Formulations and Algorithms for General and Security Stackelberg Games

Carlos Casorrán-Amilburu

UNIVERSITÉ LIBRE DE BRUXELLES
Faculté des Sciences
Département d'Informatique

UNIVERSIDAD DE CHILE
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Industrial

Formulations and algorithms for general and security Stackelberg games

Carlos Casorrán Amilburu

PhD Dissertation
submitted in partial fulfillment of the
requirements for the academic degrees of:
Docteur en Sciences (ULB)
Doctor en Sistemas de Ingeniería (UCh)

Academic year 2017 - 2018

A Santiago y Ana,

las personas que más admiro.

“Ya sabéis que yo en el mundo
por mi ciencia he merecido
el sobrenombre de docto.”

Basilio, rey de Polonia.

This thesis has been written under the joint supervision of Prof. Martine Labbé (ULB) and Prof. Fernando Ordóñez (UCh). The members of the jury are:

Declaration This thesis has been composed by the author himself and contains original work of his own execution. Some of the reported work has been done in collaboration with a number of co-authors. Chapters 4 and 5 have been co-authored with Professors Fortz, Labbé and Ordóñez. Chapters 6 and 7 have been co-authored with fellow Ph.D. candidate Víctor Bucarey, M.Sc. students Hugo Navarrete and Karla Rosas, Lieutenant Colonel Óscar Figueroa of the Carabineros de Chile and Professor Ordóñez.

Funding The author's research has been funded by the Belgian national scientific funding agency FNRS through a FRIA grant (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture). The research has been further supported by the COMEX project (Combinatorial Optimization: Metaheuristics & Exact Methods) and by ISCI (Instituto de Sistemas Complejos de Ingeniería).

Acknowledgments

I want to convey my sincere gratitude to Professors Martine Labbé and Fernando Ordóñez, supervisors of this work, for their guidance, tutelage and unwavering commitment. Their constant and meticulous proofreads of my work are sincerely appreciated.

I am very grateful to Professor Bernard Fortz, who co-authored the work presented in Chapters 4 and 5 of this thesis. His input was always welcomed.

I also want to thank the remaining members of my thesis committee for the time and effort devoted to my work. In particular, I want to convey my gratitude to Professor Yves Crama, who has been part of my *comité d'accompagnement* at ULB from the beginning of this thesis.

I would be remiss if I did not mention Professor Alfredo Marín, who encouraged me to pursue a Ph.D. degree in Brussels, Professor Antonio Rodríguez-Chía whose support and friendship have been invaluable to me and Professor Michele d'Adderio, who I've always looked up to.

Many thanks go out to all the organizers of the many workshops and conferences that make the Operations Research community the thriving community it is today.

These 4 years spent between Brussels and Santiago would not have been as enjoyable without the deep friendships developed inside and outside the workplace. My good friend Víctor Bucarey and his family deserve a very special mention for very graciously hosting me during my visits to the Chilean capital and playing the part of a surrogate family.

Finally, Diana and my parents mean the world to me, and their support and belief in me is invigorating.

Contents

Abstract

General Stackelberg games (GSG) confront two contenders, each wanting to optimize their rewards. One of the players, referred to as the leader, can commit to a given action or strategy first, and the other player, referred to as the follower, then responds by selecting an action or strategy of his own. The objective of the game is for the leader to commit to a reward-maximizing strategy, anticipating that the follower will best respond.

Finding an optimal mixed strategy for the leader in a GSG is NP-hard when the leader faces one out of a group of several followers and polynomial when there exists a single follower. Additionally, GSGs in which the strategies of the leader consist in covering a subset of at most m targets and the strategies of the followers consist in attacking some target, are called Stackelberg security games (SSG) and involve an exponential number of pure strategies for the leader.

The goal of this thesis is to provide efficient algorithms to solve GSGs and SSGs. These algorithms must not only be able to produce optimal solutions quickly, but also be able to solve real life, and thus large scale, problems efficiently.

To that end, the main contributions of this thesis are divided into three parts:

1. First, a comparative study of existing mixed integer linear programming (MILP) formulations is carried out for GSGs, where the formulations are ranked according to the tightness of their linear programming (LP) relaxations. A formal theoretical link is established between GSG and SSG formulations through projections of variables and this link is exploited to extend the comparative study to SSG formulations. A new strong SSG MILP formulation is developed whose LP relaxation is shown to be the tightest among SSG formulations. When restricted to a single attacker type, the new SSG formulation is ideal, *i.e.*, the constraints of its LP relaxation coincide with its convex hull of feasible solutions. Computational experiments show that the tightest formulations in each setting are the fastest. Notably, the new SSG formulation proposed is competitive with respect to solution time, and due to the tightness of its LP relaxation, it is better suited to tackle large instances than competing formulations.

2. Second, the bottleneck encountered when solving the formulations studied in the first part of the thesis is addressed: The tightest formulations in each setting have heavy LP relaxations which can be time-consuming to solve and thus limit the effectiveness of the formulations to tackle instances. To address this issue, in both the general and the security case, Benders cuts from the LP relaxation of the tightest MILP formulations are embedded into a Cut and Branch scheme on a sparse equivalent formulation in each setting. By combining the tightness of the bound provided by the strong formulations with the resolution speed of the formulations, the proposed algorithm efficiently solves large GSG and SSG instances which were out of the scope of previous methods.

3. Third, a special type of SSG, defined on a network, is studied, where the leader has to commit to two coverage distributions, one over the edges of the network and one over the targets, which are contained inside the nodes. A particular case of this SSG is used to tackle a real life border patrol problem proposed by the Carabineros de Chile in which the use of their limited security resources is optimized while taking into account both global and local planning considerations. A methodology is provided to adequately generate the game's parameters. Computational experiments show the good performance of the approach and a software application developed for Carabineros to schedule their border resources is described.

Résumé

Les jeux généraux de Stackelberg (GSG, par ses sigles en anglais) sont composés de deux joueurs qui s'affrontent, chacun essayant d'optimiser sa récompense. Un des joueurs, nommé le meneur, réalise la première action, et l'autre joueur, nommé le suiveur, réalise une autre action, en prenant en compte l'action du meneur. L'objectif du jeu est d'identifier la stratégie la plus favorable du meneur–celle qui maximise sa récompense–en anticipant que le suiveur va s'éngager dans une stratégie qui est une meilleure réponse à la stratégie choisie par le meneur.

Trouver une stratégie mixte optimale pour le meneur dans un GSG est un problème NP-difficile lorsque le meneur fait face à un suiveur parmi d'autres suiveurs et le problème devient polynômial quand il n'existe qu'un seul suiveur.

De plus, on parle de jeux de Stackelberg de sécurité (SSG, par ses sigles en anglais) lorsque les stratégies du meneur consistent à protéger un sous-ensemble de m cibles d'une collection alors que les stratégies des suiveurs consistent à attaquer une cible de cette collection. Ces jeux comprennent un nombre exponentiel de stratégies pures pour le meneur.

L'objectif de cette thèse est de créer des algorithmes efficaces pour la résolution des GSGs et SSGs. Ces algorithmes doivent être rapides et ils doivent fournir des solutions à des problèmes de très grande taille de façon efficace. Pour ce faire, les contributions principales de cette thèse sont divisées en trois parties.

1. Tout d'abord, nous comparons des formulations linéaires en variables entières-mixtes (MILP) pour la résolution de GSGs. Nous hiérarchisons les différentes formulations étudiées par rapport à la qualité de la borne supérieure fournie par la résolution de la relaxation linéaire du modèle. Nous établissons également un lien théorique entre les formulations pour des GSGs et SSGs en utilisant des projections de variables. Ce lien permet d'étendre l'étude comparative sur les formulations GSG aux formulations SSG. On propose aussi une nouvelle formulation MILP pour le jeu de sécurité, dont la borne supérieure fournie par la relaxation linéaire est la plus serrée. De plus, la restriction de cette formulation à un type d'attaquant est une formulation idéale, c'est a dire que les contraintes de sa relaxation linéaire coïncident avec l'enveloppe convexe de ses

solutions réalisables. Des éxperiences de calcul témoignent que les formulations dont la borne supérieure fournie par les relaxations linéaires correspondantes est proche de la valeur optimale, sont les formulations les plus performantes. Notamment, la nouvelle formulation MILP pour le jeu de sécurité est très performante au niveau du temps de résolution et à cause de la qualité de la borne supérieure fournie par sa relaxation linéaire, c'est la formulation la plus à mêne de résoudre des instances de très grande taille.

2. Ensuite, nous étudions le goulot d'étranglement auquel on est confronté lors de la résolution des formulations MILP considérées dans la première partie de cette thèse. Les formulations les plus performantes dans les deux cas–général et de sécurité–ont des relaxations linéaires denses au niveau du nombre de variables et de contraintes. Cela diminue l'efficacité des formulations à resoudre des problèmes de très grande taille. C'est pourquoi, dans chaque cas, on conçoit un algorithme de décomposition par lequel on renforce itérativement la relaxation linéaire de la formulation la moins dense avec des coupes de Benders obtenues à partir de la relaxation linéaire de la formulation la plus dense. En combinant la qualité de la borne supérieure fournie par la relaxation linéaire de la formulation dense avec la vitesse de résolution de la formulation la moins dense, les algorithmes de décomposition pour le cas général et le cas de sécurité sont beaucoup plus performants que les meilleures formulations de chaque cas, en résolvant des instances dont la taille dépasse celles pouvant être résolues par les formulations MILP étudiées dans la premère partie de la thèse.

3. Finalement, nous étudions un SSG défini sur un réseau. Le meneur doit s'engager dans deux stratégies de couverture, l'une sur les arcs du réseau et l'autre sur les cibles, qui sont contenues dans les sommets du réseau. Un cas particulier de ce jeu de sécurité est utilisé pour modéliser et résoudre un problème de patrouilles aux frontières proposé par Carabineros de Chile. Dans ce problème, les Carabineros doivent optimiser de manière globale l'utilisation de ses unités policères en prenant en compte des contraintes plus locales. Nous proposons une méthodologie mathématique pour générer les paramétres du jeu. Des éxperiences de calcul montrent une bonne performance de notre approche. Nous entamons une discussion sur le logiciel développé pour la planification des patrouilles aux frontières.

Resumen

Los Juegos de Stackelberg Generales (GSG, por sus siglas en inglés) enfrentan a dos contendientes, donde cada uno busca optimizar su propia recompensa. Uno de los jugadores, denominado líder, tiene la capacidad de comprometerse a una determinada estrategia primero, y el otro jugador, denominado seguidor, responde a la estrategia adoptada por el líder escogiendo su propia estrategia. El objetivo de dicho juego consiste en identificar la estrategia que maximice la recompensa del líder, teniendo en cuenta la capacidad del seguidor de responder a la estrategia del líder maximizando su propia recompensa.

Identificar una estrategia mixta óptima para el líder en un GSG es un problema NP-duro cuando el líder se enfrenta a varios seguidores y polinomial cuando existe un único seguidor. Más aún, juegos en los que las estrategias del líder consisten en proteger un subconjunto de a lo sumo m objetivos y las estrategias del seguidor consisten en atacar un objetivo, reciben el nombre de Juegos de Stackelberg de Seguridad (SSG, por sus siglas en inglés) e involucran un número exponencial de estrategias puras para el líder.

El objetivo primordial de la presente tesis es el de proporcionar algorítmos eficientes para la resolución de GSGs y SSGs. Estos algorítmos tienen que ser capaces tanto de proporcionar soluciones óptimas rápidamente, como de poder resolver problemas de índole real, y por tanto de gran tamaño, de forma eficaz.

A tal efecto, las principales contribuciones de este trabajo están divididas en tres bloques.

1. En primer lugar, para el caso general, llevamos a cabo un estudio comparativo de formulaciones enteras-mixtas conocidas que nos permite establecer un órden entre las diferentes formulaciones en función de lo ajustada que sea la cota, proporcionada por sus correspondientes relajaciones lineales, al valor óptimo. Establecemos una conexión teórica entre las formulaciones para el caso general y formulaciones para el caso de seguridad a través de proyecciones de variables. Haciendo uso de dicha conexión, desarrollamos una nueva formulación entera-mixta para SSGs, cuya relajación lineal proporciona la cota más ajustada al valor óptimo del problema. La restricción de dicha formulación a juegos con un único seguidor es una formulación ideal, *i.e.*, las restricciones de su relajación lineal describen la envoltura convexa de sus soluciones

factibles. Nuestros estudios computacionales muestran que las formulaciones más ajustadas, tanto en el caso general como en el de seguridad, son las más rápidas. En particular, la nueva formulación propuesta para SSGs, obtiene una cota ajustada al valor óptimo, siendo una formulación idónea para resolver instancias de gran tamaño.

2. En segundo lugar, analizamos la mayor complicación fruto del estudio anterior. Las formulaciones más ajustadas en cada caso tienen relajaciones lineales con muchas restricciones y variables haciendo que su resolución requiera de esfuerzo computacional. En particular, esta complicación limita el tamaño de las instancias que nuestros modelos pueden resolver. Para el caso general y para el caso de seguridad, desarrollamos sendos algorítmos de ramificación y corte sobre las formulaciones enteras-mixtas menos ajustadas donde sus relajaciones lineales son fortalecidas mediante cortes que provienen de descomposiciones de Benders sobre las relajaciones lineales de las formulaciones enteras-mixtas más ajustadas. Estos algorítmos combinan la rápidez de resolución de las formulaciones menos ajustadas con la calidad de la cota que proporcionan las relajaciones lineales de las formulaciones más ajustadas, haciendo de ellos, herramientas eficaces para la resolución de problemas de gran tamaño.

3. Por último, estudiamos un tipo de SSG definido sobre una red, en el que el líder debe determinar dos frecuencias de cobertura, una sobre los ejes de la red y otra sobre los objetivos a proteger que se encuentran en los nodos de la red. Empleamos un caso particular de este problema para la resolución de un problema real de patrullaje de fronteras propuesto por Carabineros de Chile en el que se debe lidiar con una planificación a gran escala, respetando requerimientos locales. Proporcionamos una metodología para la generación de los parametros del juego consistente con el problema que modelamos. Nuestros estudios computacionales indican la idonéidad de nuestro planteamiento. Describimos en detalle el software implementado para Carabineros y llevamos a cabo una evaluación completa.

Chapter 1

Introduction

Heinrich Freiherr von Stackelberg (October 31st, 1905–October 12th, 1946) was a Russian-born German economist with Argentinian and Spanish ancestry. He contributed greatly to the mathematical field of Game Theory and Industrial Organization [Möller, 1948].

Figure 1.0.1: Heinrich Freiherr von Stackelberg

His most relevant contribution–that of Stackelberg competition–proposes what is known as a *duopoly leadership model*. Stackelberg competition models a market where there are two strategic firms that must decide how much quantity of product they must produce in order to maximize profits. Very generally, in this model one firm assumes the role of market leader, and can therefore commit to a production level first, while the other firm, can observe the market leader's commitment and then choose a production level which is a best response to the observed market leader's strategy.

General Stackelberg games (GSG) (we specify the term 'general' to distinguish these games from the particular subclass known as Stackelberg security games, which will be introduced further on), in fact, can model any adversarial situation between players with different objectives, each striving to optimize a certain payoff in a sequential, one-off encounter. One of the players can commit to a given action first and he is referred to as the leader, whereas the other player, which responds to the leader's action, is referred to as the follower. The idea of Stackelberg competition has been widely used in many different fields since its original inception in the area of economics, with particular prominence in the fields of telecommunications [Bloem et al., 2007, Zhang and Zhang, 2009, Roh et al., 2011], transportation [Labbé et al., 1998, Cardinal et al., 2009], fare evasion systems [Yin et al., 2012, Correa et al., 2014], theory of incentives [Salanié, 2005, Laffont and Tirole, 1993, Laffont and Martimort, 2009], among many many others.

Stackelberg competition has found many challenging problems to tackle in the domain of security. The arena of security, adversarial by nature, is a prime candidate for Stackelberg theory. The great deal of attention this field continues to attract is partly due to the dangers our society faces of late. Terrorism, drug trafficking, crime. These problems are ubiquitous. Yet, limited security resources cannot be everywhere at all times, raising the question of how best to utilize them.

TeamCore, a research group at the University of Southern California, led by Milind Tambe, has been at the forefront of this challenging domain over the last 10 years, developing Stackelberg-related theory, methodology and software to address many pressing concerns in many security environments. They first tackled the problem of strategically placing road controls on inbound roads at the Los Angeles LAX airport and deciding when and how to deploy canine units to the different airport terminals, [Pita et al., 2008]. Then, they tackled the much larger problem of strategically assigning US Federal Air Marshals to transatlantic flights [Jain et al., 2010b] and the problem of determining optimal patrols for the US Coast Guard so as to ensure efficient protection of critical port infrastructure [Shieh et al., 2012]. They further developed a game-theoretic software with which to deploy security resources across 400 United States airports [Pita et al., 2011]. They have also used their security expertise to develop and deploy Stackelberg software to protect endangered wildlife in natural reserves [Yang et al., 2014]. In addition, they have exploited their work on physical security problems to tackle cybersecurity problems that can be modeled as Stackelberg games [Sinha et al., 2015].

This thesis is dedicated to the algorithmic and theoretical analysis of an important class of problems related to security. The framework chosen to study said problems is that of Game Theory and, more specifically, Stackelberg games. The main contributions of the thesis are theoretical, algorithmic and practical. We study mathematical formulations and enhance them through the use of well established integer programming techniques, we exploit problem structure to develop valid inequalities for the formulations and develop decomposition approaches and we present a Stackelberg approach to tackle a real life border patrol problem along the border of the northernmost province of Chile. The theoretical and algorithmic advancements are aimed at speeding up problem resolution with respect to state of the art methods and to scale-up the sizes of the instances that can be currently solved.

This thesis is arranged as follows. In Chapter 2, we provide a formal definition of the problems studied in this dissertation. In Chapter 3, we explore the state of the art by performing a brief literature review.

In Chapter 4, we concentrate on previously existing Mixed Integer Linear Programming (MILP) formulations for general purpose Stackelberg bi-matrix games between two generic players, a leader and a follower, and enhance them through the use of standard integer programming techniques. The enhancements presented have a theoretical impact–tighter linear relaxation bounds–and these enhancements, in turn, have an effect on the practical performance of the formulations. We establish a theoretical link between the general purpose Stackelberg games and Stackelberg security games, which as we will soon see, have a very particular structure. This theoretical link, allows to extend the enhancements on general Stackelberg games to the security setting.

In Chapter 5, we address the bottleneck encountered in Chapter 4–while the enhanced formulations provide a very tight integer gap, they have heavy linear programming (LP) relaxations. We thus exploit problem structure to develop algorithms for the general and security settings which embed Benders cuts from the large but strong linear relaxations of the enhanced formulations into a Cut and Branch solving scheme based on much sparser and weaker equivalent formulations. Further, we present the scaling up capabilities of the proposed decomposition approaches with respect to competing methods in the literature.

In Chapter 6, we study a specific Stackelberg security game defined on a graph, and present a compact MILP formulation for this problem. We further study two sampling strategies to recover an implementable defender strategy from the compact solution returned by the formulation. We compare the accurateness of the sampling strategies presented and the performance of the proposed formulation through computational tests.

In Chapter 7, we present a case study, where a Stackelberg security software was developed for Carabineros de Chile to tackle a real life border patrol setting along the Chilean border. We describe a parameter estimation methodology to model the border games we solve and describe in detail the software developed. We discuss how to conduct a complete evaluation of the border patrol software presented.

The remainder of this chapter is devoted to providing a brief and somewhat informal overview of the two deeply connected fields of knowledge on which this thesis heavily relies on: Game Theory and Bilevel Programming.

1.1 Game Theory

Two-player games will be represented throughout this thesis as normal-form games. A normal-form game is also known as a strategic game, or more commonly as a bimatrix-form game. The two players, called the row and column players, respectively, in the most basic form of the game, have a finite set of actions called pure strategies from which they must each select one. The strategies for the row player are arranged as rows of a payoff matrix and the strategies for the column player are arranged as the columns of this payoff matrix. Once both players have committed to their actions, this creates a profile, a vector at the intersection of the chosen rows and columns with the payoffs for each player in that particular play of the game.

Table 1.1.1 shows the bimatrix representation of the popular rock-paper-scissors game. In this game, players play simultaneously and each player selects one among his different pure strategies: rock, paper or scissors. The rules in this game are that scissors cut paper, paper covers rock and rock crushes scissors. The bimatrix representation of the game in Table 1.1.1 is consistent with the payoffs associated to the actions: a player receives a payoff of 1 when he wins and a payoff of -1 when he loses.

	R	P	S
R	$(0,0)$	$(-1,1)$	$(1,-1)$
P	$(1,-1)$	$(0,0)$	$(-1,1)$
S	$(-1,1)$	$(1,-1)$	$(0,0)$

Table 1.1.1: Bimatrix for Rock-Paper-Scissors game

More formally, let I and J denote the set of pure strategies for the row and column players respectively and let (R, C) be the game bimatrix where $R, C \in \mathbb{R}^{|I|\times|J|}$. A more elaborate action that players can take when the game is played in a repeated fashion consists in selecting each pure strategy with a given probability. Such an action is called a mixed strategy. Consider the $|I|$ and $|J|$-th dimensional simplices:

$$\begin{aligned}\mathbb{S}^{|I|} &= \{x \in [0,1]^{|I|} : \sum_{i\in I} x_i = 1\},\\ \mathbb{S}^{|J|} &= \{q \in [0,1]^{|J|} : \sum_{j\in J} q_j = 1\},\end{aligned}$$

then, $x \in \mathbb{S}^{|I|}$ is a mixed strategy for the row player, where the i-th pure strategy is played with probability x_i and $q \in \mathbb{S}^{|J|}$ is a mixed strategy for the column player, where the j-th pure strategy is played with probability q_j.

The solution concept used throughout this thesis, is the Stackelberg Equilibrium (SE). An SE is the solution concept of choice when the game being played is sequential instead of simultaneous. In these sequential games, the row player, now referred to as the leader can commit to a strategy, before the column player, now referred to as the follower. The follower observes the leader's mixed strategy and best responds by selecting a strategy of his own.

Leitman distinguishes two kinds of Stackelberg equilibria in [Leitman, 1978], the so-called weak and strong equilibria. The strong form assumes that when the follower is indifferent among best responses, he will select the best response that maximizes the leader's utility, while the weak form, instead, assumes that he will select the best response which minimizes the leader's utility. In the literature, the strong form of the equilibrium is preferred, since its existence is always guaranteed, while the weak form's isn't [Leitman, 1978]. We will later see, with an example, that this does not represent a big loss of generality, since the leader can often incentivize the follower to break ties in his favor by playing a sub-optimal strategy, arbitrarily close to the optimal strategy, [von Stengel and Zamir, 2004].

Define the mapping $\mathcal{B} : \mathbb{S}^{|I|} \longrightarrow \mathbb{S}^{|J|}$ as a mapping that, given the leader's mixed strategy x, returns a follower's best response strategy $\mathcal{B}(x)$, which need not be unique. A Strong Stackelberg Equilibrium (SSE) is then defined as follows:

Definition 1.1.1. *A profile of mixed strategies $(x, \mathcal{B}(x))$ form a Strong Stackelberg Equilibrium if they satisfy the following conditions:*

1. *The leader always plays a payoff-maximizing strategy:*

$$x^T R\mathcal{B}(x) \geq x'^T R\mathcal{B}(x') \; \forall x' \in \mathbb{S}^{|I|}.$$

2. *The follower always plays a best-response, $\mathcal{B}(x) \in F(x)$, where,*

$$F(x) = \arg\max_q \{x^T C q : q \in \mathbb{S}^{|J|}\}$$

is the set of best responses for the follower.

3. *The follower breaks ties optimally in favor of the leader:*

$$x^T R\mathcal{B}(x) \geq x^T R q \; \forall q \in F(x)$$

An interesting remark that immediately follows from the above definition is the fact that even when the leader plays a mixed strategy, there always exists a follower best response which is a pure strategy. This is a direct consequence of the sequentiality of the game.

The follower is the last player to play the game and therefore doesn't stand to gain by randomizing over his strategies. A more formal statement of this remark and its very simple proof are given next.

