.6	177	0.08	0:49:3	5/5
I_40,400	1864.8	11910.4	981.8	37.4	1.2	2555.6	944.2	22.2	63.8	0.09	3:49:3	3/5
I_50,500	2676.4	1009.6	1674.4	30.8	16.2	4170.4	365.4	7	157.2	0.09	3:22:3	4/5
I_60,600	3276.2	2988.2	1613	40.2	5	3987.4	476.2	8.6	79.8	0.10	1:52:3	3/5
I_70,700	3988.4	2534.8	1876.2	27.8	15.2	4456.6	434.4	18.8	104	0.09	2:57:6	3/5
I_80,800	4694	2987.2	2854.4	63.2	15.4	4483	629.8	7.4	92.6	0.13	3:52:7	2/5
I_90,900	6204.6	1910.4	2765.4	52.2	11	12019	761.8	6	79	0.12	3:19:5	3/5
I_100,1000	10866.4	7479.4	3596	53.8	6	15888.8	613	7	107	0.19	4:42:3	1/5
I_110,1100	14444.6	15986.8	4987.2	67.2	12	14327.2	752	13	103	0.23	4:49:5	1/5
I_120,1200	19546.6	21907.6	5503.6	55	7.2	14230.6	644.4	7.2	69	0.37	-	0/5

Table 6.8: Efficiency of the Branch and Cut algorithm in solving $F_{S,T}$

compact formulation. In fact, the Branch-and-Cut algorithm solves 23 additional instances to optimality. It also gives a better average Gap and reduces significantly the average CPU time as it is the case for the families $I_10,100$ and $I_20,200$.

					Compact formulation				Branch and Cut							
Name	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*N*	*Gap*	*CPU*	*NOpt*	*Sec*	*OI*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*NOpt*
I_10,100	139.2	515.2	201	3.8	34	0.07	0:1: 5	5/5	2.2	694.4	24.6	0	43.6	0.06	0:01:2	5/5
I_20,200	500.2	627.2	667.2	6.4	70.8	0.09	1:38:2	5/5	2.8	932.2	98.6	12	153	0.08	0:12:2	5/5
I_30,300	1103.6	2852.4	1100.8	16.2	116	0.16	4:10:5	1/5	8.2	1027.8	643.8	12.6	177	0.08	0:49:3	5/5
I_40,400	1864.8	11910.4	981.8	37.4	120.2	0.39	-	0/5	1.2	2555.6	944.2	22.2	63.8	0.09	3:49:3	3/5
I_50,500	2676.4	1009.6	1674.4	30.8	138	0.23	3:41:1	1/5	16.2	4170.4	365.4	7	157.2	0.09	3:22:3	4/5
I_60,600	3276.2	2988.2	1613	40.2	88.2	0.41	-	0/5	5	3987.4	476.2	8.6	79.8	0.10	1:52:3	3/5
I_70,700	3988.4	2534.8	1876.2	27.8	57.8	0.49	-	0/5	15.2	4456.6	434.4	18.8	104	0.09	2:57:6	3/5
I_80,800	4694	2987.2	2854.4	63.2	68.2	0.51	-	0/5	15.4	4483	629.8	7.4	92.6	0.13	3:52:7	2/5
I_90,900	6204.6	1910.4	2765.4	52.2	54	0.45	-	0/5	11	12019	761.8	6	79	0.12	3:19:5	3/5
I_100,1000	10866.4	7479.4	3596	53.8	-	-	-	0/5	6	15888.8	613	7	107	0.19	4:42:3	1/5
I_110,1100	14444.6	15986.8	4987.2	67.2	-	-	-	0/5	12	14327.2	752	13	103	0.23	4:49:5	1/5
I_120,1200	19546.6	21907.6	5503.6	55	-	-	-	0/5	7.2	14230.6	644.4	7.2	69	0.37	-	0/5

Table 6.9: Comparison with the compact formulation for $F_{S,T}$

Furthermore, when $\mid S \mid$ and $\mid T \mid$ increase, the problem becomes harder to solve either using the compact formulation or using the path formulation. Indeed, we can see that we could solve fewer instances when we consider larger size of S and T. In fact, by construction of our random instances, if $\mid S \mid$ and $\mid T \mid$ increase, then so are $\mid K \mid$ and $\mid A \mid$. Hence, we can obtain a compact formulation with a huge number of variables and constraints. That's why, in Table 6.9 we observe that none of the instances of the families I_60_600 until I_120_1200 are solved to optimality within the time limit.

The Branch-and-Cut algorithm solves the instances I_100_1000 and I_110_1100 within almost five hours but the number of nodes is quite small. Therefore, the time for treating a node in the Branch-and-Cut tree is large. This is caused by the fact that for these two instances Γ is large which implies that the separation algorithm for security inequalities and path covering inequalities takes more time.

6.2.2 Realistic instances

Before discussing the numerical results, let us first describe our family of realistic instances.

The realistic instances we consider are obtained from the *SNDlib* [25] library which is a library of test instances for Survivable Network Design.

The subgraph induced by T is chosen as SNDlib graph. We set $\mid S \mid= 10$ and we randomly connect each $s \in S$ to some nodes in T of size between 1 and 5. The arc weights are chosen randomly in the interval $[1, 5]$. We fix the same threshold d_s^t in the interval $[20, 40]$ for each couple $(s, t) \in S \times T$. For each asset-vulnerability node, we associate exactly 5 countermeasures chosen randomly among a set of 30 countermeasures described in Table 6.10.

Let us now discuss,the efficiency of the Branch-and-Cut algorithm in solving the realistic instances.

In Table 6.11, we present the numerical results obtained by solving the basic formulation and the ones obtained by solving the path formulation with optimality condition inequalities.

We can see that optimality condition inequalities have a positive impact in solving the problem, but they are still not very efficient in strengthening the formulation. In fact, for the critical instances (pioro40, janow-us, cost266 and sun), the Gap generated by the path formulation with optimality condition inequalities remains the same as the one given by the basic formulation. In addition, the CPU time as well as the number of nodes in the Branch-and-Cut tree do not significantly decrease. For instance, for sun the CPU time decreases from 2:40:24 to 2:26:39 and the number of nodes decreases from 77364 to 73674. Consequently, the impact of optimality conditions inequalities in solving the problem for $F_{S,T}$ is not strong enough. However, we will see that this is not the case for valid inequalities.

In table 6.12, we report the numerical results obtained by solving the basic formulation and the ones obtained by solving the path formulation with valid inequalities. Clearly, valid inequalities are more efficient than optimality condition inequalities in solving the realistic instances. In fact, by considering

Name	effect	cost
k1	0.5	1
k2	1	10
k3	0.8	100
k4	3	50
k5	2	150
k6	5	300
k7	4	250
k8	7	200
k9	6	400
k10	9	350
k11	8	450
k12	11	650
k13	10	550
k14	13	600
k15	12	500
k16	15	700
k17	14	800
k18	17	750
k19	16	1000
k20	19	900
k21	18	950
k22	21	850
k23	20	1050
k24	23	1250
k25	22	1150
k26	25	1200
k27	24	1100
k28	27	1400
k29	26	1350
k30	29	1450

Table 6.10: Set of countermeasures for the realistic instances

							basic formulation				Branch-and-Cut with optimality conditions					
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*Sec*	*OI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	4	11	193	0.08	0:00:02	11	20	90	0.07	0:00:01	10360
janos-us	36	26	94	260	130	0	8	28559	0.09	0:52:35	7	18	24653	0.09	0:45:10	13201
nobel-germany	27	17	36	49	85	5	2	1405	0.11	0:00:26	2	9	1398	0.11	0:00:30	14060
dfnwin	20	10	55	61	50	1	1	272	0.07	0:00:07	2	16	150	0.07	0:00:04	10400
pioro40	50	40	99	54	200	0	11	16646	0.11	0:16:01	9	76	6079	0.11	0:05:39	14050
india35	45	35	90	50	175	0	4	420	0.11	0:00:18	4	39	318	0.11	0:00:17	10460
cost266	47	37	67	51	185	0	8	147149	0.09	1:30:09	8	41	117655	0.09	1:25:07	14200
geant	32	22	46	38	110	0	2	7233	0.11	0:02:45	2	34	5017	0.11	0:02:19	14800
sun	37	27	112	270	135	3	19	77364	0.12	2:40:24	15	34	73674	0.12	2:26:39	16573
atlanta	25	15	32	42	75	1	0	1149	0.05	0:00:20	1	17	1115	0.05	0:00:21	11102
nobelu	38	28	51	44	140	3	14	2713	0.09	0:01:16	12	31	1954	0.09	0:01:04	13250
janos-us-ca	49	39	132	390	195	2	3	2068	0.06	0:06:33	3	29	2274	0.06	0:07:24	12720
newyork	26	16	59	74	80	2	0	368	0.08	0:00:12	11	25	268	0.08	0:00:08	10100
dfnwin	21	11	57	57	55	1	11	145	0.10	0:00:03	4	0	91	0.10	0:00:02	9960
germany50	60	50	98	67	250	3	6	5759	0.10	0:04:36	11	84	3647	0.10	0:03:16	12460
norway	37	27	61	117	135	2	3	7848	0.13	0:05:02	3	33	2530	0.13	0:01:45	14700
diuan	21	11	52	45	55	1	7	4768	0.08	0:01:52	7	26	1011	0.08	0:00:26	11150
giul39	49	39	182	390	195	1	14	720	0.05	0:02:32	4	40	513	0.05	0:01:56	12810

Table 6.11: Efficiency of optimality condition inequalities for realistic instances

valid inequalities we could solve 16 instances out of 18 to optimality within a CPU time less than 3 minutes. However, with optimality condition inequalities, we could only solve 10 instances out of 18 within the same time limit. In addition, considering valid inequalities in the Branch-and-Cut algorithm significantly reduces the resolution time of the critical instances. For example, the instance sun is solved within 2:26:39 if the optimality condition inequalities are added, but when the valid inequalities are considered, sun is solved within 00:7:42. Moreover, we can see that for all the instances, by considering valid inequalities we obtain a better Gap and a better number of nodes in the Branch-and-Cut tree than those given by using optimality condition inequalities.