Remark 1.1.1. *For any leader strategy x, there exists a best response for the follower that is given by a vector $q \in \{0,1\}^{|J|} : \sum_{j \in J} q_j = 1$.*

Proof. Assume that $\mathcal{B}(x) = \bar{q} \notin \{0,1\}^{|J|}$. Consider the set $\bar{J} = \{j \in J : \bar{q}_j > 0\}$. Then, each canonical vector e^j for $j \in \bar{J}$ is also a best response vector, *i.e.*, for all $j \in \bar{J}$, $e^j \in F(x)$ and $x^T R e^j \geq x^T R q$ for all $q \in F(x)$.
Since $\bar{q} = \sum_{j \in J} \bar{q}_j e^j$ and $e^j \in \mathbb{S}^{|J|}$, it follows that:

$$x^T C \bar{q} = \sum_{j \in J} \bar{q}_j (x^T C e^j) \leq \sum_{j \in J} \bar{q}_j (x^T C \bar{q}) = x^T C \bar{q},$$

where the inequality is a consequence of $\bar{q} = \mathcal{B}(x)$. Therefore, for all $j \in \bar{J}$, $x^T C e^j = x^T C \bar{q}$ so that $e^j \in F(x)$. An analogous argument provides that for all $j \in \bar{J}$, $x^T R e^j = x^T R \bar{q}$, concluding that for each $j \in \bar{J}$, e^j is a follower best response to x. ■

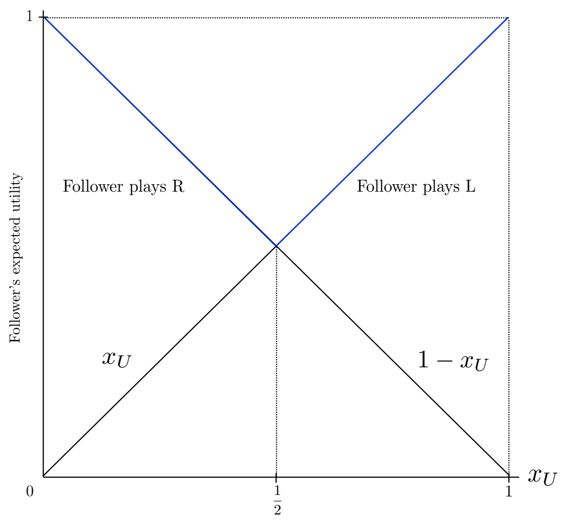
Consider the game defined in Table 1.1.2 where the pure strategies for the leader are $\{U, D\}$ and the pure strategies for the follower are $\{L, R\}$.

	L	R
U	$(2,1)$	$(4,0)$
D	$(1,0)$	$(3,1)$

Table 1.1.2: Example game

If one solves for a pure strategy Stackelberg equilibrium, the following analysis is conducted. If the leader commits to U, the follower will prefer to play L over R as by playing L he secures 1 unit of payoff. This will provide a payoff of 2 units to the leader. Similarly, if the leader commits to D, the follower will prefer to play R over L, thus securing 1 unit of profit. In this case, the leader will receive 3 units of payoff. It is in the leader's best interest to incentivize the follower to play R as that provides him with a higher payoff. He can do so by playing D. In this game the strategy profile (D, R) is a pure strategy Stackelberg equilibrium with payoffs for the leader and follower of 3 units and 1 unit, respectively.

The leader can do even better by playing a mixed strategy, *i.e.*, strategically randomizing over his two strategies. In this case, the analysis is a bit more involved. Suppose the leader plays U with probability x_U and D with probability x_D (where, of course, $x_D = 1 - x_U$). From Remark 1.1.1, it suffices to compute the expected payoff for the follower given that



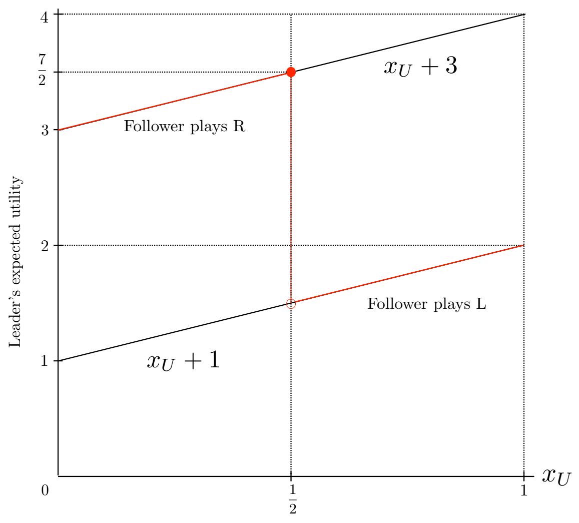


Figure 1.1.1: Follower's and leader's expected utilities when the leader commits to playing $x = (x_U, 1 - x_U)$

the leader plays mixed strategy $x = (x_U, 1-x_U)$ and the follower responds by either playing L or R:

$$\begin{aligned}
\mathbb{E}(\text{Follower's utility}|\text{fol. plays } L \text{ as a response to } x) = 1x_U + 0(1 - x_U) &= x_U, \\
\mathbb{E}(\text{Follower's utility}|\text{fol. plays } R \text{ as a response to } x) = 0x_U + 1(1 - x_U) &= 1 - x_U.
\end{aligned}$$

Observe that the follower is indifferent between L and R when he receives the same payoff from both, namely, when $x_U = 1 - x_U$, in other words, when $x_U = \frac{1}{2}$. When $x_U < \frac{1}{2}$, the follower obtains a higher profit by playing R since $1 - x_U > x_U$. Similarly, when $x_U > \frac{1}{2}$, the follower favors L because he gets a higher payoff as $x_U > 1 - x_U$. The graph on the left in Figure 1.1.1 shows the expected profit for the follower as the values of x_U change from 0 to 1. With this information about how the follower reacts, the leader can determine his payoff maximizing strategy to commit to. One can compute the expected payoff for the leader when he commits to a mixed strategy $x = (x_U, 1 - x_U)$ and the follower reacts by playing L or playing R:

$$\begin{aligned}
\mathbb{E}(\text{Leader's utility}|\text{leader plays } x \text{ and fol. plays } L) = 2x_U + 1(1 - x_U) &= x_U + 1, \\
\mathbb{E}(\text{Leader's utility}|\text{leader plays } x \text{ and fol. plays } R) = 4x_U + 3(1 - x_U) &= x_U + 3.
\end{aligned}$$

We plot the expected utility for the leader in the graph on the right in Figure 1.1.1. Note that when $x_U < \frac{1}{2}$, the follower plays R and when $x_U > \frac{1}{2}$, the follower plays L. The red line indicates the expected utility the leader receives as he chooses x_U between 0 and 1.

The overall maximum of the red line is attained at $x_U = \frac{1}{2}$, which coincides with when the follower is indifferent between strategies. It is thus crucial to understand how the follower reacts in this situation. If one considers the SSE, then the follower breaks his indifference in favor of the leader, *i.e.*, he plays R and the leader receives a payoff of $\frac{7}{2}$ units. In this case $x = (\frac{1}{2}, \frac{1}{2})$ is the optimal mixed strategy for the leader. In the weak version of the equilibrium the follower breaks his indifference by selecting the strategy that provides the leader with the least payoff, in this case, L. Under this assumption, an optimal mixed strategy does not exist, as the leader can always improve his payoff by selecting some suboptimal strategy $(\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon)$ for a sufficiently small $\varepsilon > 0$. However, by selecting this suboptimal strategy the leader can incentivize the follower into breaking ties in favor of the leader when $\varepsilon \to 0$.

Table 1.1.3 summarizes the existence of the two equilibria discussed under pure and mixed strategies.

	Pure strategies	Mixed Strategies
WSE	ALWAYS	NOT ALWAYS
SSE	ALWAYS	ALWAYS

Table 1.1.3: Is the existence of the solution concepts guaranteed in finite games?

Another well known equilibrium concept is the Nash equilibrium (NE), named after John Nash and first introduced by Antoine Cournot in [Cournot et al., 1897]. A Nash equilibrium is the preferred solution concept when players play simultaneously and is defined as follows:

Definition 1.1.2. *Let (x^*, q^*) be an array of strategy choices, one for each player. In addition, let $\pi_R(x^*, q^*) = {x^*}^T R q^*$ be the payoff for the row player and let $\pi_C(x^*, q^*) = {x^*}^T C q^*$ be the payoff for the column player when (x^*, q^*) is the chosen profile. An array of strategy choices is a Nash equilibrium if:*

$$\begin{aligned} \pi_R(x^*, q^*) &\geq \pi_R(x, q^*) \text{ for any } x \in \mathbb{S}^{|I|} \text{ and} \\ \pi_C(x^*, q^*) &\geq \pi_C(x^*, q) \text{ for any } q \in \mathbb{S}^{|J|}. \end{aligned}$$

In other words, a Nash equilibrium is reached when it is not in any player's interest to unilaterally deviate from the current strategy profile. A pure strategy NE may not exist, but if players play mixed strategies, a NE always exists [Nash, 1950].

The game in Table 1.1.2 has a pure strategy NE, attained by the profile (U, L). If the row player plays U, the best response for the column player is to play L. If the row player plays D, the column player's best response is then R. If the column player, plays L, the row player prefers to play U and if the column player plays R, the row player then prefers U. So when (U, L) is played, no player has the incentive to unilaterally deviate from their strategy as it would lead to worse results for the player that deviates. Further, it can be shown that (U, L) is the only NE, even when mixed strategies are considered, leading to a row player utility of 2 units.

The observant reader may have noticed that the row player/leader stands to gain a higher utility when the game in Table 1.1.2 is played sequentially instead of simultaneously. The following clarifying remarks are in order:

Remark 1.1.2. *When players can only select pure strategies in a two player finite bimatrix game, and the best response set for the follower is a singleton for any pure strategy selected by the leader, by committing to a pure strategy Stackelberg Equilibrium, the leader is never worse off than if he plays the Nash Equilibrium strategy, when such a strategy exists.*

Remark 1.1.3. *When players can select mixed strategies in a two player finite bimatrix game, committing to a Stackelberg Equilibrium is at least as good for the leader as playing the simultaneous game and playing the Nash Equilibrium.*

For an in-depth comparison of Nash and Stackelberg equilibria, the reader is directed to [Yin et al., 2010]. It should also be noted that Stackelberg equilibria, when they exist, need not be unique in a two player finite bimatrix game. The payoff the leader obtains, on the other hand, is unique regardless of the SE selected.

When determining the rules of a given game, it is important to determine whether players have complete or incomplete information. Complete information assumes that each player is fully aware of the payoff matrix of the game. A game is said to have incomplete information when players know their own payoff values but are unsure about the payoff values of their opponent. It may happen that players have some statistical information about the other players.

Throughout this thesis, unless explicitly stated, we concentrate on leader-follower sequential finite bimatrix games where the leader has incomplete information on the follower; he is unsure about how the follower values his different strategies. This shortcoming is tackled by assuming that the leader knows a probability distribution over finitely many distinct follower types. For each follower type, the leader has full knowledge of that follower type's payoff values and his own when playing against that follower type. Each follower type is aware of how they each value their different strategies and are aware of how the leader values his. Further, each follower type is also aware of the mixed strategy that the leader commits to. These games are referred to as p-follower or Bayesian games. The same paradigm can be used to describe a game with complete information between a leader and several distinct follower agents, of which only one plays against the leader according to a probability distribution known by the leader.

Finally, note that Definition 1.1.1 can easily be extended to p-follower games. The leader now commits to an expected reward-maximizing strategy anticipating that each follower type will best respond against the selected strategy. Further, when indifferent between follower best responses, each follower type will break ties in favor of the leader. One can also easily extend Remark 1.1.1 to the p-follower case and it follows that each follower type's best response is always a pure strategy.

1.2 Bilevel Programming

In the field of mathematical programming, the sequentiality of the Stackelberg problem has been addressed by Bilevel Programming (BP). Introduced in the early seventies [Bracken and McGill, 1973], BP targets hierarchical optimization problems in which part of the constraints translate the fact that some of the variables constitute an optimal solution to another nested optimization problem.

In this setting, the first objective function and its proper constraints constitute the so-called leader level or first level, while the nested optimization problem, characterized by a proper objective function and constraints, is referred to as the follower level or second level. This captures the sequentiality which is intrinsic to a Stackelberg game where the leader commits to a reward maximizing strategy in the first level and then the follower, in the second level, takes into account the first level decision, and then commits to a feasible strategy according to the second level constraints which selfishly optimizes his own objective function.

In general, let x and y denote decision vectors, f and g objective functions and X and Y the feasible solution sets of the leader and the follower respectively. The general BP problem can be formulated as follows:

$$\begin{array}{ll}
\text{Max}_{x,y} & f(x,y) \\
& (x,y) \in X, \\
& y \in S(x), \\
\text{where} & S(x) = \arg\max\limits_{y} g(x,y), \\
& \text{s.t. } (x,y) \in Y(x).
\end{array}$$

The leader maximizes f by committing to a feasible x and anticipating that, given x, the follower's best response, obtained by optimizing the second level problem, is given by y. Thus, the optimal decision vectors returned by BP are a pair of mutual best responses.

From a computational standpoint, BP problems are difficult to solve. Even the simplest

BP problem with linear objective functions and linear constraints has been shown to be $\mathcal{NP}$-hard [Jeroslow, 1985]. Since the 1980s, several algorithms have been proposed for solving these problems, under specific conditions on the objective functions and constraints. Important surveys of proposed solution methods are those by [Kolstad, 1985], [Savard, 1989], [Anandalingam and Friesz, 1992] and [Labbé and Violin, 2016]. An extensive bibliography review is due to [Vicente and Calamai, 1994].

BP provides an ideal framework to mathematically model Stackelberg games. A solution approach to BP consists in characterizing the optimality of the second level problem in order to obtain a single level optimization problem. In general, this manipulation leads to the so-called Mathematical Problems with Complementarity Constraints (MPEC) [Bouza, 2006, Kanzow and Schwartz, 2010]. These nonlinear problems are known to be difficult optimization problems as many of the standard constraint qualifications are violated making it difficult to characterize optimality of the second level. To circumvent this problem, special constraint qualifications are used to provide optimality conditions under some meaningful assumptions [Flegel and Kanzow, 2005a, Flegel and Kanzow, 2005b].

Our approach, and that used by other authors in the field of Stackelberg game theory ([Paruchuri et al., 2008], [Kiekintveld et al., 2009]), is based on using integer programming techniques to manipulate the difficult complementarity constraints that arise when characterizing optimality of the second level problem, to obtain single level mixed integer linear programming formulations.

Chapter 2

Problem definition

In this chapter, we formally define p-follower general Stackelberg games (GSGs) and their extension to the security setting, p-follower security Stackelberg games (SSGs).

We further motivate some of the work that will be carried out in Chapter 4 regarding SSGs by exploring the link that can be established between GSGs and SSGs. We also present a simple–yet pertinent–algorithm that, given a solution to a SSG, constructs a solution to the corresponding GSG.

2.1 General Stackelberg games–GSGs

Let K be the set of p followers. We denote by I the set of leader pure strategies and by J the set of follower pure strategies. The leader has a known probability of facing follower $k \in K$, denoted by $\pi^k \in [0,1]$. We denote the n-dimensional simplex by $\mathbb{S}^n = \{x \in [0,1]^n : \sum_n x_i = 1\}$. A mixed strategy for the leader consists in a vector $x \in \mathbb{S}^{|I|}$ such that for $i \in I$, x_i is the probability with which the leader plays pure strategy i. Analogously, a mixed strategy for a follower $k \in K$ is a vector $q^k \in \mathbb{S}^{|J|}$ such that, q^k_j is the probability with which follower k replies with pure strategy $j \in J$. The rewards or payoffs for the leader and each follower, resulting from their choice of strategy, are encoded in a different matrix for each follower. These payoff matrices are denoted by (R^k, C^k), where $R^k \in \mathbb{R}^{|I|\times|J|}$ is the leader's reward matrix when facing follower $k \in K$ and $C^k \in \mathbb{R}^{|I|\times|J|}$ is the reward matrix for follower k. The expected reward of the leader and follower k, respectively, can be expressed as follows:

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} x_i q^k_j, \tag{2.1.1}$$

$$\sum_{i\in I}\sum_{j\in J} C^k_{ij} x_i q^k_j, \quad \forall k \in K. \tag{2.1.2}$$

The solution concept used in these games, as discussed in the introduction, is the Strong Stackelberg Equilibrium (SSE).

In Mathematical Optimization, Stackelberg games are addressed by Bilevel Programming (BP). The following model, (BIL-p-G$_{x,q}$), is a Bilevel Programming formulation for the general Stackelberg game problem:

(BIL-p-G$_{x,q}$)

$$\text{Max}_{x,q} \quad \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} \pi^k R^k_{ij} x_i q^k_j \tag{2.1.3}$$

$$\text{s.t.} \quad \sum_{i\in I} x_i = 1, \tag{2.1.4}$$

$$x_i \in [0,1] \qquad \forall i \in I, \tag{2.1.5}$$

$$q^k \in \arg\max_{r^k} \left\{ \sum_{i\in I}\sum_{j\in J} C^k_{ij} x_i r^k_j \right\} \qquad \forall k \in K, \tag{2.1.6}$$

$$r^k_j \in \{0,1\} \qquad \forall j \in J, \forall k \in K, \tag{2.1.7}$$

$$\sum_{j\in J} r^k_j = 1 \qquad \forall k \in K. \tag{2.1.8}$$

The objective function maximizes the leader's expected reward. Constraints (2.1.4)-(2.1.5) characterize the mixed strategies considered by the leader. The second level problem defined by (2.1.6)-(2.1.8) indicates that the follower maximizes his own payoff by best responding with a pure strategy to the leader's commitment. If there are multiple optimal strategies for the follower, the main level problem selects one that maximizes the objective of the leader.

2.2 Stackelberg security games–SSGs

A Stackelberg security game (SSG) is a specific case of a GSG where the pure strategies for the leader, now referred to as the defender, involve allocating a limited number of security resources (law enforcement officers, for example) to protect a subset of targets (critical infrastructure, for example) and the pure strategies for each follower type consist in attacking a single target.

Formally, let J be the set of n targets that could be attacked and let Ω be the set of $m < n$ security resources available to protect these targets. Allocating resource $\omega \in \Omega$ to target $j \in J$ protects the target. The set I of defender pure strategies $i \in I$ is composed of all $\sum_{i=1}^{m} \binom{n}{i}$ subsets of at most m targets of J that the defender can protect simultaneously. The elements $j \in J$ constitute the pure strategies of each attacker.

In SSGs, payoffs for the players only depend on whether a target is attacked and whether that target was covered or not. This means that many of the strategies have identical payoffs. This fact is used to construct a compact representation of the payoffs.

We denote by D^k the utility of the defender when facing an attacker $k \in K$ and by A^k the utility of attacker k. Associated with each target and each player are two payoffs depending on whether or not the target is covered, see Table 2.2.1. Further, it is generally assumed that for each $j \in J$ and $k \in K$, $D^k(j|c) \geq D^k(j|u)$ and $A^k(j|u) \geq A^k(j|c)$, *i.e.*, it is more beneficial for the defender to receive an attack on a protected target instead of suffering an attack on an unprotected target and, similarly, it is more beneficial for the attacker to attack an unprotected target instead of a protected one.

	Covered	Uncovered
Defender	$D^k(j\|c)$	$D^k(j\|u)$
Attacker	$A^k(j\|c)$	$A^k(j\|u)$

Table 2.2.1: Payoff structure in an SSG when target j is attacked by an attacker k

The authors in [Kiekintveld et al., 2009] take advantage of the aforementioned compact representation to define a coverage vector $c \in [0,1]^{|J|}$ whose components, c_j, represent the probability of coverage of target j. The components of the vector c satisfy $c_j = \sum_{i \in I: j \in i} x_i$, $\forall j \in J$, *i.e.*, the frequency of coverage is expressed as the sum of all probabilities of the strategies that assign coverage to that target. Variables q_j^k indicate whether an attacker k strikes a target j.
The defender's and attacker k's expected rewards, are, respectively:

$$\sum_{j \in J} \sum_{k \in K} \pi^k q_j^k \{c_j D^k(j|c) + (1 - c_j) D^k(j|u)\}, \tag{2.2.1}$$

$$\sum_{j \in J} q_j^k \{c_j A^k(j|c) + (1 - c_j) A^k(j|u)\}, \quad \forall k \in K. \tag{2.2.2}$$

As with GSGs, such a game can be modeled by means of Bilevel Programming.

(BIL-p-S$_{x,c,q}$)

$$\text{Max} \quad \sum_{j \in J} \sum_{k \in K} \pi^k q_j^k \{c_j D^k(j|c) + (1 - c_j) D^k(j|u)\}$$

$$\text{s.t.} \quad \sum_{i \in I} x_i = 1, \tag{2.2.3}$$

$$x_i \geq 0 \qquad \forall i \in I, \tag{2.2.4}$$

$$\sum_{i \in I: j \in i} x_i = c_j \qquad \forall j \in J, \tag{2.2.5}$$

$$q^k \in \arg\max_{r^k} \left\{ \sum_{j\in J} r_j^k (c_j A^k(j|c) + (1-c_j)A^k(j|u) \right\} \qquad \forall k \in K,$$

$$r_j^k \in \{0,1\} \qquad \forall j \in J, \forall k \in K,$$

$$\sum_{j\in J} r_j^k = 1 \qquad \forall k \in K.$$

The objective function maximizes the defender's expected reward. Constraints (2.2.3)-(2.2.5) characterize the exponentially many mixed strategies considered by the defender and relate them to coverage frequencies over the targets. The remaining constraints constitute the second level optimization problem which ensures that the attacker maximizes his profit by attacking a single target, best responding to the defender's selected strategy.

Remark that a more compact formulation–one involving a polynomial number of variables and constraints–can be obtained if projecting out the exponentially many x variables does not lead to exponentially many constraints. To show that the number of constraints in the projection is polynomial, we provide a mathematical proof in Chapter 4 that makes use of Farkas' Lemma [Farkas, 1902]. Projecting out the x variables from Constraints (2.2.3)-(2.2.5) leads to the following constraints:

$$\sum_{j\in J} c_j \leq m, \tag{2.2.6}$$

$$c \in [0,1]^{|J|}. \tag{2.2.7}$$

Indeed, replacing Constraints (2.2.3)-(2.2.5) by (2.2.6)-(2.2.7) gives a polynomial size formulation involving only the c and the q variables. Note that Constraint (2.2.6) enforces that the total coverage provided to the targets cannot exceed m, the number of available security resources and Constraint (2.2.7) guarantees that the coverage probabilities are, in fact, probabilities and their range is thus restricted to $[0,1]$.

Further, given an optimal solution to such a compact formulation–an optimal coverage vector c and an optimal attack vector q–a probability vector x, solution to this game in extensive form, can be obtained by solving the system of linear inequalities defined by (2.2.3), (2.2.4) and (2.2.5). As this system involves $n+1$ equalities, there exists a solution in which the number of variables x_i with a positive value is not larger than $n+1$, *i.e.*, the output size of an SSG, under extensive form, is polynomial in the input size.

In the next section, we present a pictorial algorithm that, given optimal coverage probabilities on the targets, recovers an optimal mixed strategy that complies with said optimal coverage probabilities. In addition, the algorithm's validity provides a proof that projecting out the x variables from Constraints (2.2.3)-(2.2.5) leads to (2.2.6)-(2.2.7).

2.2.1 Recovering an optimal mixed strategy for the defender–The box method

The box method algorithm presented in this section provides a simple graphic way of recovering an optimal mixed strategy x, solution to the GSG, and which is easily implementable by the leader, based on the vector c of optimal coverage probabilities returned by the compact SSG.

The box method algorithm's construction and validity comes from a result, and its constructive proof, for scheduling problems found in [McNaughton, 1959]. In their work, the authors consider m processors, which are exactly alike, and there are $J = \{1, \ldots, n\}$ tasks that need to be completed. Each task j has a processing time c_j. The tasks can be split in any number of ways, but two processors cannot work on the same task at the same time. The following theorem, which is a specific case of the result in [McNaughton, 1959], guarantees the existence of a schedule.

Theorem 2.2.1. *If $\sum_{j \in J} c_j \leq m$, then a necessary and sufficient condition that there exists a schedule in which all tasks are completed by time 1 is that, for all $j \in J$, $c_j \leq 1$.*

In our setting, the m processors correspond to the security resources, the tasks, that need to be completed, correspond to the targets, that need to be covered, and the processing time of a task corresponds to the coverage probability on a target. In the constructive proof of Theorem 2.2.1, the authors construct a schedule where the following condition is fulfilled: the schedule of every processor, with the possible exception of one, is either entirely filled up or is entirely empty.

The schedule is constructed as follows. One stacks up the optimal c_j values consecutively inside m columns, which represent the processors/resources, of height one, from left to right. Whenever a column is topped up, one can either start filling up the next column with the remaining quantity of unassigned c_j from the previous column, or continue with c_{j+1}. Further, upon inspection of the resulting diagram, one can determine, before a processor switches task, all tasks that are currently being processed and the time taken before a given configuration of tasks being processed changes. In our setting these configurations correspond to the pure strategies–coverage of m targets–and the time taken between configuration changes, corresponds to the probability with which the coverage configuration prior to the change occurs. Algorithm 1, summarizes the steps required to recover the defender's implementable mixed strategy.

Note that the validity of the approach is a direct consequence of Theorem 2.2.1 and that by construction, x satisfies Constraints (2.2.3), (2.2.4) and (2.2.5).

```
1 {Start with $m$ columns of height 1};
2 Step 1. Stack up optimal $c_j$ variables inside the columns;
3 Step 2. Make transversal cuts in the box every time there is a configuration change
  along the columns and a final cut at height one;
4 Step 3. Read off the mixed strategy as follows:
      a. Starting from the bottom of the box, the pure strategies correspond to the different
      configurations of resources covering targets separated by the transversal cuts.
      b. Each configuration's probability is given by the height of the corresponding
      configuration.
```

Algorithm 1: Box method

The following example illustrates the box method procedure to recover an implementable mixed strategy. Consider a security instance with $J = 4$, $K = 1$ and $m = 3$. Let the optimal coverage probabilities be given by $c = (0.7, 0.7, 0.65, 0.95)$. Stacking the values of the c_j's up into $m = 3$ columns and performing the transversal cuts as per Steps 1 and 2 in the box method produces the drawing shown in Figure 2.2.1. Further, making the transversal cuts

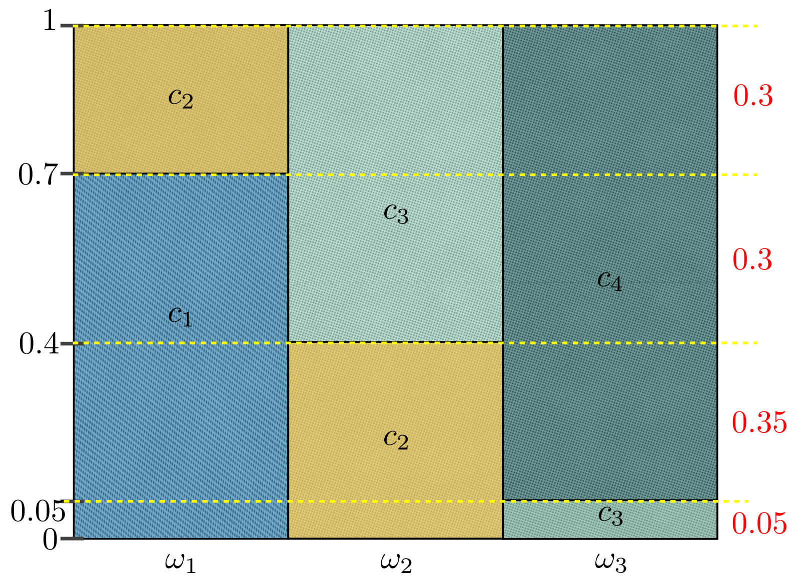


Figure 2.2.1: Box method in use

and reading off the pure strategies produces the results shown in Table 2.2.2.

The weights in the decomposition indicate the proportion of the time that the corresponding pure strategy should be implemented to adhere to the optimal mixed strategy recovered. According to the mixed strategy, strategy 1, which consists in protecting targets

Defender pure strategy	Targets protected	Weight in mixed strategy
1	(1,2,3)	0.05
2	(1,2,4)	0.35
3	(1,3,4)	0.3
4	(2,3,4)	0.3

Table 2.2.2: Defender mixed strategy

1,2 and 3, should be applied 5% of the times, strategy 2, which provides protection to targets 1,2 and 4, should be implemented 35% of the times, and so on. A defender wanting to decide how to deploy his $m = 3$ resources adhering to the recovered mixed strategy on a given turn simply has to generate a random number in $[0, 1]$ and based on where the number falls in the line between 0 and 1, commit to the corresponding strategy. In the example shown in Figure 2.2.2, the random number generated is 0.43 which indicates that strategy 3 should be played, *i.e.*, security resources should be deployed to protect targets 1, 3 and 4.

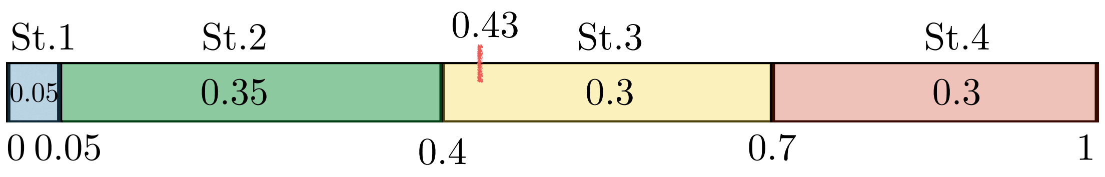


Figure 2.2.2: Choosing a pure strategy in compliance with the mixed strategy

Chapter 3

State of the art

In this chapter we review the state of the art of general and security Stackelberg games. In order to be exhaustive, we first classify different Stackelberg problems studied in the literature according to their computational complexity and discuss where the major computational challenges arise when solving the hard problems. We use the classification to describe the state of the art solution methods used in each case. Finally, we conclude the chapter with a discussion of relevant extensions of the Stackelberg games studied in this work, such as behavioral Stackelberg games and repeated Stackelberg games and make connections to Interdiction games and Patrolling games among others.

3.1 Computational complexity

Much work has been done on determining the complexity of both general and security games under different restrictions and on variations of the original games. The main papers that have contributed to studying the complexity of these games are [Conitzer and Sandholm, 2006, Korzhyk et al., 2010, Conitzer and Korzhyk, 2011, Jain et al., 2012].

The authors in [Conitzer and Sandholm, 2006] show that the GSG, when restricted to a single follower type, is polynomially solvable by providing a solution method called the multiple LP approach.

Theorem 3.1.1. *In a GSG with a single follower type, an optimal mixed strategy for the leader can be found in polynomial time.*

Proof. According to Remark 1.1.1, the follower best responds to any leader strategy by adopting a pure strategy, $q \in \mathbb{B}$, where $\mathbb{B} = \{e^1, e^2, \ldots, e^{|J|}\}$ be the canonical basis in $\mathbb{R}^{|J|}$. Then, for each follower strategy $q \in \mathbb{B}$, an optimal mixed strategy for the leader, $x^* \in \mathbb{S}^{|I|}$, under the constraint that q is indeed a follower best response, can be calculated in polynomial time by solving the following linear program:

$$
\begin{aligned}
\text{(Mult-LP)} \quad \text{Max} \quad & \sum_{i \in I} \sum_{j \in J} R_{ij} x_i q_j \\
\text{s.t.} \quad & \sum_{i \in I} x_i = 1, && (3.1.1)\\
& x_i \geq 0 \qquad \forall i \in I, && (3.1.2)\\
& \sum_{i \in I} \sum_{j \in J} C_{ij} x_i q_j \geq \sum_{i \in I} \sum_{j \in J} C_{ij} x_i \tilde{q}_j \qquad \forall \tilde{q} \in \mathbb{B}. && (3.1.3)
\end{aligned}
$$

The above linear program is solved for each follower's pure strategy $q \in \mathbb{B}$ and the optimal mixed strategy for the leader is selected as the mixed strategy that yields the highest profit among all the LP solutions. This process is illustrated in Figure 3.1.1. ■

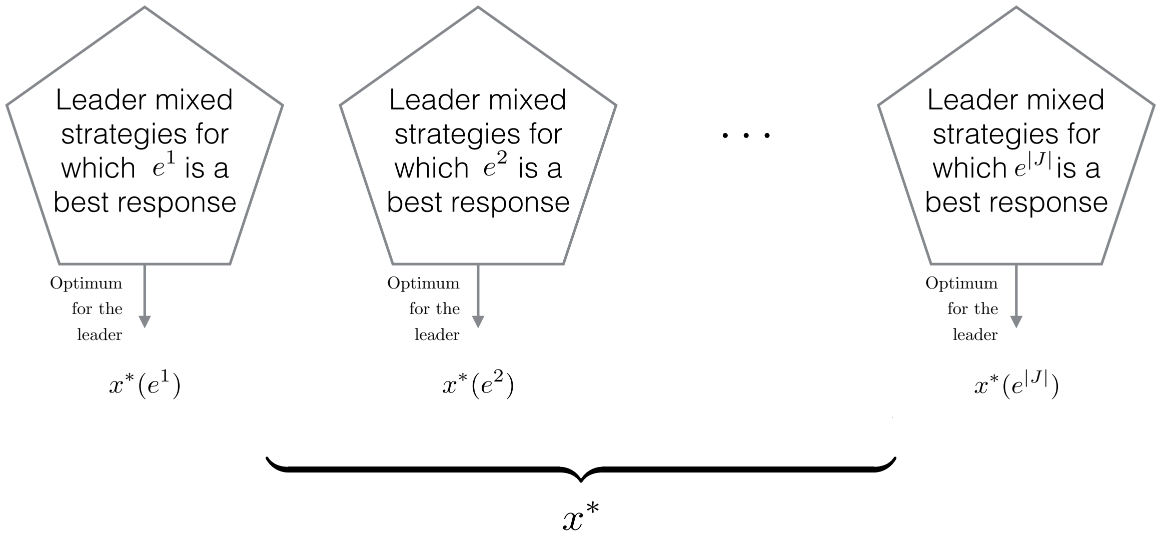

Leader mixed strategies for which e^1 is a best response
Optimum for the leader
$x^*(e^1)$
Leader mixed strategies for which e^2 is a best response
Optimum for the leader
$x^*(e^2)$
. . .
Leader mixed strategies for which $e^{|J|}$ is a best response
Optimum for the leader
$x^*(e^{|J|})$
x^*

Figure 3.1.1: Multiple LPs approach returns x^*, a reward-maximizing leader mixed strategy

The p-follower GSG was shown by [Conitzer and Sandholm, 2006] to be $\mathcal{NP}$-hard. The proof involves a polynomial reduction to the GSG from the Stable Set problem, which is a well known $\mathcal{NP}$-hard problem, [Karp, 1972]. When one looks at the specific case of Stackelberg security games, however, the classification is not as immediate.

When one restricts an SSG to a single type of attacker, other factors must be taken into account before the complexity of the game can be ascertained. These factors relate to the capabilities of the security resources considered and to the precise definition of how the security game is to be played. In an SSG, the defender has a limited number of security resources that can be deployed to protect targets. If the game is such that any resource can provide protection to any target, resources are said to be homogeneous. If, however, resources have restrictions as to what targets they can interact with, resources are then said to be heterogeneous. For instance, consider a local police force allocating a fleet of indistinguishable patrol cars to different locations in a downtown area–since any car can be assigned to any location, these resources are homogeneous. On the other hand, consider a coast guard agency allocating ground units and maritime units to protect infrastructure located along the coast and at sea–since ground units cannot protect sea targets and maritime units cannot access ground locations, resources are heterogeneous. Further, a distinction must be made between SSG games where security resources cover individual targets as in [Kiekintveld et al., 2009] or if resources are allowed to cover groups of targets simultaneously as is the case in [Jain et al., 2010b], where US Air Marshals are assigned to tours of flights and every flight on the tour is considered protected if a Marshal is assigned to the tour. In the latter case, a resource is assigned to a so-called schedule and the size of the schedule is determined by the number of targets protected by the resource.

The authors in [Korzhyk et al., 2010] provide a complete study of the complexity of the single attacker type SSG under the different variations discussed. Their results are summarized in Table 3.1.1.

	Schedules		
	Size 1	Size ≤ 2	Size ≥ 3
Homogeneous resources	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{NP}$-hard
Heterogeneous resources	$\mathcal{P}$	$\mathcal{NP}$-hard	$\mathcal{NP}$-hard

Table 3.1.1: Complexity results for single type of attacker SSG due to [Korzhyk et al., 2010]

The complexity of the p-attacker SSG with homogeneous resources and singleton schedules–which is the variation of the problem studied in this thesis–although most probably $\mathcal{NP}$-hard, to the best of our knowledge remains open as no formal proof has been given. A possible proof could be obtained by adapting the $\mathcal{NP}$-hardness proof of the p-follower GSG in [Conitzer and Sandholm, 2006] to the security case.

In this thesis we concentrate on p-follower GSGs and p-attacker SSGs with homogeneous resources and singleton schedules and their respective restrictions to a single type of follower and attacker.

3.2 Computational challenges

The authors in [Jain et al., 2012] study p-follower SSGs and how to determine the problems which are computationally hardest to solve, independent of the methodology used. To understand the algorithm-independent structural properties of SSGs they rely on the concept of phase transition introduced for decision problems in [Cheeseman et al., 1991]. Phase transitions, which are known to correlate very strongly with computational hardness, have been used to analyze the computational impact of problem structure in several optimization problems such as MAX-SAT [Slaney and Walsh, 2002] or TSP [Gent and Walsh, 1996]. In their work, [Jain et al., 2012] identify a phase transition in a decision version of a SSG. They further relate the phase transition to the concept of *deployment-to-saturation ratio* (*d:s*), a domain-spanning measure of the density of the defender coverage in an SSG. According to their work, the computationally hardest random instances of such games occur at a *d:s* ratio of 0.5, and they show that this 'hard' region corresponds to a phase transition in the probability that a corresponding decision version of the SSG has a solution.

The authors in [Jain et al., 2012] make the point that it is important that algorithms' performances be tested on the hardest problems possible: those where optimization has more of an impact and they further provide evidence that a naive defender strategy does almost as well as an optimal defender strategy where the *d:s* ratio is either small or large. This remark has been taken into consideration when determining the computational experiments presented throughout this work.

It is because the problems studied are not, for the most part, easy to solve, that much attention has been paid to tackling scaling up in the size of the instances. Scale-up challenges arise in these games for at least two reasons:

1. *Growth in the number of player strategies.* We have already discussed that when contemplating a security game, the leader has exponentially many strategies that he can commit to. In some settings, one may also face an exponential growth in the follower's pure strategies.

2. *Growth in the number of follower/attacker types.* When there is an increase in either the number of follower types in a GSG or in the number of attacker types in an SSG, the corresponding problems become much harder to tackle.

In the next section, we discuss some of the most relevant approaches to solving GSG and SSG games and we further discuss how some of the complications addressed in this section have been tackled in the literature.

3.3 Methods

When it comes to the easy problems, those that are polynomially solvable, the multiple LP approach proposed in [Conitzer and Sandholm, 2006] is the most widely used approach. The multiple LP approach is naturally adapted to the security case when applied to SSGs as shown in [Korzhyk et al., 2010] to avoid the exponentially many leader variables, one for each defender pure strategy.

For single follower type GSGs, the Multiple LP method is improved upon in [Conitzer and Korzhyk, 2011], where the LPs are merged into a single MILP formulation that is then shown to be an ideal formulation, *i.e.*, its constraints are a linear description of the convex hull of feasible solutions. Further details of this formulation will be presented in Chapter 4.

As for the hard problems, let us first discuss GSGs. A first approach to solve a p-follower GSG involves the *Harsanyi transformation* [Harsanyi and Selten, 1972]. This technique reduces a p-follower case to a single follower case involving a 'mega'-follower with an exponential number of follower pure strategies. Each pure strategy for the 'mega'-follower lies in the cross product of all the sets of strategies of the original p followers. A clarifying example is presented in Figure 3.3.1 where there are two follower types and where follower 1 is active with probability $\pi^1 \in [0,1]$ and follower 2 is active with probability $\pi^2 \in [0,1]$. The transformed normal-form game is also shown in Figure 3.3.1. Note that the leader still has two pure strategies, but the strategy space for the 'mega'-follower has grown to four (2^2). In general, for $|K|$ follower types and $|J|$ strategies per follower type, the transformation results in a game with $|J|^{|K|}$ strategies for the follower.

π^1	Follower type 1	
	a	b
1	(1,1)	(3,0)
2	(2,0)	(1,0)

π^2	Follower type 2	
	a′	b′
1	(2,1)	(1,3)
2	(3,0)	(3,4)

	aa′	ab′	ba′	bb′
1	$(1\pi^1 + 2\pi^2, 1\pi^1 + 1\pi^2)$	$(1\pi^1 + 1\pi^2, 1\pi^1 + 3\pi^2)$	$(3\pi^1 + 2\pi^2, \pi^2)$	$(3\pi^1 + \pi^2, 3\pi^2)$
2	$(2\pi^1 + 3\pi^2, 0)$	$(2\pi^1 + 3\pi^2, 4\pi^2)$	$(1\pi^1 + 3\pi^2, 0)$	$(1\pi^1 + 3\pi^2, 4\pi^2)$

Figure 3.3.1: Constructing the Harsanyi payoff matrix for a 2-follower GSG

The resulting single follower type GSG can be solved using the multiple LP approach which now involves solving an exponential number of linear programs. Clever algorithms have been designed to avoid having to solve all the linear programs. The most noteworthy contributions are due to [Jain et al., 2011a] and [Yin and Tambe, 2012]. They propose algorithms based on the following idea. One can think of the p-follower GSG as a search tree where the root node corresponds to the original problem and as one progresses down the tree, best responses for the different follower types are fixed at the different levels. Then, the leafs of this search tree correspond to the exponentially many LPs one solves when applying the multiple LP approach. For instance, consider the example in Figure 3.3.2.

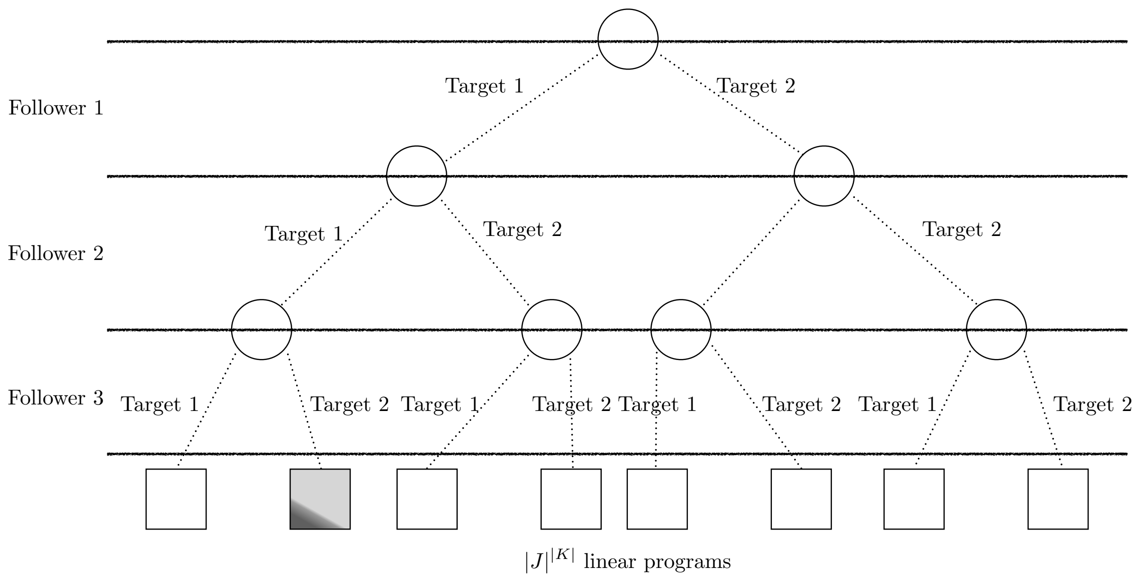


Figure 3.3.2: Search tree for a p-follower GSG with 3 follower types and 2 actions for each follower type

There are three follower types and the pure strategies for each follower type consist in either selecting target 1 or target 2. The leafs are represented by squares and the internal nodes by circles. The gray square corresponds to the linear program that computes the optimal mixed strategy for the leader under the assumption that follower 1 best responds by selecting target 1, follower 2 best responds by selecting target 1 and follower 3 best responds by selecting target 2. The algorithms proposed by [Jain et al., 2011a] and [Yin and Tambe, 2012] use MILP formulations and/or heuristic approaches, that respect the assignments at the internal nodes, to obtain tight upper bounds on the optimal solution that allow for an efficient pruning of the tree.

Naturally, several stand-alone MILP formulations have been proposed to tackle p-follower GSGs, to various degrees of success. Notable contributions are due to [Paruchuri et al., 2008] and [Conitzer and Korzhyk, 2011]. We postpone the discussion of the MILP formulations to the next chapter, where they will be analyzed in detail.

With respect to the SSGs, as mentioned in the previous chapter, work by [Kiekintveld et al., 2009] has been crucial in determining a compact representation for security games. This has allowed to extend the multiple LP approach efficiently to single type of attacker SSGs [Korzhyk et al., 2010], and to extend the algorithms of [Jain et al., 2011a] and [Yin and Tambe, 2012] to p-follower SSGs.

Compact MILP formulations for SSGs are due to [Kiekintveld et al., 2009], for the case of homogeneous resources and singleton schedules, and to [Jain et al., 2010b], for the case of heterogeneous resources and non-singleton schedules (as used in the problem of assigning Air Marshals to tours of flights). We postpone a detailed discussion of the MILP formulations to the next chapter.

To address scaling up capabilities of the formulations and to avoid an explicit enumeration of the defender strategies, Branch and Price approaches based on Column Generation [Barnhart et al., 1998] have exploited problem structure in security games [Jain et al., 2010a]. In their work, where they assign Air Marshals to flights, they build on the insight that in many real-world security problems, there exist solutions with small support sizes, *i.e*, the pure strategies that are played with positive probability constitute a small subset of all the available pure strategies. In their proposed approach, defender pure strategies are thus iteratively generated and added to the optimization formulation.

Similarly, Benders decomposition and other constraint generation techniques have been used to improve the solving capabilities of formulations by iteratively adding constraints on the fly. Benders decomposition is key in the algorithm presented in [Yin and Tambe, 2012] to efficiently solve the MILP formulation that provides the tight upper bounds on the optimal solution at the internal nodes in the search tree, and a cutting plane algorithm is presented in [Yang et al., 2013].

Further, constraint generation is used in [Haskell et al., 2014] where a complex patrolling problem is modeled for the Unites States Coast Guard in the context of protecting fisheries against a very large number of follower types. Many complex spatio-temporal constraints are disregarded and then added when they become violated. The separation problem that determines whether or not the current solution violates these constraints is solved by column generation, where only a small set of feasible patrols are considered.

Additionally, Network Pricing Problems with connected toll arcs, although not directly related to Stackelberg games as defined in this work, can be modelled through Bilevel Programming and as such can be seen as a special type of Stackelberg game between a highway authority, that plays the role of the leader and has to decide on pricing a subset of connected arcs in a network and the users, which are the followers and select a series of paths from some origins to some destinations such that the total travel cost is minimized. In this particular context, a Dantzig-Wolfe reformulation [Dantzig and Wolfe, 1960] is proposed in [Fortz et al., 2013] which is then solved by column generation.

Branch and Price and Cut approaches have also recently been used in the context of Network Pricing Problems [Violin, 2014, Morais et al., 2016]. The approach in [Morais et al., 2016] is developed on a reformulation of a bilevel formulation which is then strengthened by applying Reformulation Linearization Technique (RLT), [Sherali and Adams, 1994]. The approach in [Violin, 2014] extends upon the work in [Fortz et al., 2013] by exploring branching strategies to construct a Branch and Price framework and by further adding valid inequalities to the problem.

3.4 Noteworthy extensions

Even though the scope of this thesis is limited to p-follower GSGs and p-attacker SSGs with singleton schedules and homogeneous targets, it is interesting to note that many variations of these games have been studied, most motivated by real-life problems.

Interdiction games are a type of Stackelberg game played on a network where the leader and the follower have opposite objective functions. In these games, the follower has to maximize flow over the network from an origin to a destination and the leader, who has a limited interdiction budget can interdict or block edges in the network so as to minimize the flow. Interdiction games are motivated by military or homeland security applications. Early research is due to [McMasters and Mustin, 1970] and [Ghare et al., 1971]. Flow disruptions in drug trafficking networks have been considered by [Wood, 1993]. Comprehensive reviews of interdiction games are due to [Israeli, 1999] and [DeNegre, 2011]. More recent work is due to [Fischetti et al., 2016a].