Furthermore, considering the valid inequalities and the optimality condition inequalities is clearly better than considering each of them separately as it is shown in Table 6.13.

							basic formulation				Branch-and-Cut with valid inequalities						
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*Sec*	*PCI*	*CmPI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	4	11	193	0.08	0:00:02	8	27	2	26	0.07	0:00:01	10360
janos-us	36	26	94	260	130	8	0	28559	0.09	0:52:35	8	220	0	13509	0.07	0:23:17	13201
nobel-germany	27	17	36	49	85	5	2	1405	0.11	0:00:26	3	79	3	53	0.07	0:00:07	14060
dfnwin	20	10	55	61	50	1	1	272	0.07	0:00:07	0	99	0	61	0.06	0:00:15	10400
pioro40	50	40	99	54	200	0	11	16646	0.11	0:16:01	8	120	0	395	0.07	0:01:06	14050
india35	45	35	90	50	175	0	4	420	0.11	0:00:18	1	172	0	27	0.07	0:00:07	10460
cost266	47	37	67	51	185	0	8	147149	0.09	1:30:09	7	129	0	266	0.07	0:00:31	14200
geant	32	22	46	38	110	0	2	7233	0.11	0:02:45	2	79	2	146	0.07	0:00:14	14800
sun	37	27	112	270	135	3	19	77364	0.12	2:40:24	7	215	6	3570	0.09	0:07:42	16573
atlanta	25	15	32	42	75	1	0	1149	0.05	0:00:20	1	35	0	67	0.04	0:00:05	11102
nobelu	38	28	51	44	140	3	14	2713	0.09	0:01:16	13	63	4	55	0.04	0:00:06	13250
janos-us-ca	49	39	132	390	195	2	3	2068	0.06	0:06:33	2	239	0	494	0.04	0:01:32	12720
newyork	26	16	59	74	80	2	0	368	0.08	0:00:12	0	144	0	17	0.07	0:00:06	10100
dfnwin	21	11	57	57	55	1	11	145	0.10	0:00:03	10	64	1	11	0.07	0:00:03	9960
germany50	60	50	98	67	250	3	6	5759	0.10	0:04:36	4	207	2	125	0.09	0:00:25	12460
norway	37	27	61	117	135	2	3	7848	0.13	0:05:02	6	139	1	166	0.06	0:02:55	14700
diuan	21	11	52	45	55	1	7	4768	0.08	0:01:52	8	60	1	71	0.06	0:00:12	11150
giul39	49	39	182	390	195	1	14	720	0.05	0:02:32	11	453	0	647	0.04	0:02:25	12810

Table 6.12: Efficiency of valid inequalities for realistic instances for realistic instances

							basic formulation				Branch and Cut							
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*Sec*	*N*	*Gap*	*CPU*	*Sec*	*OI*	*PCI*	*PCmI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	4	11	193	0.08	0:00:02	7	20	24	2	20	0.06	0:00:01	10360
janos-us	36	26	94	260	130	0	2	28559	0.09	0:52:35	0	18	220	0	7953	0.07	0:13:29	13201
nobel-germany	27	17	36	49	85	5	2	1405	0.11	0:00:26	2	9	79	3	52	0.07	0:00:07	14060
dfnwin	20	10	55	61	50	1	1	272	0.07	0:00:07	1	16	95	0	47	0.06	0:00:12	10400
pioro40	50	40	99	54	200	0	11	16646	0.11	0:16:01	7	76	88	0	148	0.05	0:00:25	14050
india35	45	35	90	50	175	0	4	420	0.11	0:00:18	4	39	165	0	39	0.07	0:00:10	10460
cost266	47	37	67	51	185	0	8	147149	0.09	1:30:09	6	41	115	0	261	0.07	0:00:30	14200
geant	32	22	46	38	110	0	2	7233	0.11	0:02:45	2	34	76	2	148	0.07	0:00:15	14800
sun	37	27	112	270	135	3	19	77364	0.12	2:40:24	7	34	233	5	2837	0.08	0:05:28	16573
atlanta	25	15	32	42	75	1	0	1149	0.05	0:00:20	0	17	33	0	48	0.04	0:00:04	11102
nobelu	38	28	51	44	140	3	14	2713	0.09	0:01:16	7	31	62	4	47	0.04	0:00:06	13250
janos-us-ca	49	39	132	390	195	2	3	2068	0.06	0:06:33	3	29	239	0	502	0.04	0:01:30	12720
newyork	26	16	59	74	80	2	0	368	0.08	0:00:12	2	25	148	0	17	0.06	0:00:07	10100
dfnwin	21	11	57	57	55	1	11	145	0.10	0:00:03	3	11	59	1	11	0.07	0:00:02	9960
germany50	60	50	98	67	250	3	6	5759	0.10	0:04:36	3	84	210	1	69	0.09	0:00:23	12460
norway	37	27	61	117	135	2	3	7848	0.13	0:05:02	3	33	117	0	74	0.02	0:01:18	14700
diuan	21	11	52	45	55	1	7	4768	0.08	0:01:52	4	26	50	1	70	0.06	0:00:12	11150
giul39	49	39	182	390	195	1	14	720	0.05	0:02:32	8	40	453	0	296	0.04	0:01:05	12810

Table 6.13: Efficiency of the branch and cut algorithm for realistic instances

We can see in Table 6.14 that 14 out of 18 instances are easily solved to optimality using the compact formulation. The four instances janos-us, sun, janos-us-ca and giul39 are solved within 0:18:15, 0:18:18, 0:24:13 and 0:16:10 respectively. Using the Branch-and-Cut algorithm, the CPU time of these instances is reduced to 0:13:29, 0:05:28, 0:01:30 and 0:01:05 which shows the efficiency of our Branch-and-Cut algorithm. Note that for these four critical instances, the number of attacks $\mid \Gamma \mid$ is large compared to the rest of the instances. This shows that for large $\mid \Gamma \mid$ the problem becomes more difficult to solve. Indeed, the larger is $\mid \Gamma \mid$, the larger is the number of variables of the compact formulation and the harder is the separation problems related to the path formulation.

							compact formulation			Branch and Cut							
Name	$\|V\|$	$\|T\|$	$\|A\|$	$\|\Gamma\|$	$\|K\|$	$\|K^*\|$	*N*	*Gap*	*CPU*	*Sec*	*OI*	*PCI*	*PCmI*	*N*	*Gap*	*CPU*	*Opt*
polska	22	12	28	30	60	0	0	0	0:00:01	7	20	24	2	20	0.06	0:00:01	10360
janos-us	36	26	94	260	130	0	134	0.09	0:18:15	0	18	220	0	7953	0.07	0:13:29	13201
nobel-germany	27	17	36	49	85	0	1	0.03	0:00:01	2	9	79	3	52	0.07	0:00:14	14060
dfnwin	20	10	55	61	50	0	1	0.03	0:00:01	1	16	95	0	47	0.06	0:00:12	10400
pioro40	50	40	99	54	200	0	0	0	0:00:01	7	76	88	0	148	0.05	0:00:25	14050
india35	45	35	90	50	175	0	0	0	0:00:01	4	39	165	0	39	0.07	0:00:10	10460
cost266	47	37	67	51	185	0	0	0	0:00:01	6	41	115	0	261	0.07	0:00:30	14200
geant	32	22	46	38	110	0	0	0	0:00:01	2	34	76	2	148	0.07	0:00:15	1480
sun	37	27	112	270	135	0	40	0.12	0:18:18	7	34	233	5	2837	0.08	0:05:28	16573
atlanta	25	15	32	42	75	0	0	0	0:00:01	0	17	33	0	48	0.04	0:00:04	11102
nobelu	38	28	51	44	140	0	0	0	0:00:01	7	31	62	4	47	0.04	0:00:06	13250
janos-us-ca	49	39	132	390	195	0	166	0.06	0:24:13	3	29	239	0	502	0.04	0:01:30	12720
newyork	26	16	59	74	80	0	0	0	0:00:01	2	25	148	0	17	0.06	0:00:07	10100
dfnwin	21	11	57	57	55	0	0	0	0:00:01	3	11	59	1	11	0.07	0:00:02	9960
germany50	60	50	98	67	250	0	0	0	0:00:01	3	84	210	1	69	0.09	0:00:23	12460
norway	37	27	61	117	135	0	0	0	0:00:02	3	33	117	0	74	0.02	0:00:18	14700
diuan	21	11	52	45	55	0	0	0	0:00:01	4	26	50	1	70	0.06	0:00:12	11150
giul39	49	39	182	390	195	0	12	0.05	0:16:10	8	40	453	0	296	0.04	0:01:05	12810

Table 6.14: Comparison with the compact formulation for realistic instances

6.3 Concluding remarks

In this chapter, we have devised a Branch-and-Cut algorithm to solve the path formulation PCSP2 introduced in Chapter 4. We have first presented the different steps of the algorithm that starts with a preprocessing phase including essential countermeasures and optimality condition inequalities. We

have then discussed separation algorithms for the security inequalities and valid inequalities. In particular, we have proposed exact separation algorithms for both the security and the countermeasures path inequalities. We have also proved that the separation problems of path covering inequalities and essential-by subsets removing-countermeasure are NP-Complete. That's why we have chosen to separate them with a heuristic procedure. In addition, we have provided a primal heuristic in order to reduce the number of nodes in the Branch-and-Cut tree.

In this chapter, we have also conducted extensive experimentations on random and realistic instances of the PCSP problem. The computational study has shown the efficiency of the polyhedral investigation from an algorithmic point of view. In fact, using both optimality condition inequalities and valid inequalities in the Branch-and-Cut algorithm have significantly improved the resolution of the problem. Moreover, we have studied the sensitivity of the algorithm to the density of the graph and the number of nodes. The numerical results have shown that the difficulty of solving the problem depends not only on the size of the instance but also on the number of attacks. The experiments have also shown that the Branch-and-Cut algorithm performs better for realistic instances than the random ones.

In the next chapter, we illustrate all the risk management models and algorithms proposed in this thesis for real cases.

Chapter 7

Application in telecommunication industry

Contents

7.1 Internet of Things (IoT) 135

7.1.1 System description and risk assessment 135

7.1.2 IoT risk treatment 142

7.2 Software Defined Network (SDN) 145

7.3 Integration in a web application 146

7.4 Concluding remarks 151

In this chapter we present some applications of the complete risk management framework in a telecommunication context. We apply our approach in Internet of Things (IoT) and Software Defined Network (SDN) use cases. We also show the integration of our framework in a web application that we have developed and illustrated in a Local Network Areas (LANs) use case. We apply to each case study the risk assessment methods developed in Chapter 3 and the risk treatment optimization algorithms proposed in Chapters 4, 5 and 6.

7.1 Internet of Things (IoT)

7.1.1 System description and risk assessment

The IoT system we consider is described in Figure 7.1. The case study, corresponds to the integration of several Philips Hue [20] connected light bulbs

in a simple Information System (IS). The bulbs are connected to an administration platform controllable with smartphones and sometimes through the Internet. The bulbs are used into a network with a few connected machines (workstations, smartphones, etc.), as shown in Figure 7.1.

These machines can be used to interact with the bulbs (*i.e.* switch the lights on or off). The bulbs are also connected to an automation service hosted outside of the host IS, on the Internet. The machines can also communicate with the automation service. Most of the communication is directed to the bulbs and not the other way, except for the internally hosted server (used to keep activity logs) and the Internet hosted automation service. As the system represents a company, it also contains a few employees. The detailed list of assets of the system are described in Table 7.1. The access points of this system are A4.4, A4.5, A3.4, A3.6, A5.5 and A5.8.