Security related game theoretic problems on networks also include patrolling games, [Alpern et al., 2011, Papadaki et al., 2016]. In these games, a mobile defender patrols nodes along the edges of a graph during a certain time period. A strategic attacker selects a node and a time window in which to attack said node. Attacking a node requires for the attacker to be present at the node for some time m. If the defender passes by the attacker's selected node while the attacker is perpetrating his crime, the attack is prevented. Unlike the games we study in this thesis, patrolling games are zero-sum with the defender and the attacker wanting to maximize and minimize the interception probability of an attack respectively. Work by [Alpern et al., 2011] provides valuable insights into good network topologies, such as a hamiltonian path, where defender patrolling is shown to have a high interception rate. Work by [Papadaki et al., 2016] tackles the case of defining optimal patrolling strategies on a line graph. Research challenges in patrolling games involve tackling general graphs with a tree-like structure, allowing for multiple patrollers and/or attackers and considering the case where the distances between nodes, as well as their value, are not the same.

Stackelberg methodology is also used when the potential targets are the edges in a graph instead of its nodes. Work by [Hochbaum et al., 2014] models a game where the attacks are nuclear threats on the edges of a network. The targets in this problem are not independent, as is the case in many other SSG applications, since a patrol on the network traverses the edges in a given order. The complexities involved in solving their problem leads the authors to construct the patrolling strategy in two phases. First, targets are assumed independent and optimal coverage probabilities on the targets are computed. Then, an approximation algorithm, based on the NP-hard k-vehicle rural Chinese Postman Problem, returns maximal reward patrolling routes that visit the edges sampled according the optimal coverage probabilities returned by the SSG.

Another prolific research domain is that of the so-called urban network security setting. In these network Stackelberg games, the strategies of the defender consist in allocating resources to the edges of the network (checkpoints on a road network) and the follower strategies consist in selecting paths along the network from any source node to any target node. Work by [Jain et al., 2011b] explores an application in Mumbai to set checkpoints so as to prevent terrorists from reaching certain infrastructure targets on a graph.

Stackelberg games on networks have also been used to address fare evasion on public transportation systems [Yin et al., 2012, Correa et al., 2014]. In these games the defender schedules metro conductors to perform patrols along the metro system to check for fare evaders. The followers observe the patrols over time and decide whether or not to pay for their tickets. Patrols are adjusted to both the spatial and temporal constraints of travel within the transit network where dynamic uncertainty in the execution of the patrols is modeled through Markov Decision Processes [Howard, 1960]. This allows to update a patrol in real time in case of unforeseen interruptions. Further, [Shieh et al., 2012] also work on a network to determine patrols for the US Coast Guard in a port scenario.

The game theoretic solution concept of SSE assumes perfect rationality on the part of the follower/attacker but human adversaries, however, may have cognitive limitations and biases that require the use of different algorithms to exploit this. Human decision making has been studied in the field known as Behavioral game theory [Camerer, 2011]. It can be argued that human players rarely adopt purely rational strategies, and thus, building an optimal defender mixed strategy based precisely on the rationality of the attacker, hardly seems the best choice. Work by [Pita et al., 2009] focuses on developing methods that are robust against human decision making and that exploit facing a human adversary. Work by [Shieh et al., 2012] include a so-called quantal response model to account for human bounded rationality.

Another assumption that is generally made is that when the defender commits to a mixed strategy, the attacker conducts surveillance to learn the mixed strategy that the defender has committed to and then best responds to said strategy. In some situations, the attacker may have a restricted capability of learning the mixed strategy or it may be unclear to the defender how well the attacker can conduct surveillance. This raises the valid question of how to best compute the strategy for the defender if he cannot fully anticipate the attacker's best response to his strategy. The authors in [Yin et al., 2010] provide some insight into this extension.

Further, most of the games explored in the literature are static ; The first player commits to a strategy, the adversary best responds and the game is over. In real life, this may not be the case. The authors in [Marecki et al., 2012] provide a Monte Carlo tree search approach that identifies non-dominated optimal strategies for a leader in a p-follower GSGs where the objective is to maximize the leader's cumulative expected payoff over the different rounds of the game. The actions selected through the different rounds have to be strategic and aim for higher profit in the long run as opposed to a high immediate payoff. More recent work, by [Kar et al., 2016], further explores repeated games and introduces bounded rationality in the follower types. Further, they extend the basic repeated model to allow for adaptive follower types; followers that learn from previous successes or failures and play accordingly.

Chapter 4

A study of general and security Stackelberg game formulations

4.1 Introduction

In this chapter we study known MILP formulations for GSGs from the literature and prove an inclusion chain of the polyhedra of their LP relaxations, projected onto the appropriate space of variables. This leads to an ordering of the GSG formulations with respect to the quality of the upper bound provided by their corresponding LP relaxations.

We further provide a theoretical link between the general Stackelberg setting and the security Stackelberg setting involving projections of the polyhedra of the LP relaxations of the GSG formulations to derive valid SSG formulations, of which a new SSG formulation, (MIP-p-S$_{q,y}$), is a notable contribution.

Exploiting the link between GSGs and SSGs allows to extend the study of GSG formulations to SSG formulations, leading to an inclusion chain of the polyhedra of the LP relaxations of the SSG formulations, projected onto the appropriate space of variables. We further obtain an ordering of the SSG formulations with respect to the quality of the upper bound provided by their corresponding LP relaxations. The LP relaxation of (MIP-p-S$_{q,y}$) accounts for the tightest polyhedron and, therefore, for the best LP bound. We further prove that its single type of attacker restriction, (MIP-1-S$_{q,y}$), is an ideal formulation.

Computational studies show that the tightest formulations in each setting, are highly competitive with respect to solution time, particularly for larger instances. In particular, (MIP-p-S$_{q,y}$) scales significantly better than competing formulations over the instances tested. The largest general instances solved in our experiments correspond to games with 6 30×30 payoff bimatrices. In the security setting, the largest games considered involve 70 targets, 12 different types of attackers and 35 security resources. General and security games of this size can easily be tackled by the tightest formulation in each setting within a 30 minute time limit.

This chapter is organized as follows: In Section 4.2, we introduce GSG MILP formulations from the literature and compare them with respect to the strength of their linear relaxations. We further conduct computational experiments to analyze the behavior of the GSG formulations. In Section 4.3, we deduce SSG MILP formulations using projections, in the appropriate space of variables, of the GSG formulations presented in Section 4.2. Notably, we introduce (MIP-p-S$_{q,y}$), a new SSG MILP formulation. We further extend the comparison between GSG formulations to the security setting. We then perform computational experiments to analyze the behavior of the SSG formulations. We conclude this chapter with some closing remarks in Section 4.4.

4.2 General Stackelberg games–GSGs

In Section 4.2.1, we present equivalent MILP formulations for the p-follower GSG. In Section 4.2.2 we compare the polyhedra of the LP relaxations for the different formulations.

4.2.1 Single level MILP formulations

The authors in [Paruchuri et al., 2008] tackle the problem of solving the bilevel formulation presented earlier, (BIL-p-G$_{x,q}$) by using a MILP reformulation. They replace the second level nested optimization problem, described by

$$\begin{aligned}
&q^k \in \arg\max_{r^k} \left\{ \sum_{i\in I}\sum_{j\in J} C^k_{ij} x_i r^k_j \right\} && \forall k \in K,\\
&r^k_j \in \{0,1\} && \forall j \in J, \forall k \in K,\\
&\sum_{j\in J} r^k_j = 1 && \forall k \in K,
\end{aligned}$$

by the following set of constraints:

$$\sum_{j\in J} q^k_j = 1 \qquad \forall k \in K, \tag{4.2.1}$$

$$q^k_j \in \{0,1\} \qquad \forall j \in J, \forall k \in K, \tag{4.2.2}$$

$$0 \le (s^k - \sum_{i\in I} C^k_{ij} x_i) \le (1 - q^k_j) \cdot M^2 \qquad \forall j \in J, \forall k \in K, \tag{4.2.3}$$

where $s^k \in \mathbb{R}$ for all $k \in K$ and M^2 is a big-M constant. The two inequalities in Constraint (4.2.3) ensure that $q^k_j = 1$ only for a pure strategy that maximizes the follower's payoff. The following is a quadratic (bilinear) single level problem for the GSG where the only quadratic term appears in the objective function:

$$
\begin{array}{llll}
(\mathrm{QUAD}_{x,q,s}) & \text{Max} & \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} x_i q^k_j & \\
& \text{s.t.} & (4.2.1) - (4.2.3) & \\
& & \sum_{i\in I} x_i = 1 & (4.2.4)\\
& & x_i \geq 0 \qquad \forall i \in I. & (4.2.5)
\end{array}
$$

Formulation $(\mathrm{D2}_{x,q,s,f})$ due to [Paruchuri et al., 2008], avoids the quadratic term in the objective of $(\mathrm{QUAD}_{x,q,s})$ by adding $|K|$ new variables and introducing a second family of constraints involving a big-M constant.

$$
\begin{array}{lllll}
(\mathrm{D2}_{x,q,s,f}) & \text{Max} & \sum_{k\in K}\pi^k f^k & & (4.2.6)\\
& \text{s.t.} & (4.2.1) - (4.2.5), & & \\
& & f^k \leq \sum_{i\in I} R^k_{ij} x_i + (1 - q^k_j)\cdot M^1 & \forall j \in J, \forall k \in K, & (4.2.7)\\
& & s, f \in \mathbb{R}^{|K|} & \forall k \in K. &
\end{array}
$$

Alternatively, one can eliminate the nonlinearity in the objective function [Paruchuri et al., 2008], by adding additional variables that represent the product between x and q. To be more precise, consider $z^k_{ij} = x_i q^k_j$ for all $i \in I$, $j \in J$ and $k \in K$. This gives rise to the following formulation called $(\mathrm{DOBSS}_{q,z,s})$:

$$
\begin{array}{lllll}
(\mathrm{DOBSS}_{q,z,s}) & \text{Max} & \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} z^k_{ij} & & \\
& \text{s.t.} & (4.2.1), (4.2.2), & & \\
& & \sum_{j\in J} z^k_{ij} = \sum_{j\in J} z^1_{ij} & \forall i \in I, \forall k \in K, & (4.2.8)\\
& & \sum_{i\in I} z^k_{ij} = q^k_j & \forall j \in J, \forall k \in K, & (4.2.9)\\
& & z^k_{ij} \geq 0 & \forall i \in I, \forall j \in J, \forall k \in K, & (4.2.10)\\
& & 0 \leq s^k - \sum_{i\in I}\sum_{j'\in J} C^k_{ij} z^k_{ij'} & & \\
& & \qquad \leq (1 - q^k_j)\cdot M^2 & \forall j \in J, \forall k \in K, & (4.2.11)\\
& & s \in \mathbb{R}^{|K|}. & &
\end{array}
$$

Additionally, the real variables s^k in Constraints (4.2.3) and (4.2.11) can be projected out by using Fourier-Motzkin elimination [Dantzig and Eaves, 1973]. This gives rise to constraints:

$$\sum_{i\in I}(C^k_{ij} - C^k_{i\ell})x_i \leq (1-q^k_\ell)\cdot M^2 \qquad \forall j,\ell \in J, \forall k \in K, \tag{4.2.12}$$

$$\sum_{i\in I}\sum_{j'\in J}(C^k_{ij} - C^k_{i\ell})z^k_{ij'} \leq (1-q^k_\ell)\cdot M^2 \qquad \forall j,\ell \in J, \forall k \in K. \tag{4.2.13}$$

Replacing (4.2.3) by (4.2.12) in (D2$_{x,q,s,f}$) and (4.2.11) by (4.2.13) in (DOBSS$_{q,z,s}$) yields (D2$_{x,q,f}$) and (DOBSS$_{q,z}$). We later analyze the behavior of these last two formulations compared to that of (D2$_{x,q,s,f}$) and (DOBSS$_{q,z,s}$) to see if removing variables s at the expense of adding constraints is worthwile.

Another equivalent MILP formulation for the p-follower GSG can be obtained by replacing Constraint (4.2.11) with the following constraint:

$$\sum_{i\in I}(C^k_{ij} - C^k_{i\ell})z^k_{ij} \geq 0 \qquad \forall j,\ell \in J, \forall k \in K. \tag{4.2.14}$$

This constraint is derived by multiplying Constraint (4.2.12) by q^k_ℓ, reorganizing and replacing the nonlinear terms $x_i q^k_j$ by z^k_{ij}. This leads to (MIP-p-G$_{q,z}$):

$$\begin{array}{lll} (\text{MIP-}p\text{-G}_{q,z}) & \text{Max} & \displaystyle\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} z^k_{ij} \\ & \text{s.t.} & (4.2.1), (4.2.2), (4.2.8)-(4.2.10), \\ & & \displaystyle\sum_{i\in I}(C^k_{ij} - C^k_{i\ell})z^k_{ij} \geq 0 \qquad \forall j,\ell \in J, \forall k \in K. \qquad (4.2.15) \end{array}$$

The linear relaxation of (MIP-p-G$_{q,z}$) appears in [Yin and Tambe, 2012]. The MILP formulation is a p-follower extension to the single follower formulation (MIP-1-G$_{q,z}$), due to [Conitzer and Korzhyk, 2011]. Formal proofs that the formulations seen thus far are equivalent MILP formulations, *i.e.*, that they are valid for the p-follower GSG, appear in [Paruchuri et al., 2008], for (DOBSS$_{q,z,s}$) and [Paruchuri et al., 2008] and [Kiekintveld et al., 2009] for (D2$_{x,q,s,f}$). These proofs show that each of them is equivalent to (QUAD$_{x,q,s}$). The equivalence of (DOBSS$_{z,q}$) and (D2$_{x,q,f}$) is obtained from the Fourier-Motzkin elimination procedure [Dantzig and Eaves, 1973]. The equivalence proof for (MIP-p-G$_{q,z}$) is analogous to the proof used to show the equivalence for (DOBSS$_{q,z,s}$) and omitted here. For completeness, the proof is included in the appendix (See Appendix A, Section A.1).

[Paruchuri et al., 2008] state that the big M constants used are arbitrarily large. To be as computationally competitive as possible, we provide the tightest value for each big-M constant in the formulations discussed thus far.

Proposition 4.2.1. *The tightest values for the positive constants M are:*

1. *In (4.2.7),* $M1^{k^*}_j = \max_{i\in I}\{\max_{\ell\in J} R^k_{i\ell} - R^k_{ij}\}\ \forall j \in J, \forall k \in K.$

2. In (4.2.3) and (4.2.11), $M2^{k^*}_{j} = \max_{i \in I}\{\max_{\ell \in J} C^k_{i\ell} - C^k_{ij}\}\ \forall j \in J, \forall k \in K$.

3. In (4.2.12) and (4.2.13), $M2^{k^*}_{\ell j} = \max_{i \in I}\{C^k_{ij} - C^k_{i\ell}\}, \forall j, \ell \in J, \forall k \in K$.

For a detailed proof of this proposition, please refer to Appendix A, Section A.2.1.

4.2.2 Comparison of the formulations

Given a linear formulation F, we denote by $\overline{\mathrm{F}}$ its linear (continuous) relaxation and by $\mathcal{P}(\overline{\mathrm{F}})$ the polyhedral feasible region of $\overline{\mathrm{F}}$. Further, let $Q = \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^m : Ax + Bz \le d\}$. Then the projection of Q into the x-space, denoted $Proj_x Q$, is the polyhedron given by $Proj_x Q = \{x \in \mathbb{R}^n : \exists z \in \mathbb{R}^m$ for which $(x, z) \in Q\}$, see [Wolsey, 1998].
First, we introduce an additional formulation which we denote by $(\mathrm{DOBSS}_{x,q,z,s,f})$. This formulation is equivalent to $(\mathrm{DOBBS}_{q,z,s})$, in the sense that the optimal values of their LP relaxations coincide. In this formulation we introduce variables f^k for all $k \in K$ to rewrite the objective function so that it matches (4.2.6). We also add variables x_i for all $i \in I$ by rewriting (4.2.8) as $\sum_{j \in J} z^k_{ij} = x_i$ for all $i \in I$ and all $k \in K$. Using this last condition, one can simplify (4.2.11) to (4.2.3). The formulation $(\mathrm{DOBSS}_{x,q,z,s,f})$ is as follows.

$$
\begin{array}{llll}
(\mathrm{DOBSS}_{x,q,z,s,f}) & \mathrm{Max} & \sum_{k \in K} \pi^k f^k & \\
& \text{s.t.} & (4.2.1) - (4.2.3), & \\
& & \sum_{i \in I} z^k_{ij} = q^k_j & \forall j \in J, \forall k \in K, \quad (4.2.16) \\
& & z^k_{ij} \ge 0 & \forall i \in I, \forall j \in J, \forall k \in K, \quad (4.2.17) \\
& & f^k = \sum_{i \in I}\sum_{j \in J} R^k_{ij} z^k_{ij} & \forall k \in K, \quad (4.2.18) \\
& & \sum_{j \in J} z^k_{ij} = x_i & \forall i \in I, \forall k \in K, \quad (4.2.19) \\
& & s \in \mathbb{R}^{|K|}. &
\end{array}
$$

Further, note that from the Fourier Motzkin elimination procedure one has that

$$\mathcal{P}(\overline{\mathrm{D2}_{x,q,f}}) = Proj_{x,q,f}\mathcal{P}(\overline{\mathrm{D2}_{x,q,s,f}}) \text{ and,}$$

$$\mathcal{P}(\overline{\mathrm{DOBSS}_{q,z}}) = Proj_{q,z}\mathcal{P}(\overline{\mathrm{DOBSS}_{q,z,s}}).$$

Proposition 4.2.2. *$Proj_{x,q,s,f}\mathcal{P}(\overline{DOBSS_{x,q,z,s,f}}) \subseteq \mathcal{P}(\overline{D2_{x,q,s,f}})$. Further, there exist instances for which the inclusion is strict.*

Proof. Note that all the constraints of $\mathcal{P}(\overline{\mathrm{D2}_{x,q,s,f}})$ can be found in the description of $\mathcal{P}(\overline{\mathrm{DOBSS}_{x,q,z,s,f}})$ except for Constraints (4.2.4), (4.2.5) and (4.2.7). Constraint (4.2.4) is implied by Constraints (4.2.1), (4.2.16) and (4.2.19). Constraint (4.2.5) is implied by Constraints (4.2.17) and (4.2.19) .

Further, the projection of $\mathcal{P}(\overline{\text{DOBSS}_{x,q,z,s,f}})$ on the (x,q,s,f)-space can be obtained by applying Farkas' Lemma [Farkas, 1902]. Constraints (4.2.16), (4.2.17), (4.2.18) and (4.2.19) are the only ones involving variables z_{ij}^k and are separable by $k \in K$. For a fixed $k \in K$ the projection is given by:

$$A^k = \{(x,q,f) : \alpha f^k + \sum_{i \in I} \beta_i x_i + \sum_{j \in J} \gamma_j q_j^k \geq 0 \ \forall (\alpha,\gamma,\beta) :$$

$$\alpha R_{ij}^k + \beta_i + \gamma_j \geq 0 \ \forall i \in I, \forall j \in J\} \tag{4.2.20}$$

For a fixed $j \in J$, define $\alpha = -1$, $\beta_i = R_{ij}^k$ for all $i \in I$, $\gamma_j = 0$ and $\gamma_\ell = \max_{i \in I}(R_{i\ell}^k - R_{ij}^k)$ for all $\ell \in J$ with $\ell \neq j$. This definition of the parameters satisfies $\alpha R_{ij}^k + \beta_i + \gamma_j \geq 0$ for all $i \in I$, $j \in J$. Substituting these parameters in the generic constraint of A^k yields

$$f^k \leq \sum_{i \in I} R_{ij}^k x_i + \sum_{\ell \in J : \ell \neq j} \max_{i \in I}(R_{i\ell}^k - R_{ij}^k) q_\ell^k \qquad \forall j \in J, \forall k \in K. \tag{4.2.21}$$

Constraint (4.2.21) implies Constraint (4.2.7) for the tight value of M^1 provided in Proposition 4.2.1 since for all $j \in J$ and $k \in K$,

$$\begin{aligned} \sum_{\ell \in J : \ell \neq j} \max_{i \in I}(R_{i\ell}^k - R_{ij}^k) q_\ell^k &\leq \max_{i \in I} \left\{ \max_{\ell \in J} R_{i\ell}^k - R_{ij}^k \right\} \sum_{\ell \in J : \ell \neq j} q_\ell^k \\ &= \max_{i \in I} \left\{ \max_{\ell \in J} R_{i\ell}^k - R_{ij}^k \right\} (1 - q_j^k). \end{aligned}$$

This proves the inclusion. To show that the inclusion may be strict, consider the following example where $|I| = |J| = 3$ and $|K| = 1$. Let the payoff matrix for the game be

$$(R,C) = \begin{pmatrix} (1,0) & (0,0) & (0,0) \\ (0,0) & (1,0) & (0,0) \\ (0,0) & (0,0) & (0,0) \end{pmatrix}$$

and consider the point defined by $x = (1,0,0)^t$, $q = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$, $s = 10$ and $f = 2/3$. Such a point is feasible for $(\overline{\text{D2}_{x,q,s,f}})$ but violates Constraint (4.2.21) for $j = 2$ and is therefore infeasible for $Proj_{x,q,s,f}\mathcal{P}(\overline{\text{DOBSS}_{x,q,z,s,f}})$. ■

Next, we compare the polyhedra $\mathcal{P}(\overline{\text{MIP-}p\text{-G}_{q,z}})$ and $Proj_{q,z}\mathcal{P}(\overline{\text{DOBSS}_{q,z,s}})$.

Theorem 4.2.1. *$\mathcal{P}(\overline{\text{MIP-p-G}_{q,z}}) \subseteq \mathcal{P}(\overline{\text{DOBSS}_{q,z}}) = Proj_{q,z}\mathcal{P}(\overline{\text{DOBSS}_{q,z,s}})$. Further, there exist instances for which the inclusion is strict.*

Proof. The description of $\mathcal{P}(\overline{\text{DOBSS}_{q,z}})$ differs from that of $\mathcal{P}(\overline{\text{MIP-}p\text{-G}_{q,z}})$ by only one constraint: (4.2.13) must hold instead of (4.2.15). Hence, the remainder of the proof consists in showing that (4.2.13) is implied by (4.2.1), (4.2.8)-(4.2.10), (4.2.15) and the nonnegativity of the q variables. The LHS of (4.2.13) can be rewritten as:

$$\sum_{i\in I}(C^k_{ij} - C^k_{i\ell})z^k_{i\ell} + \sum_{i\in I}\sum_{j'\in J:j'\neq\ell}(C^k_{ij} - C^k_{i\ell})z^k_{ij'} \leq \sum_{i\in I}\sum_{j'\in J:j'\neq\ell}(C^k_{ij} - C^k_{i\ell})z^k_{ij'}, \text{ using (4.2.15)},$$

$$\leq \max_{i\in I}\{C^k_{ij} - C^k_{i\ell}\}\sum_{j'\in J:j'\neq\ell}\sum_{i\in I} z^k_{ij'} \leq M^2 \sum_{j'\in J:j'\neq\ell} q^k_{j'}, \text{ given Proposition 4.2.1 and (4.2.16)}$$

$$= M^2(1-q^k_\ell), \text{ by (4.2.1).}$$

To show that the inclusion may be strict consider the p-follower GSG between a leader and a fixed follower $k \in K$ where the payoff bimatrix is:

$$(R^k, C^k) = \begin{pmatrix} (0,1) & (1,0) \\ (0,0) & (0,0) \end{pmatrix}$$

The point with coordinates $x = (1/2, 1/2)^t$, $q^k = (1/2, 1/2)^t$ and

$$z^k = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}$$

has an objective value of 1/4 and is feasible in $\mathcal{P}(\overline{\text{DOBSS}_{q,z}})$. However it is not a feasible point in $\mathcal{P}(\overline{\text{MIP-}p\text{-G}_{q,z}})$ as it doesn't verify Constraint (4.2.15) for values of $j = 2$ and $\ell = 1$. ∎

From an interpretation point of view, (MIP-p-G$_{q,z}$) can be seen as the result of applying Reformulation Linearization Technique (RLT) [Sherali and Adams, 1994] to (DOBSS$_{q,z}$). Indeed, by multiplying both sides of Constraint (4.2.12) by variable q^k_ℓ and noticing that $q^k_\ell(1-q^k_\ell) = 0$ since q is binary, one obtains $\sum_{i\in I}(C^k_{ij} - C^k_{i\ell})x_i q^k_\ell \leq 0$ which, once linearized by introducing variables $z^k_{i\ell}$, yields (4.2.15).

For a given formulation F, we denote its optimal value by $v(\text{F})$ and the optimal value of its LP relaxation by $v(\overline{\text{F}})$. Since (D2$_{x,q,s,f}$) and (DOBSS$_{x,q,s,f}$) and (DOBSS$_{q,z}$) and (MIP-p-G$_{q,z}$) have the same objective function, the following corollary holds.

Corollary 1. $v(\overline{\textit{MIP-p-G}_{q,z}}) \leq v(\overline{\textit{DOBSS}_{q,z}}) = v(\overline{\textit{DOBSS}_{x,q,s,f}}) \leq v(\overline{\textit{D2}_{x,q,s,f}})$.

Finally, when (MIP-p-G) is restricted to a single follower type, [Conitzer and Korzhyk, 2011] showed that the integrality constraints are redundant, *i.e.*, the remaining constraints in (MIP-1-G) provide a complete linear description of the convex hull of feasible solutions. We provide a simple proof of this result, due to [Moerenhout, 2012] in Section A.3 of Appendix A.

4.2.3 Computational experiments for GSGs

We present computational experiments for the formulations in the previous section. The machine used for these experiments is an Intel Core i7-4930K CPU, 3.40GHz, equipped with 64 GByte RAM, 6 cores, 12 threads and operating system Ubuntu release 12.10 (kernel linux 3.5.0-41-generic). The experiments were coded in the programming language Python and GUROBI version 6.5.1 was the optimization solver used with a 3 hour solution time limit. The instances solved in the computational experiments are randomly generated. We consider two different ways of randomly generating the payoff matrices for the leader and the different follower types: i) matrices where all the elements are randomly generated between 0 and 10; ii) matrices where 90% of the values are between 0 and 10 but we allow for 10% of the data to deviate between 0 and 100. In the first case we say that there is no variability in the payoff matrices, in the sense that all the data is uniformly distributed, whereas in the second case, we refer to the payoff matrices as matrices with variability.
A general Stackelberg game instance is defined by three parameters: $|I|$, the number of leader pure strategies, $|J|$, the number of follower pure strategies and $|K|$, the number of follower types. For the purpose of these experiments, we have considered instances where $|I| \in \{10, 20, 30\}$, $|J| \in \{10, 20, 30\}$ and $|K| \in \{2, 4, 6\}$. For each instance size, 5 instances are generated without variability in the payoff matrices and 5 are generated with variability. In total, we consider 135 instances without variability and 135 instances with variability.
Performance profiles summarize the results, with respect to the following 4 measures: total running time employed to solve the integer problem, running time employed to solve the linear relaxation of the integer problem, total number of nodes explored in the branch and bound solving scheme and gap percentage at the root node. The gap percentage at the root node is calculated by comparing the optimal values of the formulation and of its LP relaxation: $\frac{v(\overline{\mathrm{F}})-v(F)}{v(F)} \cdot 100$. A performance profile graph plots the total percentage of problems solved for each value of these measures.
We study the behavior of ($\mathrm{D2}_{x,q,s,f}$), ($\mathrm{D2}_{x,q,f}$), ($\mathrm{DOBSS}_{q,z,s}$), ($\mathrm{DOBSS}_{q,z}$) and (MIP-p-$\mathrm{G}_{q,z}$). Figures 4.2.1 and 4.2.2 compare the performance profiles when the payoff matrices are generated without variability and with variability, respectively.

Observe that the instances where variability is introduced in the payoff matrices solve faster than those where no variability is considered. When there is no variability, ($\mathrm{DOBSS}_{q,z,s}$) and (MIP-p-$\mathrm{G}_{q,z}$) are the two most competitive formulations. ($\mathrm{D2}_{x,q,s,f}$) can also be solved efficiently for the mid-range instances but slows down for the more difficult instances. Introducing variability in the payoff matrices, however, leads to a dominance of (MIP-p-$\mathrm{G}_{q,z}$) with ($\mathrm{DOBSS}_{q,z,s}$) coming in a close second and ($\mathrm{D2}_{x,q,s}$) becoming noncompetitive for

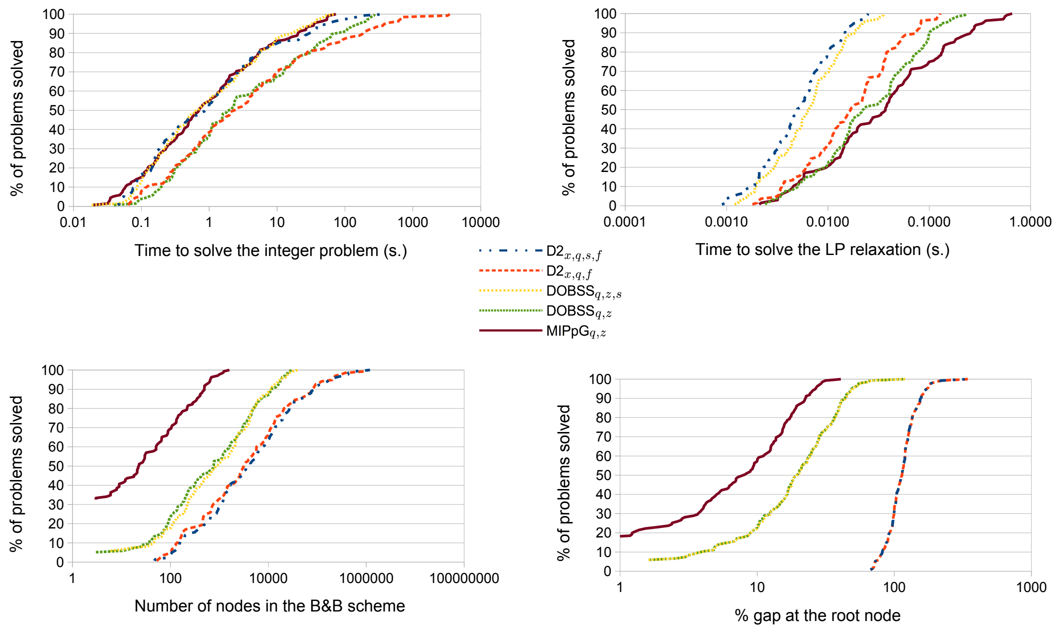


Figure 4.2.1: GSGs: $|I| \in \{10, 20, 30\}, |J| \in \{10, 20, 30\}, |K| \in \{2, 4, 6\}$–without variability

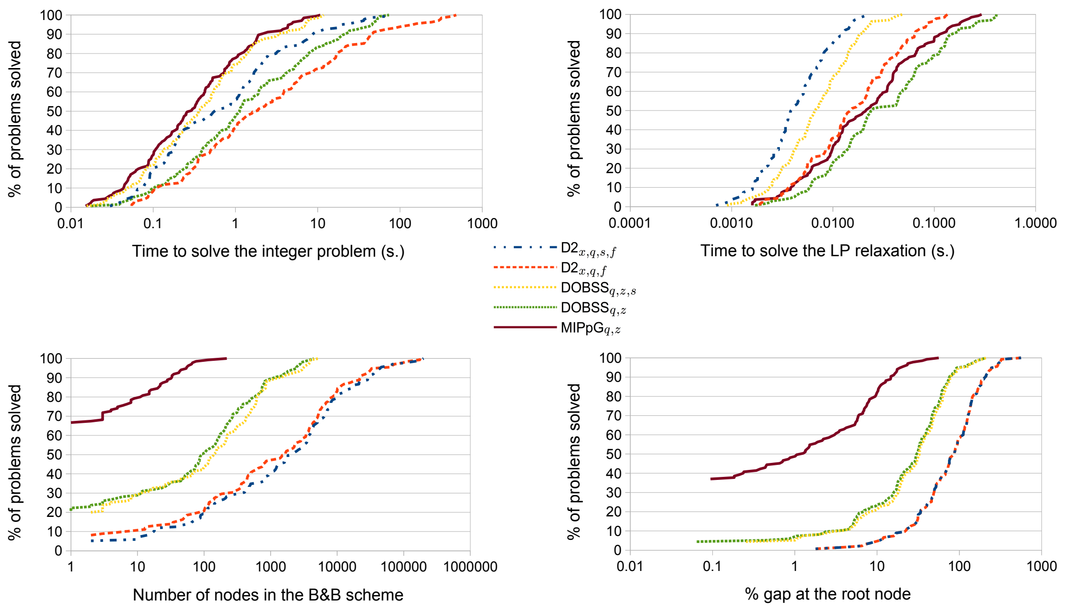


Figure 4.2.2: GSGs: $|I| \in \{10, 20, 30\}, |J| \in \{10, 20, 30\}, |K| \in \{2, 4, 6\}$–with variability

these instances. In what regards the time spent solving the linear relaxation of the problems, (MIP-p-$G_{q,z}$) is the formulation that is hardest to solve, this due to the fact that is has the most variables and constraints, $\mathcal{O}(|K||J|^2)$. On the other hand, ($D2_{x,q,s,f}$), that has the lightest LP relaxation, with $\mathcal{O}(|K||J|)$ variables and constraints, is the fastest. With respect to the number of nodes and gap percentage the theoretical findings shown in the

previous section are corroborated: (MIP-p-G$_{q,z}$) is the tightest formulation and therefore uses the fewest nodes. The effect is further intensified when variability is introduced.
Table 4.2.1 summarizes the mean gap obtained across the instances solved. Finally, remark that the formulations obtained through Fourier-Motzkin, (D2$_{x,q,f}$) and (DOBSS$_{q,z}$), explore slightly less nodes in the branch and bound scheme than their counterparts, (D2$_{x,q,s,f}$) and (DOBSS$_{q,z,s}$), but because of the increase in the number of constraints the time to solve each linear relaxation increases. This increases the overall solution time of the Fourier-Motzkin formulations.

	(D2$_{x,q,s,f}$)	(DOBSS$_{q,z,s}$)	(MIP-p-G$_{q,z}$)
Mean gap % (no variability)	117.68	23.01	9.94
Mean gap % (with variability)	103.44	40.74	5.17
Total mean gap %	110.56	31.88	7.56

Table 4.2.1: Mean gap percentage recorded for GSG formulations

4.3 Stackelberg security games–SSGs

In this section, we present three SSG formulations: (ERASER$_{c,q,s,f}$), due to [Kiekintveld et al., 2009], and (SDOBSS$_{q,y,s}$) and (MIP-p-S$_{q,y}$). We construct these formulations by exploring the inherent link between the general setting, considered up to now and the security setting, defined in Section 2.2. In this setting, the defender pure strategies $i \in I$ are the different ways in which up to m targets can be protected simultaneously. For this problem, one can think of $i \in I$ as a set indicating which targets are covered by security resources. It thus follows that the payoff matrices of SSGs satisfy:

$$R^k_{ij} = \begin{cases} D^k(j|c) & \text{if } j \in i \\ D^k(j|u) & \text{if } j \notin i \end{cases} \qquad (4.3.1) \qquad\qquad C^k_{ij} = \begin{cases} A^k(j|c) & \text{if } j \in i \\ A^k(j|u) & \text{if } j \notin i \end{cases} \qquad (4.3.2)$$

The payoff for the leader when he commits to a pure strategy $i \in I$ and a follower of type $k \in K$ responds by selecting strategy $j \in J$ is either a reward if pure strategy $i \in I$ allocates security coverage to attacked target $j \in J$, or, a penalty if strategy i does not cover target j. The same argument explains the link between payoffs for the attackers.

4.3.1 Single level MILP formulations

The first formulation derived is based on (D2$_{x,q,s,f}$). Consider (D2$_{c,x,q,s,f}$), an extended description of (D2$_{x,q,s,f}$) where the c variables are introduced through Constraint (4.3.4) (see Section 2.2). We further use relations (4.3.1) and (4.3.2) to adapt the payoff structure:

$$
\begin{aligned}
&(\mathrm{D2}_{c,x,q,s,f}) && && \\
&\text{Max} && \sum_{k\in K} \pi^k f^k && && (4.3.3)\\
&\text{s.t.} && \sum_{i\in I: j\in i} x_i = c_j && \forall j \in J && (4.3.4)\\
& && \sum_{j\in J} q_j^k = 1 && \forall k \in K, && (4.3.5)\\
& && q_j^k \in \{0,1\} && \forall j \in J, \forall k \in K, && (4.3.6)\\
& && \sum_{i\in I} x_i = 1, && && (4.3.7)\\
& && x_i \geq 0 && \forall i \in I && (4.3.8)\\
& && f^k \leq D^k(j|c)c_j + D^k(j|u)(1-c_j) + && && \\
& && \quad (1-q_j^k)\cdot M^1 && \forall j \in J, \forall k \in K, && (4.3.9)\\
& && 0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1-c_j) && && \\
& && \quad \leq (1-q_j^k)\cdot M^2 && \forall j \in J, \forall k \in K, && (4.3.10)\\
& && s, f \in \mathbb{R}^K. && &&
\end{aligned}
$$

This extended formulation is equivalent to $(\mathrm{D2}_{x,q,s,f})$, because, even though they are defined in different spaces of variables, the value of their LP relaxations coincide.

The formulation above has a large number of non-negative variables since in the security setting, the set I of all defender pure strategies is exponential in the number of targets as it contains all subsets of at most m targets of J that the defender can protect simultaneously. In order to avoid having exponentially many non-negative variables in the formulation, we project out variables $x_i,\ i \in I$, from the formulation. Note that only Constraints (4.3.4), (4.3.7) and (4.3.8) involve said variables. The following Proposition shows how this is done.

Proposition 4.3.1. *Consider the following two sets:*

$$A = Proj_c \Big\{ (x,c) \in \mathbb{R}^{|I|} \times \mathbb{R}^{|J|} : (4.3.4), (4.3.7), (4.3.8) \Big\}$$

$$B = \left\{ c \in \mathbb{R}^{|J|} : \sum_{j\in J} c_j \leq m,\ c_j \in [0,1]\ \forall j \in J \right\}$$

Then, $A = B$.

Proof. Remark first that using Farkas' Lemma [Farkas, 1902]:

$$A = \left\{ c \in \mathbb{R}^{|J|} : \sum_{j\in J} \alpha_j c_j + \alpha_{|J|+1} \geq 0\ \forall \alpha \in \mathbb{R}^{|J|+1} : \right.$$

$$\left.\sum_{j\in J:j\in i}\alpha_j+\alpha_{|J|+1}\geq 0\ \forall i\in I:|i|\leq m \text{ and } \alpha_{|J|+1}\geq 0\right\},$$

Thus $A\subseteq B$. Indeed, the following $2|J|+1$ vectors in $\mathbb{R}^{|J|+1}$:

$$\forall j\in J,\ e^j\in\mathbb{R}^{|J|+1}:e^j_j=1,\ e^j_k=0\ \forall k\in J:k\neq j \text{ and } e^j_{|J|+1}=0,$$

$$\forall j\in J,\ f^j\in\mathbb{R}^{|J|+1}:f^j_j=-1,\ f^j_k=0\ \forall k\in J:k\neq j \text{ and } f^j_{|J|+1}=1 \text{ and}$$

$$g\in\mathbb{R}^{|J|+1}:g_j=-1\ \forall j\in J \text{ and } g_{|J|+1}=m,$$

satisfy $\sum_{j\in J:j\in i}\alpha_j+\alpha_{|J|+1}\geq 0$ and $\alpha_{|J|+1}\geq 0$. Additionally, when substituting the above vectors into the generic constraint defining A, they yield all the constraints defining B.

To show that $A=B$, it remains to show that any other inequality

$$\sum_{j\in J}\alpha_j c_j+\alpha_{|J|+1}\geq 0 \tag{4.3.11}$$

such that α satisfies

$$\sum_{j\in J:j\in i}\alpha_j+\alpha_{|J|+1}\geq 0 \qquad \forall i\in I:|i|\leq m \text{ and } \alpha_{|J|+1}\geq 0, \tag{4.3.12}$$

is dominated by some nonnegative linear combination of the constraints defining B.

First, note that one can restrict one's attention to constraints such that $\alpha_j\leq 0$ for all $j\in J$. If there exists $\hat{j}\in J$ such that $\alpha_{\hat{j}}>0$, since α must satisfy (4.3.12) and $|i\setminus\{\hat{j}\}|\leq|i|\leq m$, it follows that $\bar{\alpha}$ with $\bar{\alpha}_{\hat{j}}=0$ and $\bar{\alpha}_j=\alpha_j$ for all $j\in J\setminus\{\hat{j}\}$ also satisfies (4.3.12) and since $c\geq 0$, it follows that

$$\sum_{j\in J}\bar{\alpha}_j c_j+\bar{\alpha}_{|J|+1}\leq\sum_{j\in J}\alpha_j c_j+\alpha_{|J|+1}.$$

Therefore, the constraint defined by α is dominated by the constraint defined by $\bar{\alpha}$. We thus distinguish two cases of α satisfying (4.3.12):

Case 1. $|\{j:\alpha_j<0\}|=k\leq m$, and

Case 2. $|\{j:\alpha_j<0\}|=k>m$.

In Case 1, by considering a linear combination of inequalities $c_j\leq 1$ for $1\leq j\leq k$ with respective weights $-\alpha_j\geq 0$, one obtains that:

$$0\leq\sum_{j=1}^{k}\alpha_j c_j-\sum_{j=1}^{k}\alpha_j\leq\sum_{j\in J}\alpha_j c_j+\alpha_{|J|+1},$$

since $\alpha_j=0$ for all $j>k$ and α satisfies (4.3.12) for $i=\{1,\dots,k\}$.

For Case 2, assume *w.l.o.g* that $\alpha_1\leq\alpha_2\leq\dots\leq\alpha_k<0$ and $\alpha_j=0$ for all $j>k$. Then, build a linear combination of inequality $\sum_{j\in J}c_j\leq m$ with weight $-\alpha_m\geq 0$ and inequalities $c_j \leq 1$ for $1 \leq j \leq m$ with respective weights $\alpha_m - \alpha_j \geq 0$. The valid inequality thus obtained is:

$$0 \leq \sum_{j=1}^{m} \alpha_j c_j + \sum_{j>m} \alpha_m c_j - \sum_{j=1}^{m} \alpha_j \leq \sum_{j \in J} \alpha_j c_j - \sum_{j=1}^{m} \alpha_j, \text{ since } \alpha_j \geq \alpha_m \text{ for all } j > m$$

$$\leq \sum_{j \in J} \alpha_j c_j + \alpha_{|J|+1},$$

since α satisfies (4.3.12) for $i = \{1, \ldots, m\}$. ∎

Proposition 4.3.1 leads to the following formulation based on (D2$_{c,x,q,s,f}$):

(ERASER$_{c,q,s,f}$)

$$\begin{aligned}
\text{Max} \quad & \sum_{k \in K} \pi^k f^k && && (4.3.13)\\
\text{s.t.} \quad & \sum_{j \in J} c_j \leq m, && && \\
& 0 \leq c_j \leq 1 && \forall j \in J, && \\
& \sum_{j \in J} q_j^k = 1 && \forall k \in K, && \\
& q_j^k \in \{0,1\} && \forall j \in J, \forall k \in K, && \\
& f^k \leq D^k(j|c)c_j + D^k(j|u)(1-c_j) + && && \\
& \qquad (1-q_j^k) \cdot M^1 && \forall j \in J, \forall k \in K, && (4.3.14)\\
& 0 \leq s^k - A^k(j|c)c_j - A^k(j|u)(1-c_j) && && \\
& \qquad \leq (1-q_j^k) \cdot M^2 && \forall j \in J, \forall k \in K, && (4.3.15)\\
& s, f \in \mathbb{R}^K. && &&
\end{aligned}$$

The above formulation involves a polynomial number of variables and constraints and was presented in [Kiekintveld et al., 2009]. The next result is also an immediate consequence of Proposition 4.3.1.

Corollary 2. *$Proj_{c,q,s,f}\mathcal{P}(\overline{D2_{c,x,q,s,f}}) = \mathcal{P}(\overline{ERASER_{c,q,s,f}})$.*

We now derive SSG formulations based on (DOBSS$_{q,z,s}$) and (MIP-p-G$_{q,z}$). We first present extended descriptions of both formulations by considering $y_{\ell j}^k$ variables satisfying:

$$y_{\ell j}^k = \sum_{i \in I: \ell \in i} z_{ij}^k \qquad \forall j, \ell \in J, \forall k \in K. \qquad (4.3.16)$$

We use (4.3.1) and (4.3.2) to adapt the payoffs to the security setting leading to:

$$
\begin{aligned}
&(\text{DOBSS}_{q,z,y,s})\\
&\text{Max} && \sum_{j\in J}\sum_{k\in K}\{\pi^k(D^k(j|c)y^k_{jj}+ \\
& && \quad D^k(j|u)(q^k_j - y^k_{jj}))\} && && (4.3.17)\\
&\text{s.t.} && \sum_{j\in J} z^k_{ij} = \sum_{j\in J} z^1_{ij} && \forall i\in I, \forall k\in K, && (4.3.18)\\
& && \sum_{i\in I:\ell\in i} z^k_{ij} = y^k_{\ell j} && \forall \ell, j\in J, \forall k\in K, && (4.3.19)\\
& && \sum_{i\in I} z^k_{ij} = q^k_j && \forall j\in J, \forall k\in K, && (4.3.20)\\
& && z^k_{ij} \geq 0 && \forall i\in I, \forall j\in J, \forall k\in K, && (4.3.21)\\
& && 0 \leq s^k - A^k(j|c)\sum_{j'\in J} y^k_{jj'} - \\
& && \quad A^k(j|u)(1-\sum_{j'\in J} y^k_{jj'}) \leq (1-q^k_j)\cdot M^2 && \forall j\in J, \forall k\in K, && (4.3.22)\\
& && \sum_{j\in J} q^k_j = 1 && \forall k\in K, && (4.3.23)\\
& && q^k_j \in \{0,1\} && \forall j\in J, \forall k\in K, && (4.3.24)\\
& && s\in \mathbb{R}^{|K|}. && && (4.3.25)
\end{aligned}
$$

$$
\begin{aligned}
&(\text{MIP-}p\text{-G}_{q,z,y})\\
&\text{Max} && \sum_{j\in J}\sum_{k\in K}\pi^k(D^k(j|c)y^k_{jj} + D^k(j|u)(q^k_j - y^k_{jj}))\\
&\text{s.t.} && (4.3.18) - (4.3.21), (4.3.23) - (4.3.24)\\
& && A^k(j|c)y^k_{jj} + A^k(j|u)(q^k_j - y^k_{jj}) - \\
& && \quad A^k(\ell|c)y^k_{\ell j} - A^k(\ell|u)(q^k_j - y^k_{\ell j}) \geq 0 && \forall j,\ell\in J, \forall k\in K. && (4.3.26)
\end{aligned}
$$

Further, consider the following constraint:

$$\sum_{j\in J} y^k_{\ell j} = \sum_{j\in J} y^1_{\ell j} \qquad \forall \ell\in J, \forall k\in K, \tag{4.3.27}$$

and let us define the following polyhedra C and D:

$$C := \left\{(q,z,y,s)\in[0,1]^{|K||J|}\times[0,1]^{|K||I||J|}\times[0,1]^{|K||J|^2}\times\mathbb{R}^{|K|} : (4.3.19)-(4.3.23),(4.3.25),(4.3.27)\right\}$$

$$D := \left\{(q,z,y)\in[0,1]^{|K||J|}\times[0,1]^{|K||I||J|}\times[0,1]^{|K||J|^2} : (4.3.19)-(4.3.21),(4.3.23),(4.3.26),(4.3.27)\right\}$$

Lemma 4.3.1. *$C \supseteq \mathcal{P}(\overline{DOBSS_{q,z,y,s}})$ and $D \supseteq \mathcal{P}(\overline{MIP\text{-}p\text{-}G_{q,z,y}})$*

Proof. Consider Constraint (4.3.18) and sum over all $i \in I$ such that $\ell \in i$:

$$\sum_{\substack{i \in I: \\ \ell \in i}} \sum_{j \in J} z_{ij}^k = \sum_{\substack{i \in I: \\ \ell \in i}} \sum_{j \in J} z_{ij}^1 \qquad \forall \ell \in J, \forall k \in K. \tag{4.3.28}$$

Applying (4.3.19) to (4.3.28) yields (4.3.27) and the result follows. ■

We now project the z variables from the larger polyhedra C and D. Said variables only appear in Constraints (4.3.19)-(4.3.21).