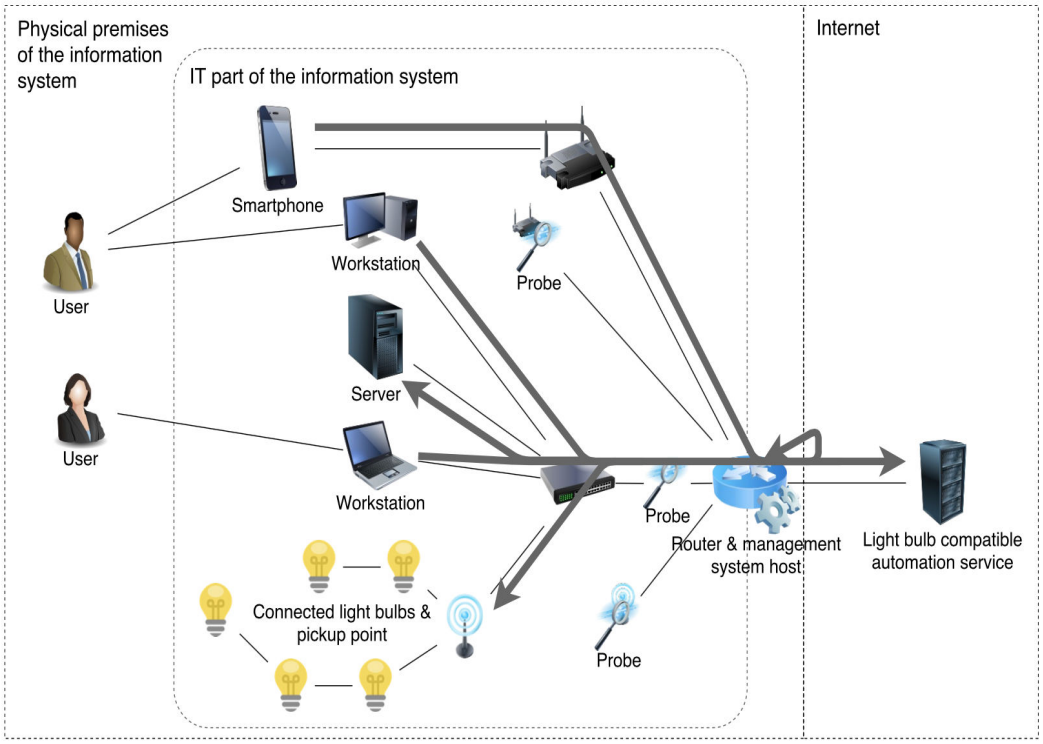


Figure 7.1: The IoT case Study

Our security risk management algorithms can be placed on the router serving as a bridge between the host IS and the Internet. Note that Several network probes report back to it: One for each type of network deployed within the IS (Ethernet, WiFi and ZigBee [21]). Probes supporting other communication technologies could have been added, in order to detect objects communicating with different technologies (Bluetooth for example) but the number of probes is kept to a minimum.

Table 7.1: List of assets for the IoT infrastructure & the host IS

	Asset	Type	Identifier
Local environment	Orders from the top of the infrastructure	primary	A1.1
	Raw data from the light bulbs	primary	A1.2
	Security configuration	primary	A1.3
	Light bulbs	secondary	A1.4
	Local pickup point	secondary	A1.5
Transport-ation	Orders from the top of the infrastructure	primary	A2.1
	Raw data from the light bulbs	primary	A2.2
	Security configuration	primary	A2.3
	Telecommunication hardware	secondary	A2.4
Storage & data mining	Orders from the top of the infrastructure	primary	A3.1
	Raw data from the light bulbs	primary	A3.2
	Aggregated data from the light bulbs	primary	A3.3
	Security configuration	primary	A3.4
	Aggregation algorithms	secondary	A3.5
	Servers	secondary	A3.6
Provision	Aggregated data from the light bulbs	primary	A4.1
	Orders from the provision level	primary	A4.2
	Security / access management configuration	primary	A4.3
	APIs / GUIs	secondary	A4.4
	Servers	secondary	A4.5
Host information system	Business data	primary	A5.1
	Intellectual property	primary	A5.2
	Customer information	primary	A5.3
	Employee information	primary	A5.4
	Company hardware (servers, PCs, phones, etc.)	secondary	A5.5
	Offices	secondary	A5.6
	Offices (related data)	secondary	A5.7
	Processes	secondary	A5.8
	Employees	secondary	A5.9

Then using all available knowledge of the IoT infrastructure, the vulnerability table can be created. The vulnerability basis is built on the list given in [85]. The threat scenarios list is shown in Tab. 7.3. The assets associated to each vulnerability are described in Table 7.4. The objective this risk analysis process is to determine the probability of occurrence of a vulnerability and its impact. The impact can be refined into six different security properties (confidentiality, integrity, availability, accountability, usability and auditability). The impact that will be used to evaluate the risks is the sum of these elementary impacts. The assets impacted in the case of an occurrence of the vulnerability are also listed as shown in Table 7.4. The impact and likelihood values used are bounded to the scales defined in Table 7.2. The potentiality of each threat can then be calculated using equation (3.1) , where $p_i = \frac{likelihood}{10}$.

Table 7.2: Scales for impact & likelihood values

	Impact	Likelihood
5.	Critical / very high	Frequent
4.	High	Likely
3.	Medium	Possible
2.	Low	Unlikely
1.	Informational / very low	Very unlikely

Table 7.3: List of vulnerabilities for the IoT infrastructure

Identifier	Vulnerability
T1	An attacker with physical access to the device tampers with the firmware in order to use it to leak sensitive information
T2	An attacker steals one or several light bulbs
T3	Part or all the telecomminication equipement used by the bulbs to communicate with the other levels of the infrastructure
T4	An attacker replaces (or adds) non legit data (or orders) to the normal communication flux
T5	An attacker replaces (or adds) non legit data (or orders) to what is stored
T6	An attacker manages to overload the system (coming from either the local environment, the transportation and/or the provision)
T7	An attacker manages to give the system non legitimate information
T8	An attacker goes arround the right management system in order to achieve action that they should not be able to achieve

Table 7.4: Results of risk analysis for integration of the IoT infrastructure in the host IS

	Vulnerabilities	Likelihood	Information system impacts											
			Confidentiality		Integrity		Availability		Accountability		Usability		Auditability	
Local environment	T1	1	4	A5.1→5.4	4	A5.1→5.4	4	A5.1→5.4	4	A5.1→5.4				
	T2	3					1	A5.6			2	A5.9		
Transportation	T3	2					1	A5.6			3	A5.9		
	T4	2			2	A5.7	1	A5.6	2	A5.7				
Storage & data mining	T5	2			3	A5.7	1	A5.6	3	A5.7				
	T6	3			3	A5.7	1	A5.6			3	A5.9		
Provision	T7	4			3	A5.7	1	A5.6	3	A5.7				
	T8	3	4	A5.1→5.4										

We study the system in a discrete time horizon $I = \{1, 2, 3, 4\}$. In Figure 7.2, we present the RAG for $i = 4$. We choose a system of colors in function of the values of the potentiality (3.1) and propagation function (3.4): green for small values and red for large ones. We can see that the main color for the RAG of Figure 7.2 is red. That is to say that the system needs to be secured which is the purpose of the next section.

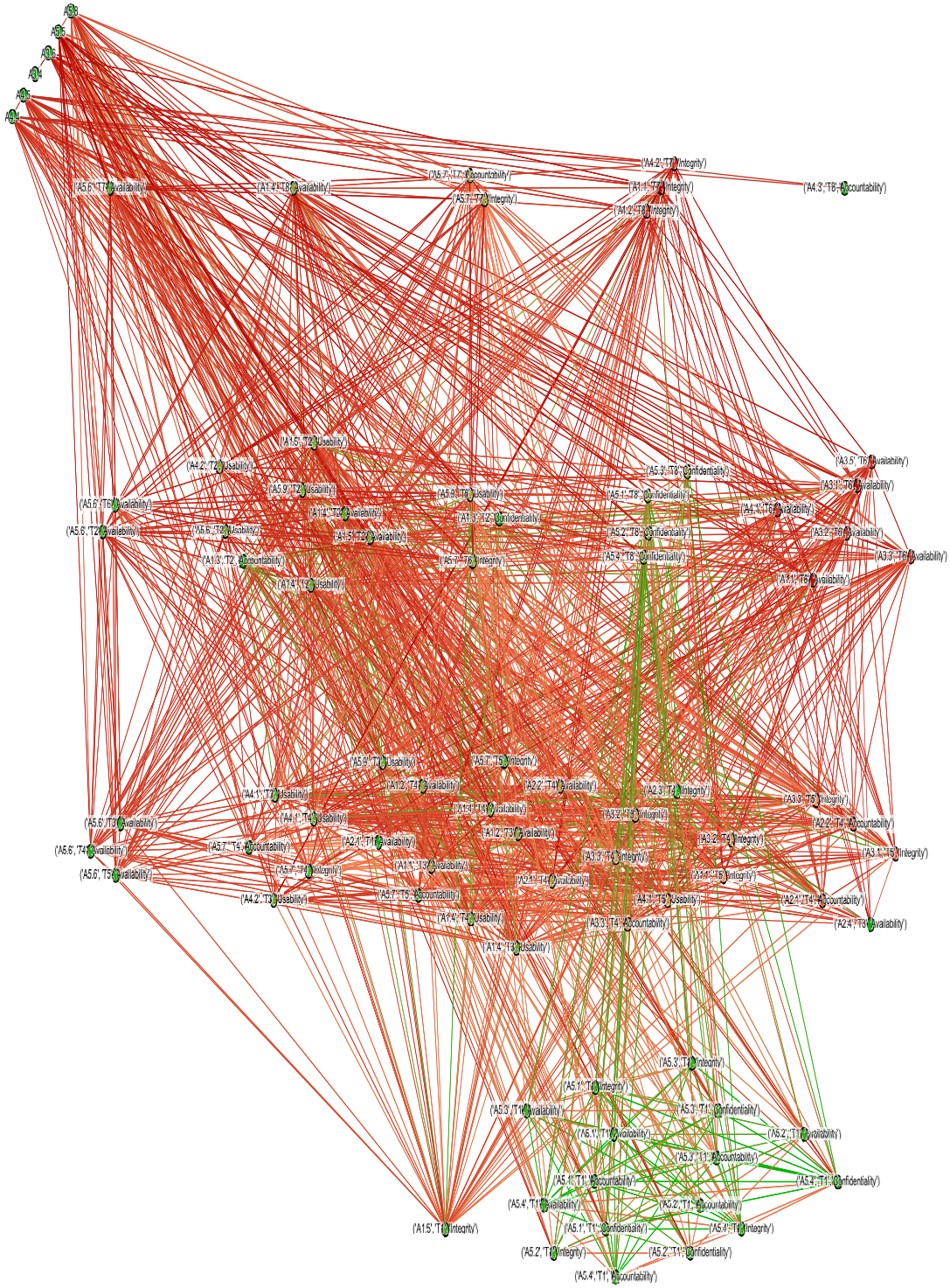


Figure 7.2: The IoT RAG at $i = 1$

7.1.2 IoT risk treatment

The results of global risk evaluation algorithm are shown in Figure 7.5. The red line represents the global risk threshold which is equal to 61. The thresholds are chosen to be the same for each access point and each asset-vulnerability node, and can be deduced from the global risk threshold using equation (3.13). The available countermeasures with their associated cost and effect for the IoT system are described in Table 7.5.

We solve the PCSP, for each time slot, to obtain the optimal placement of countermeasures. The solution of the PCSP is found within 7 seconds with a gap equal to 0.06. The number of nodes in the Branch-and-Cut tree is equal to 9. The number of generated optimality condition inequalities and path covering inequalities are respectively 17 and 15. In Figure 7.4, we represent the RAG at $i = 4$ after countermeasures placement. We can see that the main color of the graph is now green, which means that the system is secure. This is represented in Figure 7.5, where we can see the risk after countermeasures placement is less than the threshold (red line).