Lemma 4.3.2. *Consider the following two sets;*

$$\mathcal{X} = Proj_{q,y} \left\{ (q, z, y) \in \mathbb{R}^{|K||J|^2+|K||J|+|I||J||K|} : (4.3.19) - (4.3.21) \right\}$$

$$\mathcal{Y} = \{(q, y) \in \mathbb{R}^{|K||J|^2+|K||J|} : \sum_{\ell \in J} y_{\ell j}^k \leq m q_j^k \ \forall j \in J, \forall k \in K,$$

$$0 \leq y_{\ell j}^k \leq q_j^k \ \forall j, \ell \in J, \forall k \in K\}$$

Then, $\mathcal{X} = \mathcal{Y}$.

Proof. Note that Constraints (4.3.19)-(4.3.21) can be treated independently for each $k \in K$ and each $j \in J$. First consider the case where $q_{\hat{j}}^{\hat{k}} = 0$ for $\hat{j} \in J$ and $\hat{k} \in K$. Constraint (4.3.20) then implies that for all $i \in I$, $z_{i\hat{j}}^{\hat{k}} = 0$ and Constraint (4.3.19) forces $y_{\ell \hat{j}}^{\hat{k}} = 0$ for all $\ell \in J$ and the result holds. For all $j \in J$, $k \in K$ such that $q_j^k \neq 0$, consider $x_i = z_{ij}^k / q_j^k$ and $c_\ell = y_{\ell j}^k / q_j^k$ and apply Propostion 4.3.1. The result follows. ■

Consider $Proj_{q,y,s}C$ and $Proj_{q,y}D$ as the feasible regions of the linear relaxations of two MILP formulations–(SDOBSS$_{q,y,s}$) and (MIP-p-S$_{q,y}$)–where one maximizes the objective function (4.3.17) under the additional requirement that the q variables be binary. Hence, we present (SDOBSS$_{q,y,s}$), a security formulation based on (DOBSS$_{q,z,y,s}$),

(SDOBSS$_{q,y,s}$)

$$\text{Max} \quad \sum_{j \in J} \sum_{k \in K} \pi^k (D^k(j|c) y_{jj}^k + D^k(j|u)(q_j^k - y_{jj}^k))$$

$$\text{s.t.} \quad \sum_{j \in J} q_j^k = 1 \qquad \forall k \in K \tag{4.3.29}$$

$$q_j^k \in \{0, 1\} \qquad \forall j \in J, \forall k \in K, \tag{4.3.30}$$

$$\sum_{j \in J} y_{\ell j}^k = \sum_{j \in J} y_{\ell j}^1 \qquad \forall \ell \in J, \forall k \in K, \tag{4.3.31}$$

$$\sum_{\ell \in J} y_{\ell j}^k \leq m q_j^k \qquad \forall j \in J, \forall k \in K, \tag{4.3.32}$$

$$0 \leq y_{\ell j}^k \leq q_j^k \qquad \forall j, \ell \in J, \forall k \in K, \tag{4.3.33}$$

$$0 \leq s^k - A^k(j|c) \sum_{j' \in J} y^k_{jj'} - \\ A^k(j|u)(1 - \sum_{j' \in J} y^k_{jj'}) \leq (1 - q^k_j) \cdot M^2 \qquad \forall j \in J, \forall k \in K, \tag{4.3.34}$$

$$s \in \mathbb{R}^{|K|}.$$

And we also present (MIP-p-S$_{q,y}$), a security formulation based on (MIP-p-G$_{q,z,y}$),

(MIP-p-S$_{q,y}$)

$$\text{Max} \quad \sum_{j \in J} \sum_{k \in K} \pi^k (D^k(j|c) y^k_{jj} + D^k(j|u)(q^k_j - y^k_{jj}))$$

$$\text{s.t.} \quad (4.3.29) - (4.3.33)$$

$$A^k(j|c) y^k_{jj} + A^k(j|u)(q^k_j - y^k_{jj}) - \\ A^k(\ell|c) y^k_{\ell j} - A^k(\ell|u)(q^k_j - y^k_{\ell j}) \geq 0 \qquad \forall j, \ell \in J, \forall k \in K. \tag{4.3.35}$$

The following corollaries are an immediate consequence of Lemmas 4.3.1 and 4.3.2.

Corollary 3. *$Proj_{q,y,s} \mathcal{P}(\overline{DOBSS_{q,z,y,s}}) \subseteq \mathcal{P}(\overline{SDOBSS_{q,y,s}})$.*

Corollary 4. *$Proj_{q,y} \mathcal{P}(\overline{MIP\text{-}p\text{-}G_{q,z,y}}) \subseteq \mathcal{P}(\overline{MIP\text{-}p\text{-}S_{q,y}})$.*

In addition, note that if we restrict (MIP-p-G$_{q,z,y}$) to a single type of follower, Constraint (4.3.18) disappears and one thus obtains the following corollary.

Corollary 5. *$Proj_{q,y} \mathcal{P}(\overline{MIP\text{-}1\text{-}G_{q,z,y}}) = \mathcal{P}(\overline{MIP\text{-}1\text{-}S_{q,y}})$*

The above corollary immediately leads to the following theorem.

Theorem 4.3.1. *$(\overline{MIP\text{-}1\text{-}S_{q,y}})$ is a linear description of the convex hull of feasible solutions for the Stackelberg security game with a single type of attacker.*

Proof. The result follows from Corollary 5 and from [Conitzer and Korzhyk, 2011] showing that $(\overline{\text{MIP-1-G}_{q,z}})$ is a linear description for general games. ■

As in general games, we can use Fourier-Motzkin elimination on Constraints (4.3.15) and (4.3.34) to project out the s variables from formulations (ERASER$_{c,q,s,f}$) and (SDOBSS$_{q,y,s}$) respectively. This leads to the following two families of inequalities:

$$(A^k(j|c) - A^k(j|u)) c_j + (A^k(\ell|u) - A^k(\ell|c)) c_\ell + A^k(j|u) - A^k(\ell|u) \leq \\ (1 - q^k_\ell) \cdot M^2 \ \forall j, \ell \in J, \forall k \in K, \tag{4.3.36}$$

$$(A^k(j|c) - A^k(j|u)) \sum_{h \in J} y^k_{jh} + (A^k(\ell|u) - A^k(\ell|c)) \sum_{h \in J} y^k_{\ell h} + \\ A^k(j|u) - A^k(\ell|u) \leq (1 - q^k_\ell) \cdot M^2 \qquad \forall j, \ell \in J, \forall k \in K, \tag{4.3.37}$$

Replacing Constraint (4.3.15) by (4.3.36) in (ERASER$_{c,q,s,f}$) and (4.3.34) by (4.3.37) in (SDOBSS$_{q,y,s}$) leads to (ERASER$_{c,q,f}$) and (SDOBSS$_{q,y}$).

In the same spirit as Proposition 4.2.1, we present the following proposition, establishing the tightest values for the big-M constants in the formulations seen so far. A detailed proof can be found in the appendix, Section A.2.2

Proposition 4.3.2. *The tightest values for the positive constants M are:*

1. *In (4.3.14), $M1^{k^*}_j = \max_{\ell \in J}\{D^k(\ell|c), D^k(\ell|u)\} - \min\{D^k(j|c), D^k(j|u)\}$, $\forall j \in J, k \in K$.*

2. *In (4.3.15), (4.3.34), $M2^{k*}_j = \max_{\ell \in J}\{A^k(\ell|c), A^k(\ell|u)\} - \min\{A^k(j|c), A^k(j|u)\}$, $\forall j \in J, k \in K$.*

3. *In (4.3.36), (4.3.37), $M2^{k^*}_{\ell j} = \max\{A^k(j|c), A^k(j|u)\} - \min\{A^k(\ell|c), A^k(\ell|u)\}$, $\forall j, \ell \in J, k \in K$.*

4.3.2 Comparison of the formulations

First, we introduce an additional formulation which we denote by (SDOBSS$_{c,q,y,s,f}$). This formulation is equivalent to (SDOBSS$_{q,y,s}$), in the sense that the value of their LP relaxations coincide. In this formulation we introduce variables f^k for all $k \in K$ to rewrite the objective function so that it matches (4.3.13). We also add variables c_ℓ for all $\ell \in J$ and rewrite Constraint (4.3.31) as $\sum_{j \in J} y^k_{\ell j} = c_\ell$ for all $\ell \in J$ and all $k \in K$. Using this last condition one can simplify (4.3.34) to (4.3.15). The formulation (SDOBSS$_{c,q,y,s,f}$) is as follows.

$$
\begin{array}{lll}
(\text{SDOBSS}_{c,q,y,s,f}) & & \\
\text{Max} & \displaystyle\sum_{k \in K} \pi^k f^k & \\
\text{s.t.} & (4.3.29), (4.3.30), (4.3.32) - (4.3.34), & \\
& \displaystyle f^k = \sum_{j \in J} \{y^k_{jj}(D^k(j|c) - D^k(j|u)) + & \\
& q^k_j D^k(j|u)\} & \forall k \in K \qquad (4.3.38) \\
& \displaystyle\sum_{j \in J} y^k_{\ell j} = c_\ell & \forall \ell \in J, \forall k \in K, \qquad (4.3.39) \\
& s \in \mathbb{R}^{|K|}. &
\end{array}
$$

Note that

$$\mathcal{P}(\overline{\text{ERASER}_{c,q,f}}) = Proj_{c,q,f}\mathcal{P}(\overline{\text{ERASER}_{c,q,s,f}}) \text{ and}$$

$$\mathcal{P}(\overline{\text{SDOBSS}_{q,y}}) = Proj_{q,y}\mathcal{P}(\overline{\text{SDOBSS}_{q,y,s}}).$$

Proposition 4.3.3. *$Proj_{c,q,s,f}\mathcal{P}(\overline{SDOBSS_{c,q,y,s,f}}) \subseteq \mathcal{P}(\overline{ERASER_{c,q,s,f}})$. Further, there exist instances for which the inclusion is strict.*

Proof. The projection of $\mathcal{P}(\overline{\text{SDOBSS}_{c,q,y,s,f}})$ onto the (c,q,s,f)-space is obtained by applying Farkas' Lemma. Constraints (4.3.32)-(4.3.33) and (4.3.38)-(4.3.39) are the only ones involving variables $y^k_{\ell j}$ and are separable by $k \in K$. For a fixed $k \in K$, the projection is given by:

$$A^k = \{(c,q,f) : \alpha(f^k - \sum_{j\in J} D^k(j|u)q^k_j) + \sum_{\ell\in J}\beta_\ell c_\ell + m\sum_{j\in J}\gamma_j q^k_j + \sum_{j\in J}\sum_{\ell\in J}\delta_{\ell j} q^k_j \geq 0$$

$$\forall(\alpha,\beta,\gamma,\delta) : \gamma,\delta \geq 0,\ \beta_\ell + \gamma_j + \delta_{\ell j} \geq 0\ \forall \ell, j \in J : \ell \neq j, \text{ and}$$

$$\alpha(D^k(j|c) - D^k(j|u)) + \beta_j + \gamma_j + \delta_{jj} \geq 0\ \forall j \in J\} \tag{4.3.40}$$

Consider, for each $k \in K$, the following set B^k:

$$B^k = \{(c,q,f) : c_\ell \leq \sum_{j\in J} q^k_j, \qquad \forall \ell \in J, \tag{4.3.41}$$

$$c_\ell \geq 0, \qquad \forall \ell \in J, \tag{4.3.42}$$

$$\sum_{\ell\in J} c_\ell \leq m\sum_{j\in J} q^k_j, \tag{4.3.43}$$

$$f^k \leq c_j(D^k(j|c) - D^k(j|u)) + \sum_{\ell\in J:\ell\neq j} q^k_\ell D^k(\ell|c) + q^k_j D^k(j|u) \qquad \forall j \in J, \tag{4.3.44}$$

$$q^k_j \geq 0 \qquad \forall j \in J, \forall k \in K.\}$$

Let us see that $A^k \subseteq B^k$ for all $k \in K$. First note that if we set $\alpha = 0$, the following definitions of the parameters β, γ and δ comply with the conditions in (4.3.40):

$$\beta = e^h, \gamma = \{0\}_{j\in J}, \delta = \{0\}_{\ell,j\in J},\ \forall h \in J,$$

$$\beta = -e^\ell, \gamma = \{0\}_{j\in J}, \delta_\ell = \{1\}_{j\in J},\ \forall \ell \in J,$$

$$\beta = \{-1\}_{\ell\in J}, \gamma = \{1\}_{j\in J}, \delta = \{0\}_{\ell,j\in J},$$

$$\beta = \{0\}_{\ell\in J}, \gamma = \{0\}_{j\in J}, \delta_1 = \{e^j\},\ \forall j \in J.$$

Substituting these valid parameters into the generic constraint in A^k, produces all of the constraints in B^k except (4.3.44). Further, for a fixed $j \in J$, consider $\alpha = -1$, $\beta_\ell = 0$ for all $\ell \neq j$ and $\beta_j = D^k(j|c) - D^k(j|u)$. Set $\gamma_\ell = 0$ for all $\ell \in J$. Finally, set $\delta_{\ell j} = 0$ for all $\ell \neq j$, $\delta_{\ell\ell} = (D^k(\ell|c) - D^k(\ell|u))$ for all $\ell \neq j$ and $\delta_{jj} = 0$. This definition of parameters is valid as it satisfies the conditions in (4.3.40). Substituting in the generic constraint in A^k yields (4.3.44).

It remains to show that for all $k \in K$, Constraint (4.3.44) implies (4.3.14) for the tight value of M^1 shown in Proposition 4.3.2. The implication holds because

$$\begin{aligned}\sum_{\ell \in J:\ell \neq j} q_\ell^k D^k(\ell|c) &\leq \max_{\ell \in J}\{D^k(\ell|c)\} \sum_{\ell \in J:\ell \neq j} q_\ell^k \\ &= (1-q_j^k) \max_{\ell \in J}\{D^k(\ell|c)\} \qquad \forall j \in J, \forall k \in K.\end{aligned}$$

Hence, $Proj_{c,q,s,f}\mathcal{P}(\overline{\text{SDOBSS}_{c,q,y,s,f}}) \subseteq \mathcal{P}(\overline{\text{ERASER}_{c,q,s,f}})$. To show that the inclusion may be strict, consider the following example where $m = 1$, $|J| = 3$ and $|K| = 1$. Let the reward and penalty matrices for the defender and attacker be $D(\cdot|c) = [1, 0, 0]$, $D(\cdot|u) = [0, 0, 0]$, $A(\cdot|c) = [0, 0, 0]$ and $A(\cdot|u) = [0, 0, 0]$. Consider the point defined by $q = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})^t$, $c = (1, 0, 0)^t$, $s = 10$ and $f = 2/3$. Such a point is feasible for $(\overline{\text{ERASER}_{c,q,s,f}})$ but violates Constraint (4.3.44) for $j = 2$ and is therefore infeasible for $Proj_{c,q,f,s}\mathcal{P}(\overline{\text{SDOBSS}_{c,q,y,s,f}})$. ■

Based on Theorem 4.2.1 we can present the following theorem comparing the polyhedra $\mathcal{P}(\overline{\text{MIP-}p\text{-S}_{q,y}})$ and $Proj_{q,y}\mathcal{P}(\overline{\text{SDOBSS}_{q,y,s}})$:

Theorem 4.3.2. *$\mathcal{P}(\overline{\text{MIP-}p\text{-S}_{q,y}}) \subseteq \mathcal{P}(\overline{\text{SDOBSS}_{q,y}}) = Proj_{q,y}\mathcal{P}(\overline{\text{SDOBSS}_{q,y,s}})$.*

Proof. The inclusion is a consequence of Theorem 4.2.1, the relations between the payoffs described in (4.3.1) and (4.3.2) and the relation between the z and y variables described in (4.3.16).

To show that the inclusion may be strict, consider the following game. We set $m = 2$, $|J| = 2$ and $|K| = 1$. The reward and penalty payoff matrices for both the defender and the attacker are given by $D(\cdot|c) = [1, 0]$, $D(\cdot|u) = [0, 0]$, $A(\cdot|c) = [0, 0]$ and $A(\cdot|u) = [0, 1]$. Additionally, the point with coordinates

$$c^t = (1/2, 1/2),\ q^t = (1/2, 1/2) \text{ and } y^k = \begin{pmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{pmatrix}$$

has an objective value of 1/4 and is a valid feasible solution of $\mathcal{P}(\overline{\text{SDOBSS}_{q,y}})$. However, it is not feasible in $\mathcal{P}(\overline{\text{MIP-}p\text{-S}_{q,y}})$ as it does not verify Constraint (4.3.35) for $j = 1$ and $\ell = 2$. ■

Remark that (MIP-p-S$_{q,y}$) can be obtained by applying RLT [Sherali and Adams, 1994] to (SDOBSS$_{q,y}$). Multiplying both sides of Constraint (4.3.36) by variable q_ℓ^k and noticing that $q_\ell^k(1 - q_\ell^k) = 0$, since q_ℓ^k is binary, one obtains a constraint that once linearized, introducing variables $y_{\ell j}^k$, yields (4.3.35).

Since (ERASER$_{c,q,s,f}$) and (f-SDOBSS$_{c,q,s,f}$) and (SDOBSS$_{q,y}$) and (MIP-p-S$_{q,y}$) have the same objective function, the following corollary holds.

Corollary 6. $v(\overline{MIP\text{-}p\text{-}S_{q,y}}) \leq v(\overline{SDOBSS_{q,y}}) = v(\overline{SDOBSS_{c,q,s,f}}) \leq v(\overline{ERASER_{c,q,s,f}})$.

To recap, in Figure 4.3.1 we present a symbolic representation of the ordering of the polyhedral regions of the linear relaxations of the GSG and SSG formulations as well as the projection link between the general-setting formulations and their security counterparts. Bear in mind that the polyhedron corresponding to the LP relaxation of (ERASER) coincides with the projection, on the appropriate space of variables, of the polyhedron of the LP relaxation of (D2). The remaining security polyhedra are somewhat larger than the corresponding projections, on the appropriate space of variables, of the general polyhedra from which they are constructed. We omit the specification of each formulation's space of variables as this can be adapted by means of appropriate variable projections.

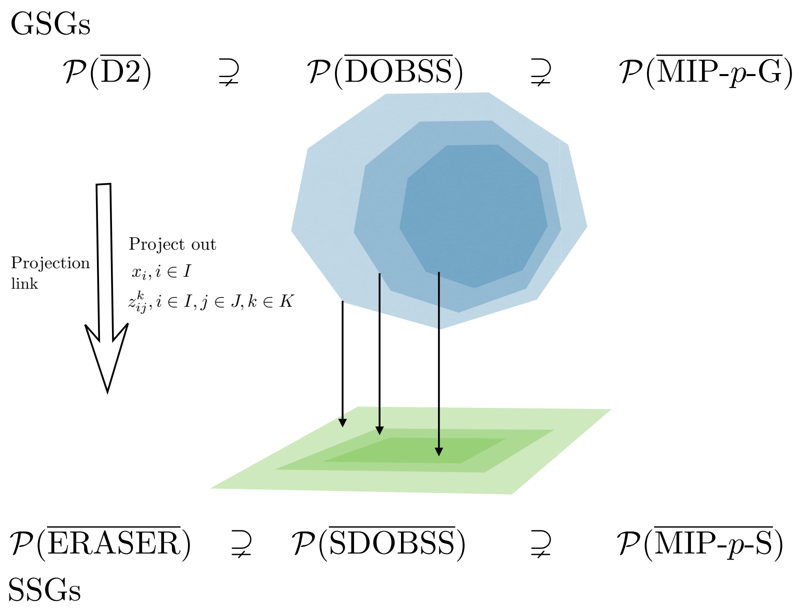

GSGs
$\mathcal{P}(\overline{\text{D2}}) \supsetneq \mathcal{P}(\overline{\text{DOBSS}}) \supsetneq \mathcal{P}(\overline{\text{MIP-}p\text{-G}})$
Projection link
Project out
$x_i, i \in I$
$z_{ij}^k, i \in I, j \in J, k \in K$
$\mathcal{P}(\overline{\text{ERASER}}) \supsetneq \mathcal{P}(\overline{\text{SDOBSS}}) \supsetneq \mathcal{P}(\overline{\text{MIP-}p\text{-S}})$
SSGs

Figure 4.3.1: Symbolic representation of inclusions between the polyhedral regions of the linear relaxations of GSG and SSG formulations and link between both settings

4.3.3 Computational experiments for SSGs

The security experiments are run on randomly generated instances. For each instance, four payoff matrices have to be generated that satisfy $D^k(\cdot|c) \geq D^k(\cdot|u)$ and $A^k(\cdot|u) \geq A^k(\cdot|c)$. We consider two ways of generating these matrices. First, we generate matrices where the values for the penalty matrices ($D^k(\cdot|u)$ and $A^k(\cdot|c)$) are randomly generated between 0 and 5 and all values for the reward matrices ($D^k(\cdot|c)$ and $A^k(\cdot|u)$) are randomly generated between 5 and 10. We shall refer to these as matrices with no variability. Second, we consider an alternative where 90% of the values for the penalty matrices are randomly generated between 0 and 5 (between 5 and 10 for the reward matrices) and 10% of the values for the penalty matrices are randomly generated between 0 and 50 (between 50 and 100 for the reward matrices). We refer to these as matrices with variability. We impose a solution limit of 3 hours.

A Stackelberg security game instance is defined by $|J|$, the number of targets, $|K|$ the number of attacker types and m, the number of security resources available to the defender. Recall from the computational experiments for GSGs that using payoff matrices with variability amounts to endowing the game with more structure, thus making it somewhat easier to solve. We have encountered the same phenomenon in SSGs. For games whose payoff matrices have variability, we have considered $J = \{30, 40, 50, 60, 70\}$, $K = \{6, 8, 10, 12\}$ and we have allowed m to be either 25%, 50% or 75% of the number of targets. For games whose payoff matrices don't have variability we have had to be less ambitious in order to solve all instances to optimality within the stipulated time limit and have considered $J = \{10, 20, 30, 40, 50\}$, $K = \{2, 4, 6, 8\}$ while still considering m to be either 25%, 50% or 75% of the number of targets. In either case, for each instance size we generate 5 random instances as described above. In total, we consider 300 randomly generated instances.

We study the behavior of ($\text{ERASER}_{c,q,s,f}$), ($\text{SDOBSS}_{q,y,s}$) and (MIP-p-$\text{S}_{q,y}$). For the sake of clarity we no longer consider the Fourier-Motzkin formulations ($\text{ERASER}_{c,q,f}$) and ($\text{SDOBSS}_{q,y}$). Performance-wise, ($\text{ERASER}_{c,q,s,f}$) and ($\text{SDOBSS}_{q,y,s}$) compare to their Fourier-Motzkin formulations in a similar way to how ($\text{D2}_{x,q,s,f}$) and ($\text{DOBSS}_{q,z,s}$) compared to theirs in Section 4.2. We plot performance profile graphs in Figures 4.3.2 and 4.3.3.

Remark that for the experiments with variability, ($\text{ERASER}_{c,q,s,f}$) is the fastest formulation for most of the instances. However, one sees that for the more difficult instances, its solution time increases significantly, surpassing the solution time of (MIP-p-$\text{S}_{q,y}$). For these instances ($\text{ERASER}_{c,q,s,f}$) is slower than (MIP-p-$\text{S}_{q,y}$). As for the instances whose payoff matrices have no variability, and are thus harder to solve, one can observe that ($\text{ERASER}_{c,q,s,f}$) outperforms the running time of the other two formulations for 80% of the instances. However, for the most difficult instances, (MIP-p-$\text{S}_{q,y}$) is faster than the other two formulations. For the last 5% of the instances, ($\text{ERASER}_{c,q,s,f}$) is the worst formulation. In terms of size of the formulations, ($\text{ERASER}_{c,q,s,f}$) is the formulation with the least number of constraints and variables: $\mathcal{O}(|J||K|)$. Observe that (MIP-p-$\text{S}_{q,y}$) and ($\text{SDOBSS}_{q,y,s}$) have $\mathcal{O}(|J|^2|K|)$ constraints and variables. Thus, these formulations have significantly heavier LP relaxations and thus take longer time to solve than ($\text{ERASER}_{c,q,s,f}$) does. However, Figures 4.3.2 and 4.3.3 confirm the theoretical findings: (MIP-p-$\text{S}_{q,y}$) has the tightest LP relaxation and this translates into a clear dominance with respect to node

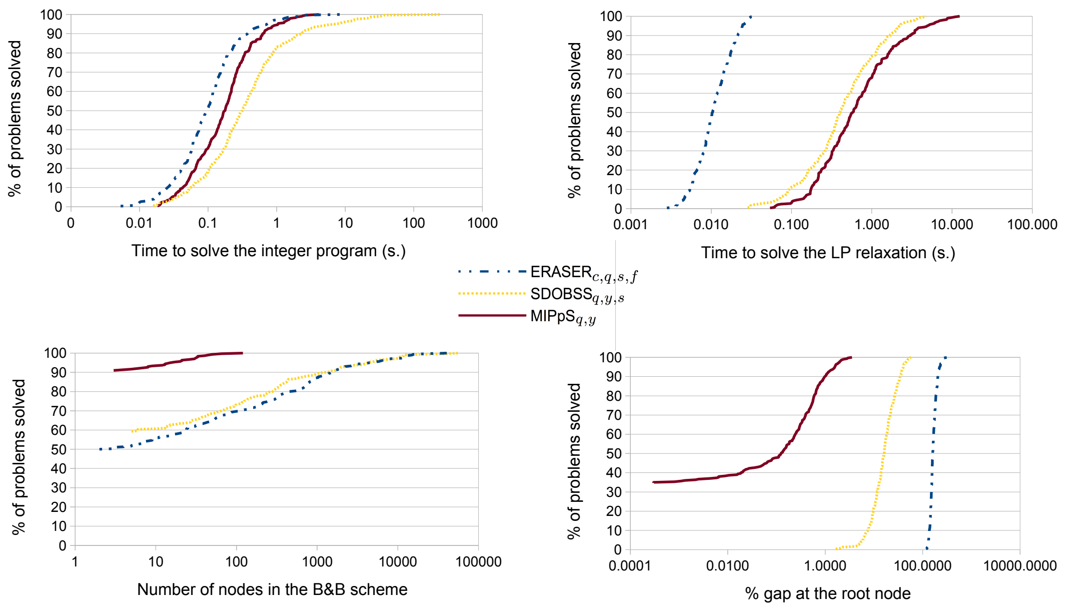


Figure 4.3.2: SSGs: $K = \{6, 8, 10, 12\}$, $J = \{30, 40, 50, 60, 70\}$–with variability

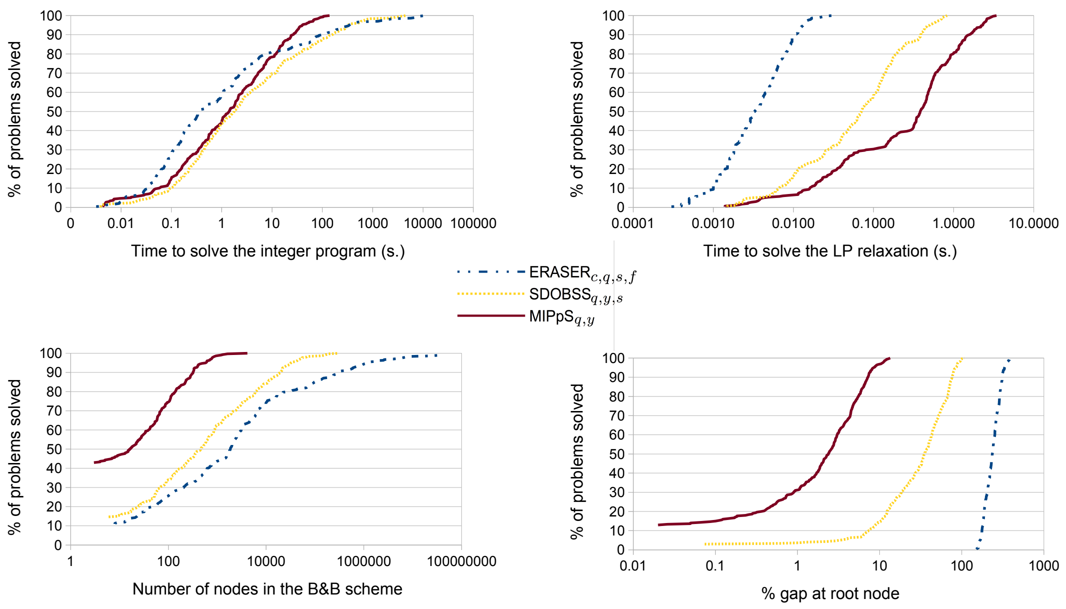


Figure 4.3.3: SSGs: $K = \{2, 4, 6, 8\}$, $J = \{10, 20, 30, 40, 50\}$–without variability

usage in the B&B solving scheme.

In the above results we observed a trend that indicates that for difficult instances, particularly in the case of payoff matrices with no variability, one could expect (ERASER$_{c,q,s,f}$) and (SDOBSS$_{q,y,s}$) to perform very poorly compared to (MIP-p-S$_{q,y}$). To analyze this, we consider instances where the payoff matrices have no variability and where $K = \{6, 8, 10, 12\}$,

$J = \{30, 40, 50, 60, 70\}$ and m is 25%, 50% and 75% of the targets. We generate 5 random instances for each size. In addition, for practical reasons, we consider a time limit of 30 minutes. The computational results for these instances are shown in Figure 4.3.4.

Note that (MIP-p-S$_{q,y}$) is able to solve 95% of the 300 instances within the stipulated time

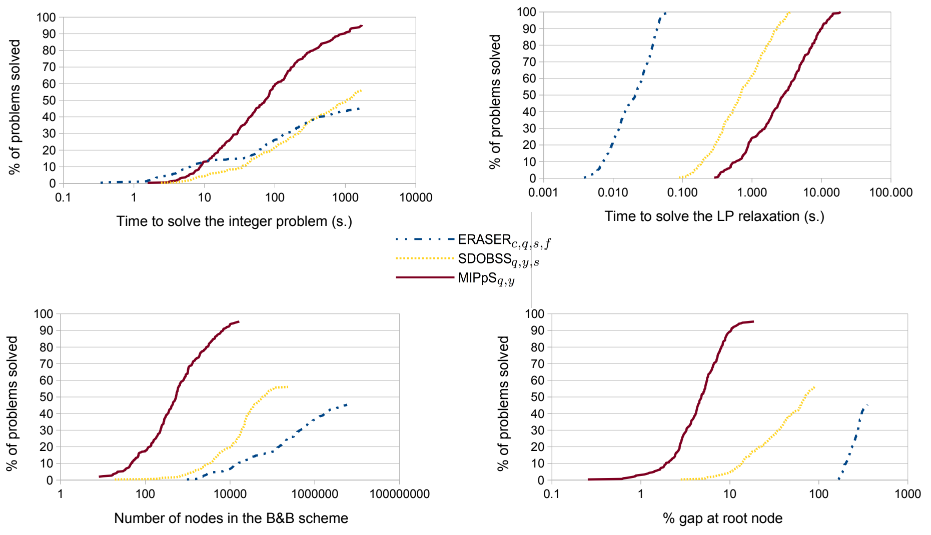


Figure 4.3.4: SSGs: $K = \{6, 8, 10, 12\}, J = \{30, 40, 50, 60, 70\}$–without variability

limit, outperforming (SDOBSS$_{q,y,s}$) and (ERASER$_{c,q,s,f}$) which are only able to solve 56% and 45% of the instances, respectively, within the same time frame. For the 45% of instances which can be solved by the three formulations, one can observe that (MIP-p-S$_{q,y}$) offers a much tighter gap percentage than the other two formulations. Because of this, the node usage in the branch and bound scheme is significantly smaller in (MIP-p-S$_{q,y}$) compared to (ERASER$_{c,q,s,f}$) and (SDOBSS$_{q,y,s}$).

Table 4.3.1 records the mean gap percentage across all the instances for the three formulations under study. Observe that $(\overline{\text{MIP-}p\text{-S}_{q,y}})$ is significantly tighter than the LP relaxations of the other formulations. Therefore, for the payoff matrices without variability, (MIP-p-

	(ERASER$_{c,q,s,f}$)	(SDOBSS$_{q,y,s}$)	(MIP-p-S$_{q,y}$)
Mean gap % (no variability)	241.26	38.87	3.09
Mean gap % (with variability)	168.37	18.66	0.35
Total mean gap %	204.82	28.76	1.72

Table 4.3.1: Mean gap percentage recorded for SSG formulations

S$_{q,y}$) is the fastest formulation for the most difficult instances. (ERASER$_{c,q,s,f}$) is the fastest formulation when the security game is endowed with further structure by allowing matrices to experience variability. Even then, (ERASER$_{c,q,s,f}$) looses ground to (MIP-p-S$_{q,y}$). This is due to the fact that (MIP-p-S$_{q,y}$) has the tightest LP relaxation. The quality of the upper bound obtained from ($\overline{\text{MIP-}p\text{-S}_{q,y}}$) translates into a smaller B&B tree and this translates into reaching optimality of the integer problem faster in many cases.

4.4 Conclusions

We have explored the effect of altering the payoff structure of the games on the computational performance of the different formulations, in both the general and the security setting. Instances that are generated with variability (by allowing 10% of the values to peak) have a less compact payoff composition and we have seen this to affect the performance of the formulations in two ways: First, it endows the games, in either setting, with more structure and allows for a faster resolution throughout and, second, it emphasizes the weakness with respect to LP bound quality of the formulations with big-M constants. In these formulations, having a few large payoff entries results in large big-M constants that account for loose LP relaxations. For such instances, the tightness achieved by the LP relaxations of the tight formulations (MIP-p-G$_{q,z}$) and (MIP-p-S$_{q,y}$) is much greater than that achieved by the LP relaxations of competing formulations.

Our computational tests have shown, in both the general and the security setting, that some formulations are more efficient with respect to running time than others depending on the sizes of the instances being solved. The weak formulations, (D2$_{x,q,s,f}$) and (ERASER$_{c,q,s,f}$), should be preferred when solving smaller instances whereas the strong formulations, (MIP-p-G$_{q,z}$) and (MIP-p-S$_{q,y}$), should be preferred when solving larger instances.

Further, the obvious bottleneck, at this time, is solving the tighter but significantly heavier LP relaxations provided by (MIP-p-G$_{q,z}$) and (MIP-p-S$_{q,y}$). The main challenge is to provide an efficient way of solving these tight formulations. The next chapter, exploits the inherent problem structure in the Stackelberg paradigm to develop decomposition and cutting plane approaches for GSG and SSG formulations.

Chapter 5

Benders decomposition methods

5.1 Introduction

In this chapter, we address the bottleneck encountered in the previous chapter, allowing for significant scaling of the instances that can be solved. The tightest formulation in each setting, (MIP-p-G$_{q,z}$) in the general setting, and (MIP-p-S$_{q,y}$) in the security setting, outperform competing formulations in terms of solution time and in terms of MIP gap percentage. However, the tightness achieved by the linear relaxations of these strong formulations is at the expense of having a large number of variables and constraints, which leads to a significant computational effort being devoted to solving the root node of said formulations. In this chapter, we exploit the inherent structure of GSGs and SSGs to develop cutting plane approaches based on Benders decomposition that efficiently obtain a tight upper bound on the optimal solution at the root node. The approaches developed allow for a significant scaling up in the GSG and SSG instances tackled, compared to the instances handled by the formulations discussed in the previous chapter.

Specifically, we embed Benders cuts from the linear relaxation of the tightest MILP formulation in each setting into a Cut and Branch scheme for the weakest MILP formulation in each setting. In a Cut and Branch scheme, we generate valid cuts exclusively at the root node of the weak formulation by using separation problems from different Benders decompositions on the LP relaxation of the tight MILP formulation. The generated cuts are then added to the weak formulation without interrupting the optimization process. We present different Benders reformulations of the LP relaxations of the tight MILP formulations, leading to different separation problems and therefore to different families of valid cuts and ultimately to different Cut and Branch approaches. For both the general and security setting, we perform detailed computational experiments to analyze the performance of the different approaches against that of alternative solution methods including the best performing MILP formulation for each setting and CPLEX automatic Benders decomposition of said MILP. We further compare the performance of the proposed Cut and Branch approaches with full Branch and Cut approaches that generate valid cuts for the weak formulations down the entire search tree.

In addition, we explore crucial implementation features such as a root-node cut loop stabilization and different primal heuristics to both warm start the proposed algorithms and to obtain improved lower and upper bounds during the solving process. We conduct a detailed study of the impact that different settings of a root-node cut loop stabilization and different primal heuristics can have on the proposed approaches. We further explore the effect on said approaches of interrupting the cut generation at the root node before the root node is solved.

Our computational results indicate that fine tuning the above implementation features improves the performance of the decomposition approaches. Further, the fine-tuned decomposition algorithms perform better than competing solution methods with respect to running time and sizes of the instances that can be tackled within a three hour computation time limit. Our decomposition algorithms allow us to tackle general instances with up to 95 5×5 payoff bimatrices or 23 10×10 payoff bimatrices. Also, our security setting decomposition algorithms allow us to solve security instances with 5 targets, 2 security resources and up to 100 attacker types as well as instances with 4 attacker types, up to 175 targets and where the number of resources is half the number of targets. The computational tests show that instances of the described sizes are beyond the scope of what competing solution methods can solve.

This chapter is organized as follows: In Section 5.2, we fully describe Bender Decomposition, which is a crucial tool used throughout this chapter. In Section 5.3, we present two different Benders reformulations for the linear relaxation of (MIP-p-G$_{x,q,z}$) and extend the work to the security setting, leading to two Benders reformulations for the linear relaxation of (MIP-p-S$_{c,q,y}$). In Section 5.4, we detail the Cut and Branch scheme on the weak formulations in each setting, (D2$_{x,q,s,f}$) in the general setting and (ERASER$_{c,q,s,f}$) in the security setting. In Section 5.5, we discuss important implementation considerations such as the root node cut loop stabilization and the use of different primal lower and upper bound heuristics to help the solver close the optimality gap faster and reduce the size of the branch and bound trees explored. In Section 5.6, we first fine tune the root-node cut generation strategy and the use of primal heuristics for the different cutting plane approaches proposed and throughly compare the performance of these approaches to that of the other solution methods considered. We conclude this chapter with some closing remarks in Section 5.7.

5.2 Benders decomposition

Benders decomposition [Benders, 1962] targets the efficient resolution of Mixed Integer Linear Programming (MILP) formulations. Consider the following general MILP formulation with n integer variables and p continuous variables

$$\begin{array}{lll} \text{P} & \text{Min}_{x,y} & c^T x + h^T y \\ & \text{s.t.} & Ax + Gy \geq b \\ & & x \in X (\subset \mathbb{N}^n) \\ & & y \in \mathbb{R}^p_+ . \end{array}$$

When the values of the x variables are fixed, the induced problem–P(x)–is a Linear Program (LP) that can be formulated as follows:

$$\begin{array}{lllll} \text{P}(x) & \text{Min}_y & h^T y & & \\ & \text{s.t.} & Gy \geq b - Ax & (\lambda) & (5.2.1) \\ & & y \geq 0, & & (5.2.2) \end{array}$$

where (λ) is the vector of dual variables associated to Constraint (5.2.1).

For the sake of clarity, an assumption is made that P always admits a finite optimal solution. This implies that for all x, either the value of P(x) is finite, denoted $v(\text{P}(x)) > -\infty$ or the problem is infeasible, denoted $v(\text{P}(x)) = +\infty$. Therefore, one need only consider the values of x for which P(x) has a solution. We denote the set of all such x by $\mathcal{R}$. By Farkas' Lemma [Farkas, 1902], one knows that the system of equations defined by (5.2.1)-(5.2.2) admits a solution if and only if for all $\mu \geq 0$ such that $\mu G \leq 0$, then $\mu(b - Ax) \leq 0$. Further, note that $\{\mu : \mu \geq 0, \mu G \leq 0\}$ is a polyhedral cone and by Minkowski-Weyl's theorem for cones, it is also a finitely generated cone. Let the finitely many generating rays be denoted by r^d for $d = 1 \ldots D$. Thus, every point μ in this cone can be expressed as $\mu = \sum_{d=1}^{D} \alpha_d r^d$ where $\alpha_d \geq 0$ for all $d = 1 \ldots D$. This can be used to rewrite the generic constraint obtained from Farkas as:

$$\sum_{d=1}^{D} \alpha_d r^d (b - Ax) \leq 0 \qquad \alpha_d \geq 0, \forall d = 1 \ldots D.$$

This condition holds if

$$r^d (b - Ax) \leq 0 \qquad \forall d = 1 \ldots D. \tag{5.2.3}$$

Further, the dual problem of P(x), which we shall denote by D(x) is given by:

$$
\begin{array}{lll}
\mathrm{D}(x) \quad \mathrm{Max}_{\lambda} & \lambda(b - Ax) & \\
\text{s.t.} & \lambda G \leq h & (5.2.4) \\
& \lambda \geq 0. & (5.2.5)
\end{array}
$$

Note that the polyhedron describing the feasible solutions of the D(x) no longer depends on x.

Therefore, since P admits a finite optimum, the value of the optimal solution of D(x), denoted $v(\mathrm{D}(x))$, with $x \in \mathcal{R}$ (recall that this implies that P(x) admits a finite optimum) is necessarily finite and $v(\mathrm{P}(x)) = v(\mathrm{D}(x))$.

It thus follows that the optimum of D(x) is attained at least at an extreme point of the polyhedron defined by (5.2.4)-(5.2.5). For simplicity, let us denote this polyhedron Q. Let $\lambda^1, \ldots, \lambda^q$ be its q extreme points. Then $v(\mathrm{D}(x))$ can be expressed as follows:

$$v(\mathrm{D}(x)) = \max_{i=1,\ldots,q} \lambda^i(b - Ax).$$

It follows that when P admits a finite optimum, $v(\mathrm{P})$ can be reformulated as follows:

$$v(P) = \min_{x \in \mathcal{R}} \left\{ c^T x + v(\mathrm{P}(x)) \right\} = \min_{x \in \mathcal{R}} \left\{ c^T x + \max_{i=1,\ldots,q} \lambda^i(b - Ax) \right\}.$$

It suffices to replace the condition $x \in \mathcal{R}$ by (5.2.3), the conditions of feasibility obtained through Farkas' Lemma. This leads to the following master problem, denoted by M:

$$
\begin{array}{llll}
\mathrm{M} \quad \mathrm{Min}_{x,\theta} & c^T x + \theta & \\
\text{s.t.} & \theta \geq \lambda^i(b - Ax) & \forall i = 1, \ldots, q \\
& r^d(b - Ax) \leq 0 & \forall d = 1, \ldots, D \\
& x \in X. &
\end{array}
$$

Note that in M, the continuous variables y no longer play a role. The master problem is an integer linear problem where the λ variables represent the extreme points and r represent the extreme rays of the polyhedron Q. Note that in M, there is one constraint per extreme point of Q and one constraint per extreme ray of Q, so potentially M has exponentially many constraints. Solving such an intractable problem directly is out of the question. The Benders approach originally considers a subset of these constraints and iteratively adds the remaining constraints until an optimal solution to P is found, hopefully without using all of the constraints in M.

Suppose that one has obtained a small subset of the constraints in M such that the following relaxed master problem, RM, is feasible.

$$\begin{array}{llll} \text{RM} & \text{Min}_{x,\theta} & c^T x + \theta & \\ & \text{s.t.} & \theta \geq \lambda^i(b - Ax) & \forall i \in I_1 (\subsetneq \{1, \ldots, q\}) \\ & & r^d(b - Ax) \leq 0 & \forall d \in I_2 (\subsetneq \{1, \ldots, D\}) \\ & & x \in X. & \end{array}$$

Solving RM yields an optimal solution: (x^*, θ^*). To determine whether such a solution satisfies all of the constraints in M, and if it doesn't, to find the constraint that is most violated by the current optimal solution, one solves the following separation problem, S:

$$\begin{array}{lll} \text{S} & \text{Max}_\lambda & \lambda(b - Ax^*) \\ & \text{s.t.} & \lambda G \leq h \\ & & \lambda \geq 0. \end{array}$$

Remark that S is nothing more than D(x^*). Thus solving S, provides $v(\text{D}(x^*))$, the value of its optimal solution and $\bar{\lambda}$, the optimal solution that attains that value. One of the three following cases may occur:

(1.) $v(\text{D}(x^*)) \leq \theta^*$. In this case, x^* is an optimal solution to M.

(2.) $v(\text{D}(x^*)) > \theta^*$. In this case, add the constraint $\theta \geq \bar{\lambda}(b - Ax)$ to RM and reoptimize.

(3.) $v(\text{D}(x^*))$ is unbounded (which makes the P(x^*) infeasible). This means that there exists an extreme ray $\bar{r}$ such that $\bar{r}(b - Ax^*) > 0$. Add $\bar{r}(b - Ax) \leq 0$ to RM and reoptimize.

The Benders procedure consists in, given a small initial set of constraints that make the relaxed master problem feasible, iteratively solving this relaxed master problem by adding violated constraints, identified by solving a separation problem, until an optimal solution is obtained.

5.3 Decomposition approaches

In this section, we describe the different Benders decompositions for the linear relaxations of (MIP-p-G$_{x,q,z}$) and (MIP-p-S$_{c,q,y}$). First, recall that (MIP-p-G$_{q,z}$) is given by

$$\begin{array}{lllll} (\text{MIP-}p\text{-G}_{q,z}) & \text{Max} & \sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \pi^k R^k_{ij} z^k_{ij} & & \\ & \text{s.t.} & \sum_{j \in J} q^k_j = 1 & \forall k \in K, & (5.3.1) \\ & & q^k_j \in \{0, 1\} & \forall j \in J, \forall k \in K & (5.3.2) \end{array}$$

$$\sum_{j\in J} z_{ij}^k = \sum_{j\in J} z_{ij}^1 \qquad \forall i \in I, \forall k \in K, \tag{5.3.3}$$

$$\sum_{i\in I} z_{ij}^k = q_j^k \qquad \forall j \in J, \forall k \in K, \tag{5.3.4}$$

$$z_{ij}^k \geq 0 \qquad \forall i \in I, \forall j \in J, \forall k \in K, \tag{5.3.5}$$

$$\sum_{i\in I} (C_{ij}^k - C_{i\ell}^k) z_{ij}^k \geq 0 \qquad \forall \ell, j \in J, \forall k \in K. \tag{5.3.6}$$

Note that the x variables can be introduced in (MIP-p-G$_{q,z}$), leading to (MIP-p-G$_{x,q,z}$), by rewriting Constraint (5.3.3) as:

$$\sum_{j\in J} z_{ij}^k = x_i \qquad \forall i \in I, \forall k \in K. \tag{5.3.7}$$

Further, recall that (MIP-p-S$_{q,y}$) is given by

(MIP-p-S$_{q,y}$)

$$\text{Max} \quad \sum_{k\in K}\sum_{j\in J} \pi^k (D^k(j|c) y_{jj}^k + D^k(j|u)(q_j^k - y_{jj}^k))$$

$$\text{s.t.} \quad \sum_{j\in J} q_j^k = 1 \qquad \forall k \in K, \tag{5.3.8}$$

$$q_j^k \in \{0,1\} \qquad \forall j \in J, \forall k \in K \tag{5.3.9}$$

$$\sum_{j\in J} y_{\ell j}^k = \sum_{j\in J} y_{\ell j}^1 \qquad \forall \ell \in J, \forall k \in K, \tag{5.3.10}$$

$$\sum_{\ell\in J} y_{\ell j}^k \leq m q_j^k \qquad \forall j \in J, \forall k \in K, \tag{5.3.11}$$

$$0 \leq y_{\ell j}^k \leq q_j^k \qquad \forall \ell, j \in J, \forall k \in K, \tag{5.3.12}$$

$$\begin{aligned} &A^k(j|c) y_{jj}^k + A^k(j|u)(q_j^k - y_{jj}^k) - \\ &\quad A^k(\ell|c) y_{\ell j}^k - A_{\ell|u}^k (q_j^k - y_{\ell j}^k) \geq 0 \qquad \forall \ell, j \in J, \forall k \in K. \end{aligned} \tag{5.3.13}$$

Similarly, the c variables can be introduced in (MIP-p-S$_{q,y}$), leading to (MIP-p-S$_{c,q,y}$), by rewriting Constraint (5.3.10) as:

$$\sum_{j\in J} y_{\ell j}^k = c_\ell \qquad \forall \ell \in J, \forall k \in K. \tag{5.3.14}$$

We later use the separation problems from the decompositions shown next and the valid cuts they separate to develop cutting plane approaches to strengthen the linear relaxations of the weaker equivalent formulations (D2$_{x,q,s,f}$) and (ERASER$_{c,q,s,f}$), respectively.