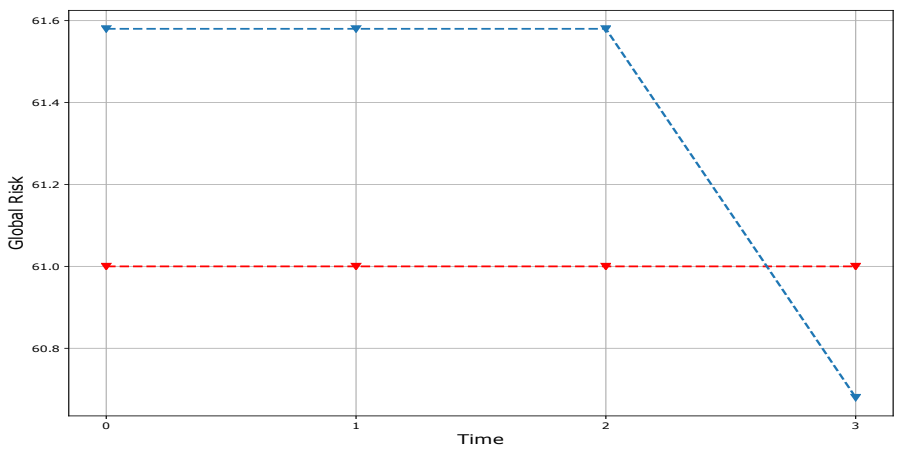


Figure 7.3: Risk evaluation before countermeasures placement of the IoT system

Table 7.5: List of proposed countermeasures for the vulnerabilities of the IoT system

Vulnerability	Countermeasure	Description	Associated asset	Cost	Effect
T1	T1K1	Sensitization about access to the premises (badges, notification to security...)	A5.9	3	3
	T1K2	Installation of a better access control system	A5.6	6	6
	T1K3	Installation of light variation detection sensors	A5.6	9	6
T2	T2K1	Sensitization about access to the premises (badges, notification to security...)	A5.9	3	5
	T2K2	Installation of a better access control system	A5.6	6	7.5
	T2K3	Installation of locks on the bulbs to make their removal harder	A5.6	2	2
T3	T3K1	Installation of a backup communication mechanism	A2.4	10	1.75
	T3K2	Installation of a backup set of non connected light bulbs	A5.6	7	1.5
T4	T4K1	Enhancement of the encryption in the transportation level	A2.3	5	1
	T4K1bis	Enhancement of the encryption in the transportation level	A1.3, A3.4	5	0.4
	T4K2	Installation of external sensors to validate data from the bulbs (on/off), see T1C3	A5.6	9	7.5
T5	T5K1	Use of a correlation engine to detect incorrect data	A3.5	7	.3
	T5K2	Installation of a better access control system (IT)	A3.6	6	1
	T5K3	Installation of a better access control system (physical)	A3.6	9	1
	T5K4	Installation of a backup set of non connected light bulbs	A5.6	7	1.5
T6	T6K1	Installation of a load balancing system	A3.6	6	7
	T6K2	Installation of a backup set of non connected light bulbs	A5.6	7	4
T7	T7K1	Enhancement of the user input validation mechanisms	A4.4	3	6
	T7K2	Installation of a web application firewall between the API/GUI and the user	A5.5 (server)	5	5
T8	T8K1	Enhancement of the user input validation mechanisms	A4.4	3	9
	T8K2	Installation of a web application firewall between the API/GUI and the user	A5.5 (server)	5	7
	T8K3	Use of better right management policies	A5.8	4	6

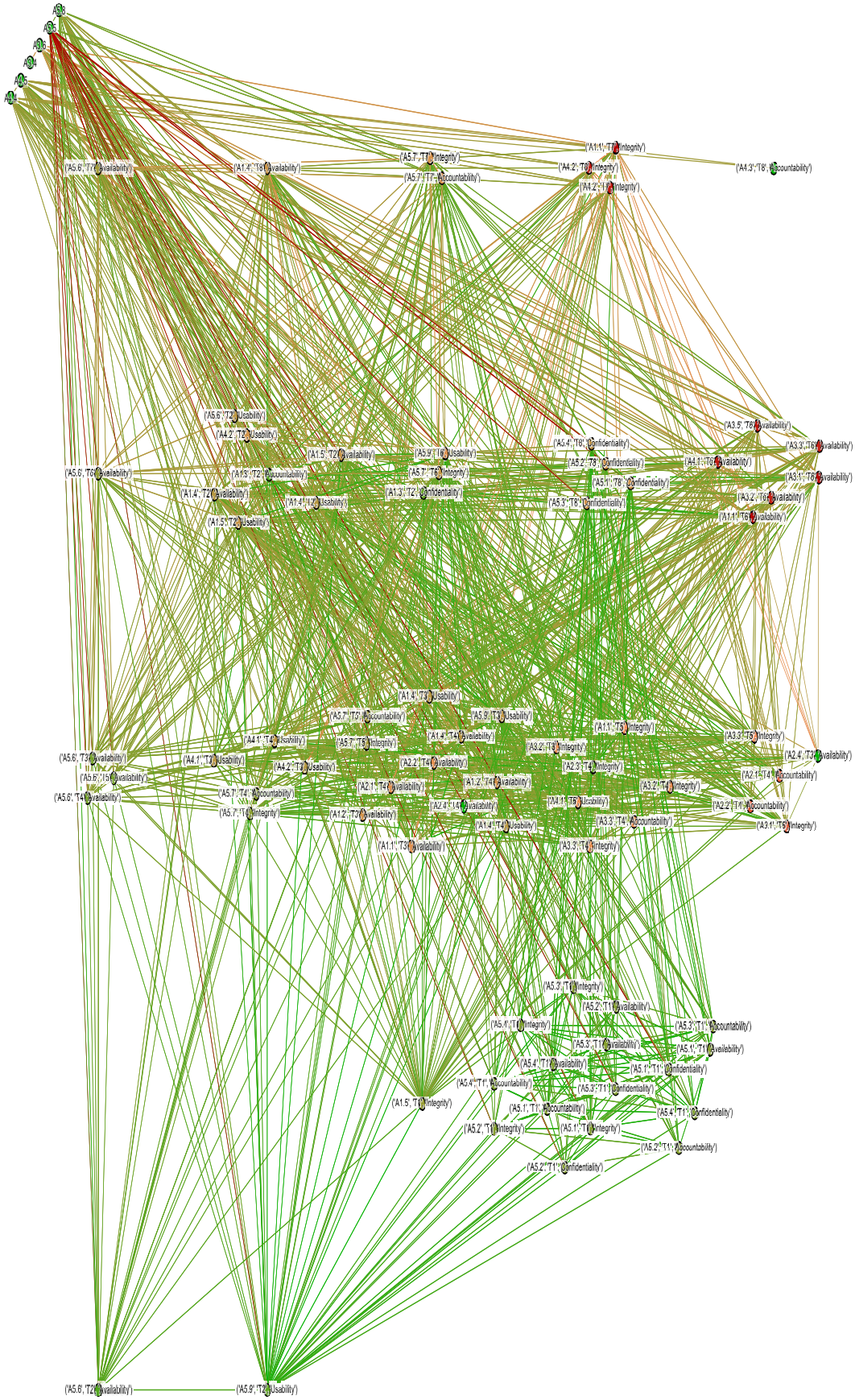


Figure 7.4: RAG after countermeasure placement at $i = 4$ of the IoT system

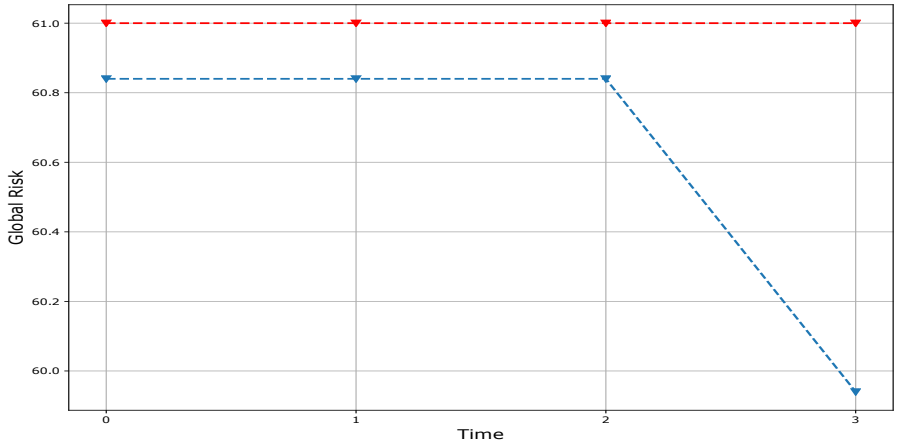


Figure 7.5: Risk evaluation after countermeasure placement of the IoT system

7.2 Software Defined Network (SDN)

In Chapter 3, we have illustrated our risk assessment approach in a SDN case study for which we have generated the RAGs and have evaluated the system risks using Algorithm 2. In what follows, we apply the risk treatment approach that we developed to the same case study in the time horizon $I = \{1, 2, 3, 4\}$. The global system risk evaluation over the time is recalled in Figure 7.6. As we can see in this figure, the global risk threshold is set to 27. Three countermeasures are assigned to each asset-vulnerability node.

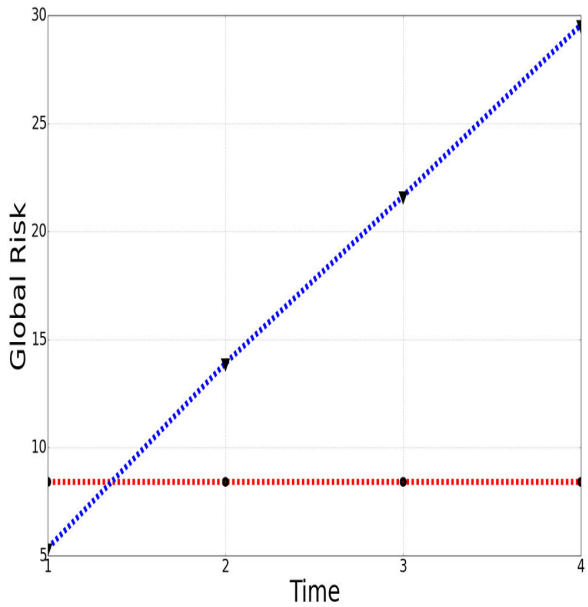


Figure 7.6: SDN system risk before countermeasures placement

For each time slot, we solve the PCSP to obtain the optimal placement of countermeasures. The Branch-and-Cut algorithm for this case study run in a time less than 1s and a gap equal to 0.06. The number of nodes in the Branch-and-Cut tree is equal to 16. The number of generated optimality condition inequalities and path covering inequalities are respectively 20 and 12. The global system risk after countermeasures placement is given in Figure 7.7.

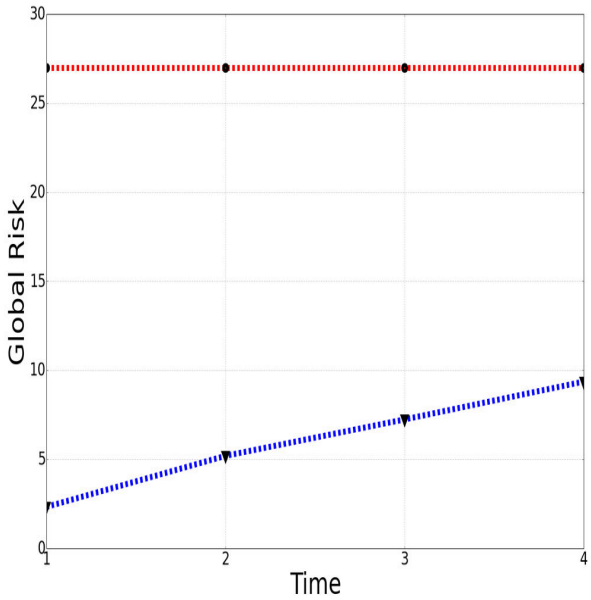


Figure 7.7: SDN system risk after countermeasures placement

7.3 Integration in a web application

A web application has been developed to provide a user friendly tool to security experts and optimizers. The web application covers the complete risk management framework we have proposed in this thesis including risk analysis (the RAGs), risk assessment and risk treatment, along with a visualization interface.

The application architectureis based on: *a frontend* that contains the majority of the application's logic, and *a backend* that behaves like an API, called by the front-end when needed. The frontend handles the display, as well as the management of user events. The technological tools used are Javascript/NodeJS [13] with React [23] and Redux [24] libraries. The backend includes all the risk assessment and risk treatment algorithms developed along this thesis using Python.

In Figure 7.8, we can see a screenshot of the application. We can see a large

space for displaying the RAGs as well as the three steps: risk analysis, risk assessment and risk management.

The use case that is presented is a LANs system in a time horizon $I = \{1, 2, 3, 4\}$. The challenge of this the risk analysis part is to visualize the evolution of RAGs over time. For that, a slider allows to begin the visualization, to go back or to pass to the next time slot. The user can click on the risk assessment button to visualize the RAGs. In Figure 7.9, the application displays the RAG at $i = 1$ and in Figure 7.10 it shows the RAG at $i = 4$.

The risk evaluation button allows to launch the corresponding algorithms. The results are displayed in Figure 7.14. The application also offers a set of available countermeasures for the system that is displayed as shown in Figure 7.12.

The risk treatment button permits to run the the Branch-and-Cut algorithm. In Figure 7.13, we can see the optimal placement of countermeasures for this LANs case study and the solution cost. Once the problem is solved, the user can also display the risk evaluation after countermeasures placement as shown in Figure 7.14.

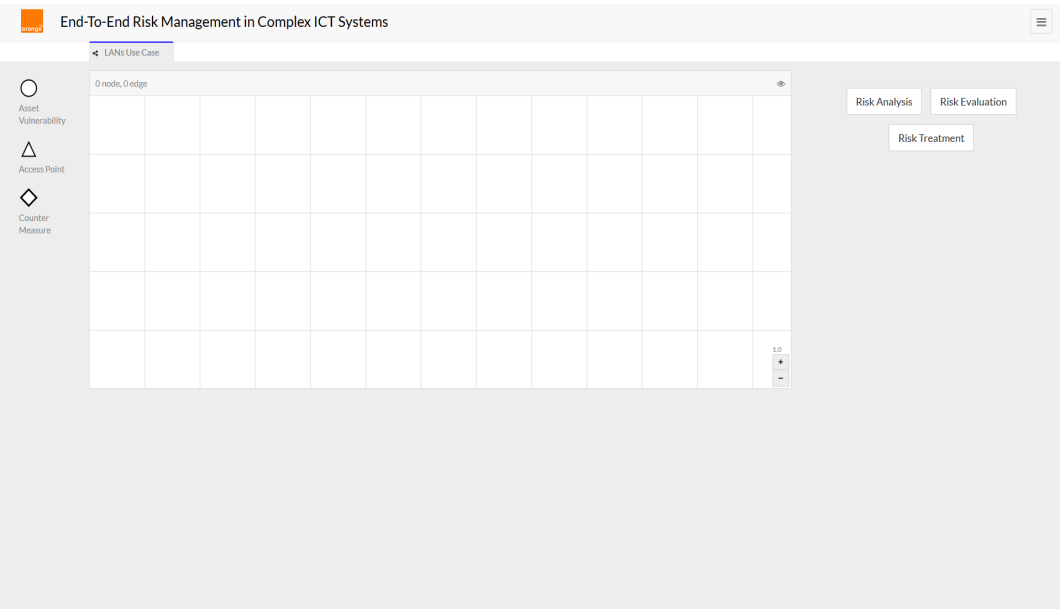


Figure 7.8: Overview of the web application

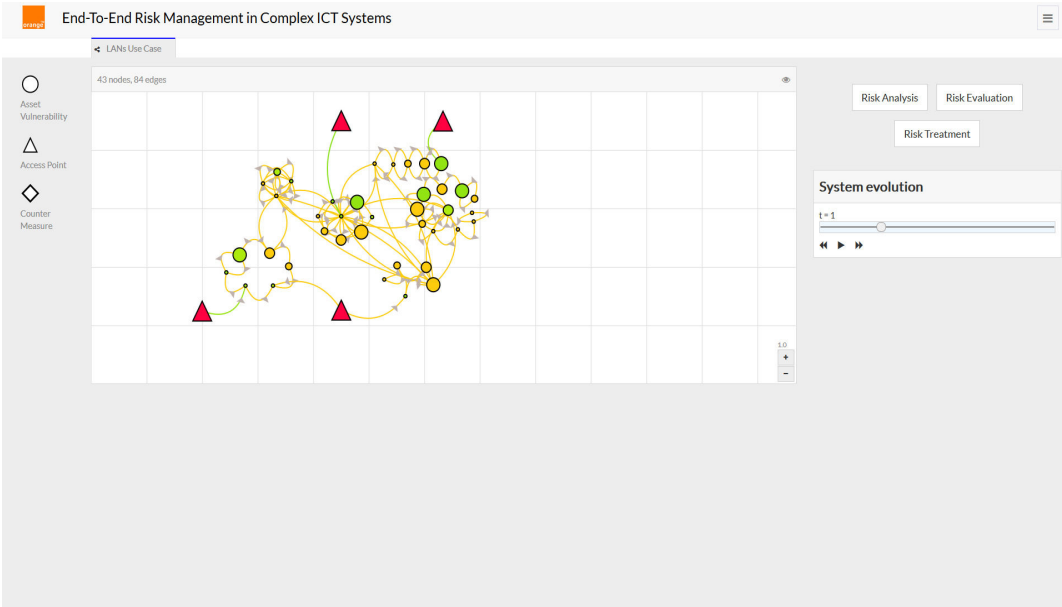


Figure 7.9: screenshot of the RAG at $i = 1$ in the web application

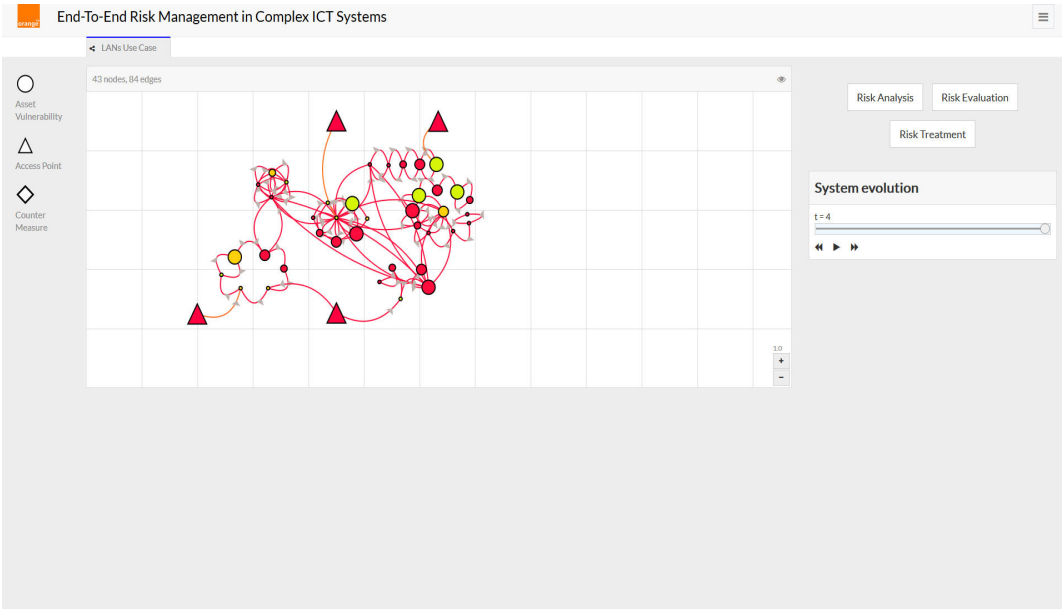


Figure 7.10: screenshot of the RAG at $i = 4$ in the web application

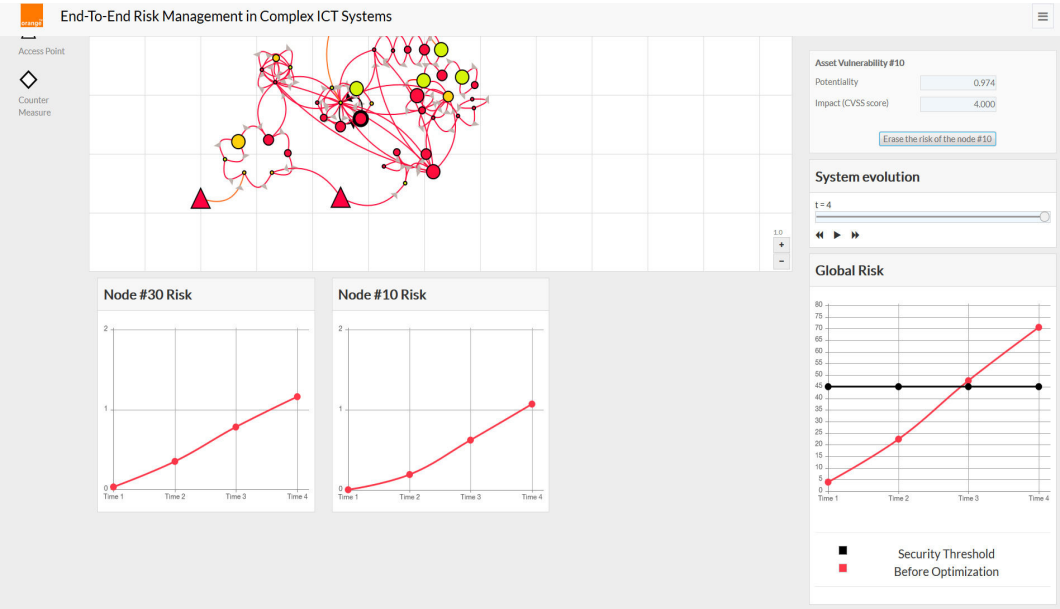


Figure 7.11: Risk evaluation in the web application

Figure 7.12: Available countermeasures in the web application

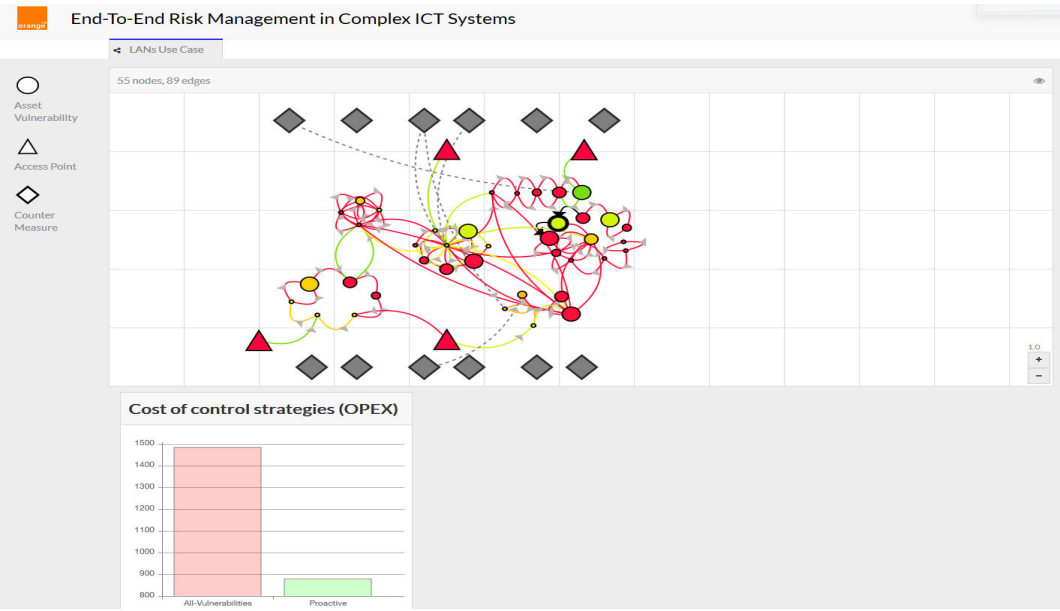


Figure 7.13: Deployment and cost of countermeasures placement in the web application

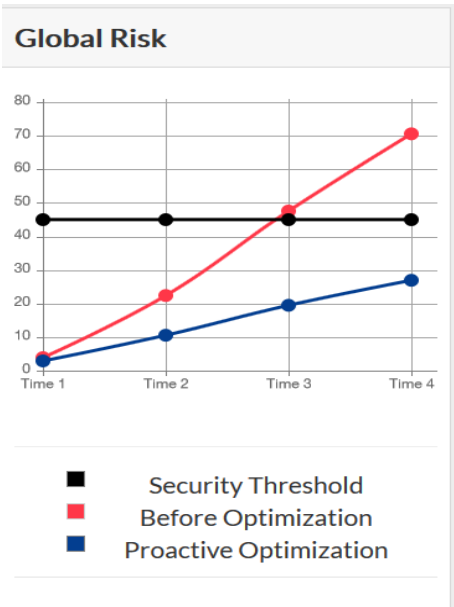


Figure 7.14: Risk evaluation after countermeasures placement in the web application

7.4 Concluding remarks

In this chapter, we have presented some realistic applications arising in the telecommunication domain, including Internet of Things, Software Defined Network (SDN) and Local Network Areas (LANs) use cases. We have also shown the integration of our framework in a web application that has been developed to give an intelligent management and visualization of the framework we have proposed in this thesis.

Conclusion and perspectives

In this thesis, we have proposed a new risk assessment framework based on the Risk Assessment Graphs (RAGs) model. The RAGs take into account at the same time the vulnerabilities, the system topology and the accessibility. They consider not only the current system state, but also the way it evolves throughout a time horizon. In addition, all possible attackers and attack scenarios are explicitly considered as paths in the RAGs.

We have also provided a risk evaluation approach based on these graphs. We have defined new security metrics namely the propagation difficulty, the propagated risk, the node risk, and the global risk. Our risk assessment approach has been illustrated in a SDN case study. Several simulations on random systems were conducted to show the sensitivity of our metrics to the size of the network, the vulnerability, the topology and the accessibility between the network assets.

Then, we have introduced a risk treatment problem, called the Proactive Countermeasure Selection Problem (PCSP). This consists in finding the placement of countermeasures on the assets in such a way that that the system is secured at minimal cost. We have shown that the PCSP is NP-complete. We have also considered a bilevel model for the PCSP for which we have proposed two single-level reformulations. The first one is a compact formulation and based on primal-dual optimality conditions. The second one is a path formulation, with an exponential number of security inequalities, obtained from the compact formulation by projection. In addition, we have introduced optimality condition inequalities that have been used during the preprocessing phase to improve the algorithmic aspect.

Moreover, we have studied the PCSP path formulation from a polyhedral point of view. We have discussed the facial aspect of the basic inequalities. We have also introduced three families of valid inequalities: the path covering inequalities, the countermeasures path inequalities and the essential- by subsets removing- countermeasure inequalities. Necessary conditions and sufficient conditions for these inequalities to be facet defining have been discussed.

Furthermore, we have devised a Branch-and-Cut algorithm to solve the

path formulation. The essential countermeasure equations and the optimality condition inequalities have been added in the preprocessing phase. We have also studied the separation problems associated to security inequalities and each class of valid inequalities. This has led us to propose exact separation algorithms for the security inequalities and countermeasures path inequalities. We have also shown that the separation problems associated to path covering inequalities and essential- by subsets removing- countermeasure inequalities are NP-Complete. For this reason, we have used heuristic procedures for seprating these inequalities. Furthermore, we have proposed a primal heuristic in order to reduce the number of nodes in the Branch-and-Cut tree.

We have then conducted extensive experimentations on random and realistic instances of the PCSP problem. The computational study has shown the efficiency of the polyhedral study from an algorithmic point of view. In fact, considering optimality condition inequalities and valid inequalities within the Branch-and-Cut algorithm has significantly improved the resolution of the problem. In addition, we have studied the sensitivity of the algorithmic performances to some problem parameters like the density of the graph, the number of nodes and the number of attacks.

Finally, we have presented some real applications in the telecommunication industry to our framework. In particular, we have applied our approach in an Internet of Things, a Software Defined Network (SDN) and a Local Network Areas (LANs) use cases. Moreover, we have developed a web application in order to give an intelligent management and visualization of the complete framework which have been proposed in this thesis.

The perspectives of this thesis are mostly related to our optimization-based risk treatment approach. This has shown its efficiency in solving random and realistic instances and has also been used in several applications in telecommunication industry. However, there are different directions in which our future research related to the PCSP can be conducted.

Actually, the two proposed formulations can be strengthened. In fact, We can enrich our polyhedral study with further optimality condition inequalities and further classes of valid inequalities. These results can improve the Branch-and-Cut algorithm which will potentially be able to solve the path formulation for a series of critical instances that we are not able to solve so far.

Moreover, the compact formulation and the path formulation have the placement variables in common. As these variables are the only variables considered in the polyhedral investigation, one can use the polyhedral results obtained in this thesis with the compact formulation. This will allow us to examine the efficiency of the polyhedral study for the compact formulation from an algorithmic point of view. In addition, more efficient separation heuristics and more sophisticated preprocessing methods can be developed in order to

improve the resolution of the problem.

Furthermore, we can investigate other variants of the PCSP. For instance, one can consider that the countermeasures are only associated to the arcs. We can also study the variant of the problem where the countermeasures can be associated to the arcs and to the nodes at the same time. From a security point of view, this allows us to consider several types of countermeasures and from a polyhedral point of view, this can lead us to very interesting formulations. Note finally that some problem parameters such as the difficulty of propagation can be uncertain. An interesting direction in that case would consist of using robust optimization [44] to investigate the uncertain variant of the PCSP.

List of Figures

1 Impact de la topologie et la rapidité de convergence de l'accessibilité sur le risque (p et β) XIII

2 Impact de la rapidité de convergence de la potentialité α sur le risque XIV

3 Vue d'ensemble de l'application XXII

4 Placement et coût des contre-mesures XXII

1.1 The risk management process 12

1.2 Simplified SDN architecture 16

2.1 An undirected graph G 22

2.2 Subgraph H_1 of G 23

2.3 Spanning subgraph H_2 of G 23

2.4 A directed graph G 24

2.5 Directed Subgraph H_3 of G 25

2.6 Covering directed subgraph H_4 of G 25

2.7 Relations between P, NP and NP-Complete 30

2.8 A convex hull 31

2.9 Valid inequality, facet and extreme points 33

2.10 A hyperplan separating x^* and P 34

2.11 A Branch-and-Cut tree 37

2.12 The linear bilevel programming problem 41

3.1 Framework Description 47

3.2 Simplified representation of a RAG 52

3.3 SDN Use Case 57

3.4 SDN Risk Assessment Graphs ($I = 4$) 58

3.5 Node Risk in Function of Time 59

3.6 Global Risk in Function of Time 60

3.7 Mean Global Risk in Function of the Number of Nodes 62

3.8 Impact of the topology and the accessibility convergence speed (p and β) 62

3.9 Impact of the potentiality convergence speed α 63

4.1 A first PCSP example 70

4.2 A second PCSP example 71

4.3 A third PCSP example 71

4.4 Dominance of countermeasures on a path 80

5.1 An instance with essential countermeasures 87

5.2 The graph G_K 87

5.3 Matrix of Equations (5.5) 88

5.4 Path covering inequalities 95

5.5 Reduction 102

6.1 The bipartite graph $\tilde{K} = (K' \cup K'', \tilde{A})$ used for separating the Countermeasures Path inequalities 114

7.1 The IoT case Study 136

7.2 The IoT RAG at $i = 1$ 141

7.3 Risk evaluation before countermeasures placement of the IoT system 142

7.4 RAG after countermeasure placement at $i = 4$ of the IoT system 144

7.5 Risk evaluation after countermeasure placement of the IoT system 145

7.6 SDN system risk before countermeasures placement 145

7.7 SDN system risk after countermeasures placement 146

7.8 Overview of the web application 147

7.9 screenshot of the RAG at $i = 1$ in the web application 148

7.10 screenshot of the RAG at $i = 4$ in the web application 148

7.11 Risk evaluation in the web application 149

7.12 Available countermeasures in the web application 149

7.13 Deployment and cost of countermeasures placement in the web application 150

7.14 Risk evaluation after countermeasures placement in the web application 150

List of Tables

1 L'ensemble de countre-measures XVI

2 Ensemble de contre-mesures pour les instances réalistes XVIII

3 Efficacité de l'algorithme Branch and Cut dans la résolution de la famille $F_{S,T}$ XIX

4 Comparaison de la formulation chemin (avec Branch and Cut) et la formulation compacte pour la famille $F_{S,T}$ XX

5 Efficacité de l'algorithme Branch and Cut pour les instances réalistes XX

6 Comparaison de la formulation chemin (avec Branch and Cut) et la formulation compacte pour les instances realistes XXI

1.1 Vulnerability categories 20

3.1 Topology and Vulnerability Data Basis Mapping 65

6.1 The set of countermeasures 121

6.2 Efficiency of optimality conditions in solving F_p 123

6.3 Efficiency of valid inequalities in solving f F_p 124

6.4 Efficiency of the Branch and Cut algorithm in solving F_p . . . 124

6.5 Comparison with the compact formulation for F_p 125

6.6 Efficiency of optimality conditions in solving $F_{S,T}$ 126

6.7 Efficiency of valid inequalities in solving $F_{S,T}$ 127

6.8 Efficiency of the Branch and Cut algorithm in solving $F_{S,T}$ . . 128

6.9 Comparison with the compact formulation for $F_{S,T}$ 128

6.10 Set of countermeasures for the realistic instances 130

6.11 Efficiency of optimality condition inequalities for realistic instances 131

6.12 Efficiency of valid inequalities for realistic instances for realistic instances 132

6.13 Efficiency of the branch and cut algorithm for realistic instances 132

6.14 Comparison with the compact formulation for realistic instances 133

7.1 List of assets for the IoT infrastructure & the host IS 137

7.2 Scales for impact & likelihood values 138

7.3 List of vulnerabilities for the IoT infrastructure 139

7.4 Results of risk analysis for integration of the IoT infrastructure in the host IS 139

7.5 List of proposed countermeasures for the vulnerabilities of the IoT system 143

Bibliography

[1] Accenture. https://www.accenture.com/fr-fr.

[2] Cplex. http://www-01,ibm,com/software/commerce/optimization/cplex-optimizer/.

[3] Cve. http://cve,mitre,org/.

[4] Cve. https://cpe,mitre,org/.

[5] Databreach. http://www.idtheftcenter.org/Table/Data-Breaches/.

[6] Ebios. http://www,ssi,gouv,fr.

[7] Enisa. https://www.enisa.europa.eu/.

[8] Etsi. https://www.etsi.org/.

[9] Gartner. http://www.gartner.com/newsroom/id/3404817.

[10] Idg. https://www.idg.com/.

[11] Ietf. https://www.ietf.org/.

[12] Iso. https://www.iso.org/.

[13] Javascript, nodejs. https://nodejs.org/fr/.

[14] Juniper. https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion.

[15] Matplotlib. https://matplotlib.org/.

[16] Networkx. https://networkx.github.io/.

[17] Nist. http://nvd,nist,gov/download,cfm.

[18] Pendas. https://pandas.pydata.org/.

[19] Penom. https://securityintelligence.com/media/2016-cost-data-breach-study/.

[20] Philips hue. http://www2.meethue.com/en-us/.

[21] Philips hue – developer program. http://www.developers.meethue.com/.

[22] Python. https://www.python.org/.

[23] React. https://facebook.github.io/react/.

[24] Redux. http://redux.js.org/.

[25] Sndlib. http://sndlib.zib.de/home.action.

[26] Stats. https://thebestvpn.com/cyber-security-statistics-2019/.

[27] Muhammad Abedin, Syeda Nessa, Ehab Al-Shaer, and Latifur Khan. Vulnerability analysis for evaluating quality of protection of security policies. In *Proceedings of the 2nd ACM workshop on Quality of protection*, pages 49–52. ACM, 2006.

[28] Mamta Agiwal, Abhishek Roy, and Navrati Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 18(3):1617–1655, 2016.

[29] Ijaz Ahmad, Suneth Namal, Mika Ylianttila, and Andrei Gurtov. Security in software defined networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(4):2317–2346, 2015.

[30] Mohammad Salim Ahmed, Ehab Al-Shaer, and Latifur Khan. A novel quantitative approach for measuring network security. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1957–1965. IEEE, 2008.

[31] Christopher J Alberts, Sandra G Behrens, Richard D Pethia, and William R Wilson. Operationally critical threat, asset, and vulnerability evaluation (octave) framework, version 1.0. 1999.

[32] Paul Ammann, Duminda Wijesekera, and Saket Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 217–224. ACM, 2002.

[33] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook. *The traveling salesman problem: a computational study*. Princeton University Press, 2007.

[34] Andrea S Atzeni and Antonio Lioy. Why to adopt a security metric? a brief survey. *Quality of Protection*, 23:1–12, 2006.

[35] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[36] Michael O Ball, Bruce L Golden, and Rakesh V Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2):73–76, 1989.

[37] Michael O Ball, Bruce L Golden, and Rakesh V Vohra. Finding the most vital arcs in a network. *Operations Research Letters*, 8(2):73–76, 1989.

[38] John S Baras and George Theodorakopoulos. Path problems in networks. *Synthesis Lectures on Communication Networks*, 3(1):1–77, 2010.

[39] Jonathan F Bard. An algorithm for solving the general bilevel programming problem. *Mathematics of Operations Research*, 8(2):260–272, 1983.

[40] Jonathan F Bard. Convex two-level optimization. *Mathematical programming*, 40(1-3):15–27, 1988.

[41] Halil Bayrak and Matthew D Bailey. Shortest path network interdiction with asymmetric information. *Networks: An International Journal*, 52(3):133–140, 2008.

[42] Richard Bellman. On a routing problem. *Quarterly of applied mathematics*, 16(1):87–90, 1958.

[43] W. Ben-ameur, A. R. Mahjoub, and J. Neto. Paradigms of combinatorial optimization. In V. Th. Paschos, editor, *The Maximum Cut Problem*, pages 131–164. ISTE-WILEY, 2010.

[44] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[45] Theophilus Benson, Aditya Akella, and David A Maltz. Unraveling the complexity of network management. In *NSDI*, pages 335–348, 2009.

[46] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Citeseer, 1976.

[47] Endre Boros, Konrad Borys, V Gurevich, and Gabor Rudolf. Inapproximability bounds for shortest-path network interdiction problems. Technical report, Technical report, Rutgers University, Piscataway, NJ, USA, 2006.

[48] Paul Brody and Veena Pureswaran. Device democracy: Saving the future of the internet of things. *IBM, September*, 2014.

[49] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[50] HW Corley and Y Sha David. Most vital links and nodes in weighted networks. *Operations Research Letters*, 1(4):157–160, 1982.

[51] Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

[52] Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

[53] Todd Deshane, Zachary Shepherd, Jeanna Matthews, Muli Ben-Yehuda, Amit Shah, and Balaji Rao. Quantitative comparison of xen and kvm. *Xen Summit, Boston, MA, USA*, pages 1–2, 2008.

[54] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[55] Alexandros G Dimakis, Kannan Ramchandran, Yunnan Wu, and Changho Suh. A survey on network codes for distributed storage. *Proceedings of the IEEE*, 99(3):476–489, 2011.

[56] J. Edmonds. Covers and packings in a family of sets. *Bulletin of the American Mathematical Society*, 68(5):494–499, 1962.

[57] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards (B)*, 69:125–130, 1965.

[58] Thomas Arthur Edmunds and Jonathan F Bard. Algorithms for nonlinear bilevel mathematical programs. *IEEE transactions on Systems, Man, and Cybernetics*, 21(1):83–89, 1991.

[59] P ERDdS and A R&wi. On random graphs i. *Publ. Math. Debrecen*, 6:290–297, 1959.

[60] P Erdős and A Rényi. On random matrices ii. *Studia Sci. Math. Hungar*, 3:459–464, 1968.

[61] P Erdős and Alfréd Rényi. On the existence of a factor of degree one of a connected random graph. *Acta Mathematica Hungarica*, 17(3-4):359–368, 1966.

[62] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[63] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

[64] Paul Erdős and Alfréd Rényi. On the strength of connectedness of a random graph. *Acta Mathematica Hungarica*, 12(1-2):261–267, 1961.

[65] Hong Fan and Haozhong Cheng. Transmission network expansion planning with security constraints based on bi-level linear programming. *European transactions on electrical power*, 19(3):388–399, 2009.

[66] Shahin Farahani. *ZigBee wireless networks and transceivers*. Newnes, 2011.

[67] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[68] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.

[69] Pimmy Gandotra and Rakesh Kumar Jha. A survey on green communication and security challenges in 5g wireless communication networks. *Journal of Network and Computer Applications*, 96:39–61, 2017.

[70] M. R. Garey and D. S. Johnson. *Computers and intractability*, volume 174. freeman New York, 1979.

[71] Jacques Gauvin and François Dubeau. Differential properties of the marginal function in mathematical programming. In *Optimality and Stability in Mathematical Programming*, pages 101–119. Springer, 1982.

[72] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[73] Asvin Gohil, Hardik Modi, and Shobhit K Patel. 5g technology of mobile communication: A survey. In *2013 international conference on intelligent systems and signal processing (ISSP)*, pages 288–292. IEEE, 2013.

[74] Bruce Golden. A problem in network interdiction. *Naval Research Logistics (NRL)*, 25(4):711–713, 1978.

[75] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Berlin [u.a]: Springer4060 XII, 362 S, 1988.

[76] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[77] Martin Grötschel, László Lovász, and Alexander Schrijver. Corrigendum to our paper ?the ellipsoid method and its consequences in combinatorial optimization? *Combinatorica*, 4(4):291–295, 1984.

[78] Akhil Gupta and Rakesh Kumar Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE access*, 3:1206–1232, 2015.

[79] Bo Han, Vijay Gopalakrishnan, Lusheng Ji, and Seungjoon Lee. Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine*, 53(2):90–97, 2015.

[80] Ryan B Hayward, Jeremy Spinrad, and R Sritharan. Weakly chordal graph algorithms via handles. 2000.

[81] John Homer, Su Zhang, Xinming Ou, David Schmidt, Yanhui Du, S Raj Rajagopalan, and Anoop Singhal. Aggregating vulnerability metrics in enterprise networks using attack graphs. *Journal of Computer Security*, 21(4):561–597, 2013.

[82] Jin Hong and Dong-Seong Kim. Harms: Hierarchical attack representation models for network security analysis. 2012.

[83] Heqing Huang, Su Zhang, Xinming Ou, Atul Prakash, and Karem Sakallah. Distilling critical attack graph surface iteratively through minimum-cost sat solving. In *Proceedings of the 27th Annual Computer Security Applications Conference*, pages 31–40. ACM, 2011.

[84] Hong Huang, Nihal Ahmed, and Pappu Karthik. On a new type of denial of service attack in wireless networks: The distributed jammer network. *IEEE Transactions on Wireless Communications*, 10(7):2316–2324, 2011.

[85] ISO/IEC. ISO/IEC 27005:2011 – Information technology – Security techniques – Information security risk management.

[86] Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.

[87] Gabriel Jakobson. Mission cyber security situation assessment using impact dependency graphs. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE, 2011.

[88] BD Jenkins. Security risk analysis and management. *Countermeasures, Inc*, 1998.

[89] Ari Juels et al. Rfid security and privacy: A research survey. *IEEE journal on selected areas in communications*, 24(2):381–394, 2006.

[90] R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[91] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[92] Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled Elbassioni, Vladimir Gurvich, Gabor Rudolf, and Jihui Zhao. On short paths interdiction problems: Total and node-wise limited interdiction. *Theory of Computing Systems*, 43(2):204–233, 2008.

[93] Nizar Kheir, Nora Cuppens-Boulahia, Frédéric Cuppens, and Hervé Debar. A service dependency model for cost-sensitive intrusion response. *Computer Security–ESORICS 2010*, pages 626–642, 2010.

[94] Phongphun Kijsanayothin and Rattikorn Hewett. Analytical approach to attack graph analysis for network security. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 25–32. IEEE, 2010.

[95] Diego Kreutz, Fernando MV Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76, 2015.

[96] Sc Li and Kw Zhang. Principles and applications of wireless sensor networks, 2008.

[97] Richard Paul Lippmann and Kyle William Ingols. An annotated review of past papers on attack graphs. Technical report, MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB, 2005.

[98] Roberto Lucchetti, F Mignanego, and G Pieri. Existence theorems of equilibrium points in stackelberg. *Optimization*, 18(6):857–866, 1987.

[99] Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

[100] A. R. Mahjoub. Polyhedral approaches. In *Concepts of Combinatorial Optimization, Volume 1*, pages 261–324. Wiley Online Library, 2010.

[101] Ali Ridha Mahjoub. Polyhedral approaches. *Concepts of Combinatorial Optimization*, pages 261–324, 2014.

[102] Sergio Mascetti, Letizia Bertolaja, and Claudio Bettini. A practical location privacy attack in proximity services. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 87–96. IEEE, 2013.

[103] Nick McKeown. How sdn will shape networking. *Open Networking Summit*, 2011.

[104] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[105] Vaibhav Mehta, Constantinos Bartzis, Haifeng Zhu, Edmund Clarke, and Jeannette Wing. Ranking attack graphs. In *RAID*, volume 4219, pages 127–144. Springer, 2006.

[106] Peter Mell, Karen Scarfone, and Sasha Romanosky. A complete guide to the common vulnerability scoring system version 2.0. In *Published by FIRST-Forum of Incident Response and Security Teams*, volume 1, page 23, 2007.

[107] Alexis L Motto, José M Arroyo, and Francisco D Galiana. A mixed-integer lp procedure for the analysis of electric grid security under disruptive threat. *IEEE Transactions on Power Systems*, 20(3):1357–1365, 2005.

[108] Aristides Mpitziopoulos, Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. A survey on jamming attacks and countermeasures in wsns. *IEEE Communications Surveys & Tutorials*, 11(4):42–56, 2009.

[109] Yong Niu, Yong Li, Depeng Jin, Li Su, and Athanasios V Vasilakos. A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges. *Wireless Networks*, 21(8):2657–2676, 2015.

[110] Steven Noel, Sushil Jajodia, Brian O'Berry, and Michael Jacobs. Efficient minimum-cost network hardening via exploit dependency graphs. In *Computer security applications conference, 2003. proceedings. 19th annual*, pages 86–95. IEEE, 2003.

[111] Xinming Ou, Wayne F Boyer, and Miles A McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345. ACM, 2006.

[112] Jiri Outrata, Michal Kocvara, and Jochem Zowe. *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*, volume 28. Springer Science & Business Media, 2013.

[113] M. W. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review 33*, pages 60–100, 1991.

[114] Thomas R Peltier. *Information security risk analysis*. Auerbach publications, 2010.

[115] Cynthia Phillips and Laura Painton Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 workshop on New security paradigms*, pages 71–79. ACM, 1998.

[116] Grant Purdy. Iso 31000: 2009?setting a new standard for risk management. *Risk analysis*, 30(6):881–886, 2010.

[117] Theodore S Rappaport, Shu Sun, Rimma Mayzus, Hang Zhao, Yaniv Azar, Kevin Wang, George N Wong, Jocelyn K Schulz, Mathew Samimi, and Felix Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE access*, 1:335–349, 2013.

[118] Tzvi Raz and David Hillson. A comparative review of risk management standards. *Risk Management*, 7(4):53–66, 2005.

[119] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R Wood, and Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, 1998.

[120] Layal Samarji, Frédéric Cuppens, Nora Cuppens-Boulahia, Wael Kanoun, and Samuel Dubus. Situation calculus and graph based defensive modeling of simultaneous attacks. *CSS*, 8300:132–150, 2013.

[121] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Verlag, 2003.

[122] Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

[123] Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. Sdn security: A survey. In *2013 IEEE SDN For Future Networks and Services (SDN4FNS)*, pages 1–7. IEEE, 2013.

[124] Vivek Shandilya, Chris B Simmons, and Sajjan Shiva. Use of attack graphs in security systems. *Journal of Computer Networks and Communications*, 2014, 2014.

[125] Piya Shedden, Tobias Ruighaver, and Atif Ahmad. Risk management standards œ the perception of ease of use. 2006.

[126] Hanif D Sherali, Allen L Soyster, and Frederic H Murphy. Stackelberg-nash-cournot equilibria: characterizations and computations. *Operations Research*, 31(2):253–276, 1983.

[127] Oleg M Sheyner. Scenario graphs and attack graphs. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2004.

[128] Kiyotaka Shimizu, Yo Ishizuka, and Jonathan F Bard. *Nondifferentiable and two-level mathematical programming*. Springer Science & Business Media, 2012.

[129] Mikko Siponen and Robert Willison. Information security management standards: Problems and solutions. *Information & Management*, 46(5):267–270, 2009.

[130] Gary Stoneburner, Alice Y Goguen, and Alexis Feringa. Sp 800-30. risk management guide for information technology systems. 2002.

[131] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1):1–11, 2011.

[132] J Tully. Mass adoption of the internet of things will create new opportunities and challenges for enterprises. 2015.

[133] Johan Van Benthem. *Logical dynamics of information and interaction*. Cambridge University Press, 2011.

[134] Valentina Viduto, Wei Huang, and Carsten Maple. Toward optimal multi-objective models of network security: Survey. In *Automation and Computing (ICAC), 2011 17th International Conference on*, pages 6–11. IEEE, 2011.

[135] Heinrich Von Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

[136] Shuzhen Wang, Zonghua Zhang, and Youki Kadobayashi. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. *Computers & security*, 32:158–169, 2013.

[137] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[138] Jeannette M Wing et al. Scenario graphs applied to network security. *Information Assurance: Survivability and Security in Networked Systems*, pages 247–277, 2008.

[139] Peng Xie, Jason H Li, Xinming Ou, Peng Liu, and Renato Levy. Using bayesian networks for cyber security analysis. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 211–220. IEEE, 2010.

[140] Haijun Zhang, Na Liu, Xiaoli Chu, Keping Long, Abdol-Hamid Aghvami, and Victor CM Leung. Network slicing based 5g and future mobile networks: mobility, resource management, and challenges. *IEEE Communications Magazine*, 55(8):138–145, 2017.

[141] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.

RÉSUMÉ

Dans cette thèse, nous proposons une nouvelle approche de gestion de risques pour les réseaux de télécommunications. Celle-ci est basée sur le concept de graphes d'analyse de risques appelés Risk Assessment Graphs (RAGs). Ces graphes contiennent deux types de noeuds : des points d'accès qui sont des points de départ pour les attaquants, et des noeuds appelés bien-vulnérabilité. Ces derniers doivent être sécurisés. La propagation potentielle d'un attaquant entre deux noeuds est représentée par un arc dans le RAG. Un poids positif représentant la difficulté de propagation d'un attaquant est associé à chaque arc. D'abord, nous proposons une approche quantitative d'évaluation de risques basée sur le calcul des plus courts chemins entre les points d'accès et les noeuds bien-vulnérabilité. Nous considérons ensuite un problème de traitement de risque appelé Proactive Countermeasure Selection Problem (PCSP). Etant donnés un seuil de difficulté de propagation pour chaque paire de point d'accès et noeud bien-vulnérabilité, et un ensemble de contremesures pouvant être placées sur les noeuds bien-vulnérabilité, le problème PCSP consiste à déterminer le sous ensemble de contremesures de coût minimal, de manière à ce que la longueur de chaque plus court chemin d'un point d'accès à un noeud bien-vulnérabilité soit supérieure ou égale au seuil de difficulté de propagation.
Nous montrons que le PCSP est NP-complet même quand le graphe est réduit à un arc. Nous donnons aussi une formulation du problème comme un modèle de programmation bi-niveau pour lequel nous proposons deux reformulations en un seul niveau: une formulation compacte basée sur la dualité en programmation linéaire, et une formulation chemins avec un nombre exponentiel de contraintes, obtenue par projection. Nous étudions cette deuxième formulation d'un point de vue polyhèdral. Nous décrivons différentes classes d'inégalités valides. Nous discutons l'aspect facial des inégalités de base et des inégalités valides. Nous concevons aussi des méthodes de séparation pour ces inégalités. En utilisant ces résultats, nous développons un algorithme de coupes et branchements pour le problème. Nous discutons enfin d'une étude numérique approfondie montrant l'efficacité des résultats polyhèdraux d'un point de vue algorithmique.

Notre approche s'applique à une large gamme de cas réels dans le domaine de télécommunications. Nous l'illustrons à travers plusieurs cas d'utilisation couvrant l'internet des objets (IoT), les réseaux orientés logiciel (SDN) et les réseaux locaux (LANs). Aussi, nous montrons l'intégration de notre approche dans une application web.

MOTS CLÉS

Gestion de la sécurité, systèmes de télécommunications modernes, programmation bi-niveau, approche polyhèdrale, facette, algorithme de coupes et branchements.

ABSTRACT

In this thesis, we propose a new risk management framework for telecommunication networks. This is based on the concept of Risk Assessment Graphs (RAGs). These graphs contain two types of nodes: access point nodes, or starting points for attackers, and asset-vulnerability nodes. The latter have to be secured. An arc in the RAG represents a potential propagation of an attacker from a node to another. A positive weight, representing the propagation difficulty of an attacker, is associated to each arc. First, we propose a quantitative risk evaluation approach based on the shortest paths between the access points and the asset-vulnerability nodes. Then, we consider a risk treatment problem, called Proactive Countermeasure Selection Problem (PCSP). Given a propagation difficulty threshold for each pair of access point and asset-vulnerability node, and a set of countermeasures that can be placed on the asset vulnerability nodes, the PCSP consists in selecting the minimum cost subset of countermeasures so that the length of each shortest path from an access point to an asset vulnerability node is greater than or equal to the propagation difficulty threshold.
We show that the PCSP is NP-Complete even when the graph is reduced to an arc. Then, we give a formulation of the problem as a bilevel programming model for which we propose two single-level reformulations: a compact formulation based on LP-duality, and a path formulation with an exponential number of constraints, obtained by projection. Moreover, we study the path formulation from a polyhedral point of view. We introduce several classes of valid inequalities. We discuss when the basic and valid inequalities define facets. We also devise separation routines for these inequalities. Using this, we develop a Branch-and-Cut algorithm for the PCSP along with an extensive computational study. The numerical tests show the efficiency of the polyhedral results from an algorithmic point of view.

Our framework applies to a wide set of real cases in the telecommunication industry. We illustrate this in several practical use cases including Internet of Things (IoT), Software Defined Network (SDN) and Local Area Networks (LANs). We also show the integration of our approach in a web application.

KEYWORDS

Security management, modern telecommunication systems, bilevel programming, polyhedral approach, facets, Branch-and-Cut algorithm.