5.3.1 General Stackelberg games

We follow the Benders decomposition steps described in Section 5.2 to develop two different decompositions on the linear relaxation of the tight GSG formulation (MIP-p-G$_{x,q,z}$).

First Benders approach–x and q variables remain in the master problem

In this first decomposition of ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$), the x and q variables remain as master problem variables and the z variables are sent to $|K|$ induced linear problems, $\mathrm{P}_k(x, q^k)$ for $k \in K$. The duals of these LPs are referred to as the separation problems and denoted $\mathrm{D}_k(x, q^k)$ for $k \in K$. The k-th separation problem is given by

$$\begin{aligned}
\mathrm{D}_k(x,q^k) \quad & \mathrm{Min}_{\alpha^k,\beta^k,\gamma^k} && \sum_{j\in J}\alpha_j^k q_j^k + \sum_{i\in I}\beta_i^k x_i && \\
& \text{s.t.} && \alpha_j^k + \beta_i^k + \sum_{\ell\in J:\ell\neq j}(C_{i\ell}^k - C_{ij}^k)\gamma_{\ell j}^k \geq R_{ij}^k && \forall i\in I, \forall j\in J, \quad (5.3.15)\\
& && \gamma_{\ell j}^k \geq 0 && \forall \ell, j\in J: \ell\neq j. \quad (5.3.16)
\end{aligned}$$

The master problem, which is equivalent to ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$), is given by

$$\begin{aligned}
\text{(MG1)} \quad & \mathrm{Max}_{c,q} && \sum_{k\in K}\pi^k\theta^k && \\
& \text{s.t.} && \theta^k \leq \sum_{j\in J}\alpha_j^{kg}q_j^k + \sum_{i\in I}\beta_i^{kg}x_i && \forall g = 1,\ldots,G_k, \forall k\in K, \quad (5.3.17)\\
& && \sum_{j\in J}s_j^{kh}q_j^k + \sum_{i\in I}t_i^{kh}x_i \geq 0 && \forall h = 1,\ldots,H_k, \forall k\in K, \quad (5.3.18)\\
& && \sum_{j\in J}q_j^k = 1 && \forall k\in K,\\
& && q_j^k \geq 0 && \forall j\in J, \forall k\in K,
\end{aligned}$$

where for each $k \in K$, G_k and H_k are the number of extreme points and extreme rays, respectively, of the polyhedron that defines the feasible region of $\mathrm{D}^k(x, q^k)$. Constraints (5.3.17) impose a bound on the objective function of the master problem and are known as *optimality cuts*. Constraints (5.3.18) guarantee feasibility of the primal subproblems $\mathrm{P}_k(x, q^k)$–the dual of $\mathrm{D}^k(x, q^k)$–and are therefore known as *feasibility cuts*. Note that in (MG1), the continuous variables z no longer play a part.

Second Benders approach–x variables remain in the master problem

The second Benders reformulation we look at is inspired by the work in [Yin and Tambe, 2012], where only the x variables are left as master variables. They add the following valid (redundant) inequalities to ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$):

$$\sum_{i\in I} x_i = 1, \qquad (5.3.19)$$

$$x \geq 0. \qquad (5.3.20)$$

In this case, the k-th separation problem is given by

$$\begin{aligned}
\mathrm{D}_k(x) \quad & \mathrm{Min}_{\alpha^k,\beta^k,\gamma^k,\delta^k} && \alpha^k + \sum_{i \in I} \gamma_i^k x_i && \\
& \text{s.t.} && \alpha^k - \beta_j^k \geq 0 && \forall j \in J, \quad (5.3.21)\\
& && \beta_j^k + \delta_i^k + \sum_{\ell \in J: \ell \neq j} \delta_{\ell j}^k (C_{i\ell}^k - C_{ij}^k) \geq R_{ij}^k && \forall i \in I, \forall j \in J, \quad (5.3.22)\\
& && \delta_{\ell j}^k \geq 0 && \forall \ell, j \in J : \ell \neq j, \quad (5.3.23)
\end{aligned}$$

and the master problem, which is equivalent to ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$), is given by

$$\begin{aligned}
(\text{MG2}) \quad & \text{Max} && \sum_{k \in K} \pi^k \theta^k && \\
& \text{s.t.} && \theta^k \leq \alpha^{kh} + \sum_{i \in I} \gamma_i^{kh} x_i && \forall h \in 1, \dots, H_k, \forall k \in K, \quad (5.3.24)\\
& && (5.3.19) - (5.3.20), &&
\end{aligned}$$

where for each $k \in K$, H_k is the number of extreme points of the polyhedron that defines the feasible region of $\mathrm{D}_k(x)$. Constraints (5.3.24) are the optimality cuts in this case and impose a bound on the value of the master problem's objective function. In this setting, it can be seen that no feasibility cuts are needed, as for each $k \in K$, the corresponding induced linear program $\mathrm{P}_k(x)$, obtained from fixing the value of the x variables in ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$) and separating by follower type $k \in K$, is always feasible given an x satisfying (5.3.19) and (5.3.20).

5.3.2 Stackelberg security games

One can easily derive analogous Benders decompositions for the linear relaxation of the tight SSG formulation (MIP-p-S$_{c,q,y}$) following the same steps described in Section 5.2.

First Benders approach–c and q variables remain in the master problem

A Benders decomposition of ($\overline{\text{MIP-}p\text{-S}_{c,q,y}}$) where the c and q variables remain in the master and the y variables are sent to $|K|$ induced linear problems, $\mathrm{P}_k(c, q^k)$ for $k \in K$, can easily be derived. In this case, the k-th separation problem, denoted by $\mathrm{D}_k(c, q^k)$, is given by

$$\begin{aligned}
& \mathrm{D}_k(c, q^k) && \\
& \mathrm{Min}_{\xi^k,\alpha^k,\beta^k,\gamma^k} && \sum_{\ell \in J} \xi_\ell^k c_\ell + m \sum_{j \in J} \alpha_j^k q_j^k + \sum_{\ell \in J} \sum_{j \in J} \left\{ \beta_{\ell j}^k q_j^k + (A^k(j|u) - A^k(\ell|u)) \gamma_{\ell j}^k q_j^k \right\} \\
& \text{s.t.} && \xi_\ell^k + \alpha_j^k + \beta_{\ell j}^k + (A^k(\ell|c) - A^k(\ell|u)) \gamma_{\ell j}^k \geq 0 \ \forall \ell, j \in J : \ell \neq j, \quad (5.3.25)\\
& && \xi_j^k + \alpha_j^k + \beta_{jj}^k +
\end{aligned}$$

$$(A^k(j|u) - A^k(j|c)) \sum_{\ell \in J: \ell \neq j} \gamma^k_{\ell j} \geq \pi^k (D^k(j|c) - D^k(j|u)) \;\; \forall j \in J, \tag{5.3.26}$$

$$\alpha^k, \beta^k, \gamma^k \geq 0. \tag{5.3.27}$$

The master problem is given by:

$$\begin{aligned}
\text{(MS1)} \quad & \text{Max}_{c,q} \quad \sum_{k \in K} \pi^k \theta^k \\
\text{s.t.} \quad & \theta^k \leq \sum_{j \in J} D^k(j|u) q^k_j + \sum_{\ell \in J} \xi^{ki}_\ell c_\ell + m \sum_{j \in J} \alpha^{ki}_j q^k_j + \\
& \quad \sum_{\ell \in J} \sum_{j \in J} \left\{ \beta^{ki}_{\ell j} q^k_j + (A^k(j|u) - A^k(\ell|u)) \gamma^{ki}_{\ell j} q^k_j \right\} \qquad \forall i = 1, \ldots, t_k, \forall k \in K, \qquad (5.3.28) \\
& \sum_{\ell \in J} s^{kh}_\ell c_\ell + m \sum_{j \in J} q^k_j t^{kh}_j + \\
& \quad \sum_{j \in J} \sum_{\ell \in J} \left\{ u^{kh}_{\ell j} q^k_j + (A^k(j|u) - A^k(\ell|u)) v^{kh}_{\ell j} q^k_j \right\} \geq 0 \quad \forall h = 1, \ldots, H_k, \forall k \in K, \qquad (5.3.29) \\
& \sum_{j \in J} q^k_j = 1 \qquad \forall k \in K, \\
& q^k_j \geq 0 \qquad \forall j \in J, \forall k \in K,
\end{aligned}$$

where for each $k \in K$, t_k and H_k are the numbers of vertices and extreme rays, respectively, of the polyhedron that defines the feasible region of $\mathrm{D}_k(c, q^k)$. Constraints (5.3.28) are thus the optimality cuts and (5.3.29), the feasibility cuts.

Second Benders approach–c variables remain in the master problem

Alternatively, a Benders decomposition of ($\overline{\text{MIP-}p\text{-S}_{c,q,y}}$) can be developed where only the c variables are left as master variables. The following valid (redundant) inequalities are added to ($\overline{\text{MIP-}p\text{-S}_{c,q,y}}$):

$$\sum_{j \in J} c_j \leq m, \tag{5.3.30}$$

$$c \in [0,1]^{|J|}. \tag{5.3.31}$$

In this case the k-th separation problem, denoted by $\mathrm{D}_k(c)$ is given by

$$\begin{aligned}
\mathrm{D}_k(c) \quad & \text{Min}_{\alpha^k, \beta^k, \gamma^k, \delta^k, \lambda^k, \eta^k} \quad \sum_{\ell \in J} \delta^k_\ell c_\ell + \lambda^k \\
\text{s.t.} \quad & \alpha^k_j \geq 1 \qquad \forall j \in J, \qquad (5.3.32) \\
& \beta^k, \gamma^k, \eta^k \geq 0, \qquad (5.3.33) \\
& -\alpha^k_j D^k(j|u) - m\beta^k_j -
\end{aligned}$$

$$\sum_{\ell \in J} \gamma_{\ell j}^k + \lambda^k + \sum_{\ell \in J} (A^k(\ell|u) - A^k(j|u))\eta_{\ell j}^k \geq 0 \qquad \forall j \in J, \tag{5.3.34}$$

$$\alpha_j^k (D^k(j|u) - D^k(j|c)) + \beta_j^k + \tag{5.3.35}$$

$$\gamma_{jj}^k + \delta_j^k + \sum_{\ell \in J : \ell \neq j} \eta_{\ell j}^k (A^k(j|u) - A^k(j|c)) \geq 0 \qquad \forall j \in J, \tag{5.3.36}$$

$$\beta_j^k + \gamma_{\ell j}^k + \delta_\ell^k + (A^k(\ell|c) - A^k(\ell|u))\eta_{\ell j}^k \geq 0 \qquad \forall \ell, j \in J : \ell \neq j. \tag{5.3.37}$$

The master problem is given by:

$$\begin{aligned} \text{(M2)} \quad \text{Max}_c \quad & \sum_{k \in K} \pi^k \theta^k \\ \text{s.t.} \quad & \theta^k \leq \lambda^{ki} + \sum_{\ell \in J} \delta_\ell^{ki} c_\ell \qquad \forall i \in I_k, \forall k \in K. && (5.3.38) \\ & (5.3.30) - (5.3.31), \end{aligned}$$

where for $k \in K$, I_k is the set of extreme points of the polyhedron which defines the feasible region of $\mathrm{D}_k(c)$.

5.4 Cutting plane approach

In this section we propose a cutting plane scheme for the general and the security setting. The approach proposed consists in strengthening the LP relaxation of the weaker MILP formulations, ($\mathrm{D2}_{x,q,s,f}$) in the general setting and ($\mathrm{ERASER}_{c,q,s,f}$) in the security setting, by using the optimality and/or feasibility cuts obtained from the Benders decompositions on the LP relaxations of (MIP-p-$\mathrm{G}_{x,q,z}$) and (MIP-p-$\mathrm{S}_{c,q,y}$). The separation problems described in the previous section are used, in Cut and Branch mode, to identify the best of these valid cuts iteratively.

Given that ($\mathrm{D2}_{x,q,s,f}$) and (MIP-p-$\mathrm{G}_{x,q,z}$) are equivalent MILP formulations for the general Stackelberg game, the cuts of type (5.3.17), (5.3.18) and (5.3.24) are valid for ($\mathrm{D2}_{x,q,s,f}$) and cut the polyhedron of its LP relaxation. Similarly, because ($\mathrm{ERASER}_{c,q,s,f}$) and (MIP-$p$-$\mathrm{S}_{c,q,y}$) are equivalent formulations for the Stackelberg security game, the cuts of type (5.3.28), (5.3.29) and (5.3.38) are valid for ($\mathrm{ERASER}_{c,q,s,f}$) and cut the polyhedron of its LP relaxation.

The goal is to strengthen the LP relaxations of ($\mathrm{D2}_{x,q,s,f}$) and ($\mathrm{ERASER}_{c,q,s,f}$) with a limited number of Benders cuts. The resulting strengthened formulations, ($\mathrm{D2}_{x,q,s,f}$)+{valid cuts} and ($\mathrm{ERASER}_{c,q,sf}$)+{valid cuts}, will have the same tight bound as the LP relaxations of (MIP-p-$\mathrm{G}_{q,z}$) and (MIP-p-$\mathrm{S}_{q,y}$), respectively, but will be significantly sparser than these in terms of variables and constraints leading to an improved performance of the approach proposed over the tightest formulation in each setting.

Next, we present the cut and branch approach on the weak formulation, as implemented in CPLEX [CPL, 2017] in a generic fashion and describe how to make it specific to the different formulations and separation problems that we have described up to now. The term MASTER in the following algorithm refers to the weak formulation, *i.e.*, ($\mathrm{D2}_{x,q,s,f}$) when in the general case and ($\mathrm{ERASER}_{c,q,s,f}$) when in the security case.

```
{Input: Game data (Payoff matrices, probability distribution $\pi$, # of strategies for each player, # of follower/attacker types)};
{initialization};
UB:= $+\infty$;
DualLB:= $-\infty$;
PrimalLB:= $CPLEX.Incumbent$;
{Commence CPLEX solve on MASTER};
while $UB$-$PrimalLB > \varepsilon UB$ do
    {Continue CPLEX solve on MASTER};
    {Identify optimal MASTER (fractional) solution $a_{\text{out}}$};
    {Update UB = $v$(MASTER)};
    if $Node = ROOT$ then
        while $UB$-$DualLB > \varepsilon UB$ do
            {Feed solution $a_{\text{out}}$ into $|K|$ separation subproblems};
            for $k \in K$ do
                {Solve $k$-th separation problem};
                if $k$-th separation problem is unbounded then
                    {Generate feasibility cut};
                else
                    {Generate optimality cut};
                end
            end
            if $\exists$ generated cuts which are violated by $a_{out}$ then
                {ADD cuts to MASTER problem};
            end
            {Update DualLB through objective values of the separation subproblems};
        end
    end
end
{Finish CPLEX solve};
return integer solution;
```

Algorithm 2: Cut and Branch approach on the weak formulations

Algorithm 2 leads to the four different approaches we study in this work depending on the formulation chosen as MASTER, and on the chosen separation problems. For the general setting, if MASTER is ($\mathrm{D2}_{x,q,s,f}$) and the separation problems are $\{D_k(x, q^k)\}_{k\in K}$, we name the approach ($\mathrm{C\&B}_{x,q}$). If MASTER is ($\mathrm{D2}_{x,q,s,f}$) and the separation problems are $\{D_k(x)\}_{k\in K}$, we name the approach ($\mathrm{C\&B}_{x}$). Analogously, in the security setting, MASTER is ($\mathrm{ERASER}_{c,q,s,f}$) and the approach will either be ($\mathrm{C\&B}_{c,q}$) or ($\mathrm{C\&B}_{c}$) depending on whether the separation problems are $\{D_k(c, q^k)\}_{k\in K}$ or $\{D_k(c)\}_{k\in K}$. In order to add violated cuts to MASTER within CPLEX's internal solving procedure, one uses CPLEX *user callback functions*.

5.5 Implementation considerations

In this section, we discuss important implementation considerations for the cutting plane approaches. Specifically, we describe in detail a cut loop stabilization procedure at the root node, much in the spirit of that shown in [Fischetti et al., 2016b], designed to quickly close the gap between the upper bound and the dual lower bound and thus reduce the number of root node iterations. By reducing the number of cut generating iterations, one reduces the number of cuts and thus produces sparser formulations for CPLEX to continue solving after the root node.

In addition, we discuss several primal heuristics designed to enhance the lower bound during the solving process, thus pruning the branch and bound tree explored by each formulation. We further propose a very simple upper bound heuristic to aid in closing the optimality gap. In Section 5.6, we discuss the effects that these add-ons have on the performance of the proposed approaches and whether a profitable configuration of the parameters involved leads to an improved performance of the method.

5.5.1 Root node cut loop stabilization

The purpose of a root node cut loop stabilization procedure is to reduce the number of times we have to solve the separation problems before we conclude solving the root node, *i.e.*, the stabilization procedure ensures a quicker closing of the gap between the upper bound and the dual lower bound.

The cut generating scheme shown in lines 7-28 in Algorithm 2 is referred to as the classical Kelley scheme, [Kelley, 1960], which is known to have a very bad performance. The reason for this bad performance is that the dual lower bound calculated at each iteration can be very erratic and as a result, the process has a somewhat slow convergence. The authors in [Fischetti et al., 2016b] point out that the convergence behavior of the overall cut strategy greatly depends on the point chosen to be separated at each iteration and, as such, the performance of this cut loop can be easily improved by implementing a simple in-out stabilization procedure like the one shown in [Ben-Ameur and Neto, 2007].

In general, at a given root node iteration, suppose one has a (fractional) solution, a_{out}, obtained from solving the master problem, and a feasible solution for the original MILP, a_{in}. Then, at that iteration, rather than attempting to separate a_{out}, one can separate instead the following 'intermediate' point:

$$a_{\text{sep}} = \lambda a_{\text{out}} + (1-\lambda)a_{\text{in}},$$

where $\lambda \in]0,1]$. Note that when $\lambda = 1$, this amounts to not performing any stabilization, and the lower the value of λ, the more aggressive the stabilization performed becomes, in the sense that the point that one feeds into the separation problems is closer to a feasible solution. Algorithm 3 shows pseudocode for this stabilization procedure.

```
{Input: Current (fractional) solution $a_{\text{out}}$, Feasible solution $a_{\text{in}}$};
{Set value of $\lambda \in ]0,1]$};
{Construct $a_{\text{sep}} = \lambda a_{\text{out}} + (1-\lambda)a_{\text{in}}$};
{Feed $a_{\text{sep}}$ into the $|K|$ separation subproblems};
for $k \in K$ do
    {Solve $k$-th separation problem};
    if k-th separation problem is unbounded then
        {Generate feasibility cut};
    else
        {Generate optimality cut};
    end
    if $\exists$ generated cut which is violated by $a_{sep}$ then
        {ADD cut to MASTER problem};
    end
end
if No cuts have been added then
    {Update 'interior' point: $a_{\text{in}} = a_{\text{sep}}$};
end
```

Algorithm 3: Cut loop stabilization procedure

Note that it is important to adequately tune the value of λ for a good performance of the stabilization procedure. No stabilization, as in Kelley's scheme, can lead to slow convergence but an over-aggressive stabilization can lead to too much time consumed on solving separation subproblems which do not generate violated cuts. This tuning is further studied in Section 5.6.

In the meanwhile, let us present an example to show two things. First, the smoothing effect that in-out stabilization can produce on the lower bound compared to the classical Kelley's scheme. Second, the significant improvement with respect to solution time and sparseness of the resulting formulation that can be achieved by stabilizing. These two facts suggest that a fine-tuned cut loop stabilization could indeed be a very useful asset in enhancing the performance of the cutting plane approaches.

Consider a randomly generated security instance where $K = 12$, $J = 45$ and $m = 22$. Penalties for the players are uniformly generated between 0 and 5 and rewards between 5 and 10. The probability distribution over the different follower types is also uniformly randomly generated. Further, compare solving this SSG instance with an unstabilized (C&B$_{c,q}$) to solving the same instance with the same approach but with moderate stabilization ($\lambda = 0.5$). Figure 5.5.1, shows the upper and lower bound behavior at the root node without stabilization (on the left) and with a stabilization of $\lambda = 0.5$ (on the right). Note

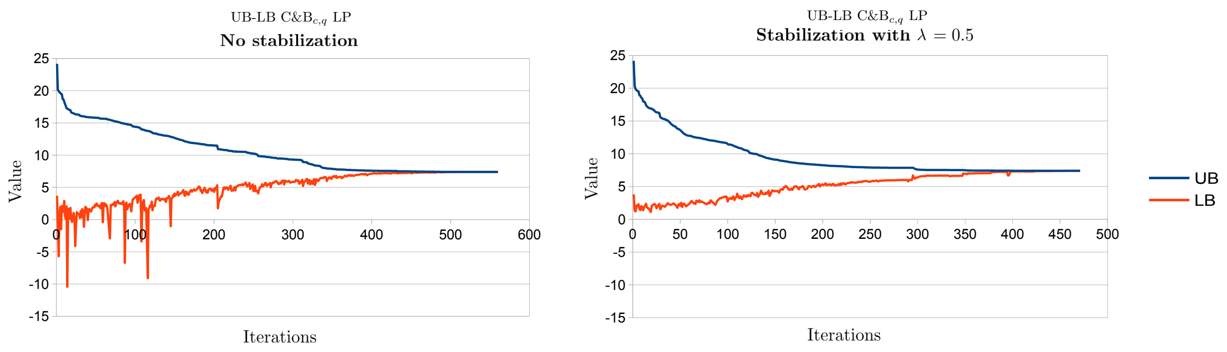


Figure 5.5.1: Upper bound-lower bound behavior at the root node

how erratic the lower bound can be when there is no stabilization involved and how this behavior becomes much smoother if one allows for some stabilization. Further, consider the information from the solving procedure shown in Table 5.5.1. Note how the number

	Cuts	Iter.	Nodes	IP time	LP time	Tot. time
$\lambda = 1$	5391	560	105559	2158.95	1138.00	3296.94
$\lambda = 0.5$	3455	471	74259	636.64	877.10	1513.74

Table 5.5.1: Unstabilized (C&B$_{c,q}$) vs. stabilized ($\lambda = 0.5$) (C&B$_{c,q}$) on an SSG instance

of cuts is dramatically reduced if one stabilizes and how this reduces the time it takes to solve the linear relaxation. In this example, the formulation one obtains when stabilizing is significantly 'thinner' as far fewer cuts are added and this results, in this case, in a reduction of more than 50% in the solution time of the integer problem. Another interesting implementation feature we analyze in the computational experiments section is the effect on the different cutting plane solution methods of interrupting the cut generation before the root node is solved.

5.5.2 Primal lower bound-enhancing heuristics

In this section, we analyze several primal lower bound heuristics which can be used with a twofold purpose, first, to obtain a feasible solution which can be used, not only in the stabilization procedure described in the previous section, but also to warm-start the optimization process and second, to provide tight lower bounds which can lead to an efficient pruning of the branch and bound tree and in turn to a better overall performance of the resolution method. Since the heuristics discussed are specific to either the general or the security setting, we will discuss them separately. We first detail the heuristics for GSGs and then extend the discussion to SSGs.

General Stackelberg games

The first heuristic designed to produce an initial feasible solution appears in [Paruchuri et al., 2007]. This heuristic is based on limiting the possible leader mixed strategies one optimizes over. The authors in [Paruchuri et al., 2007] limit the feasible mixed strategies by considering what are known as s-uniform strategies; strategies where the probability of playing a given pure strategy is a multiple of $\frac{1}{s}$, for some integer s. In their paper, they solve for the optimal leader s-uniform strategy by solving a (DOBSS)-like formulation, where for all $i \in I$, the x_i variables are forced to take values in the set $\{0, \frac{1}{s}, \frac{2}{s}, \dots, 1\}$ through the use of auxiliary integer variables $r_i \in \{0, 1, \dots, s\}$ for all $i \in I$ such that $sx_i = r_i$. Based on the theoretical and computational results on the efficiency of GSG MILP formulations shown in Chapter 4, the efficiency of this heuristic can be enhanced by using (MIP-p-G$_{x,q,z}$) as a base model instead of using (DOBSS$_{q,z,s}$). We refer to this MILP as (MIP-p-G_H). The MILP (MIP-p-G_H) returns an s-uniform mixed strategy for the leader which is feasible in the GSG and as such it provides a lower bound on the optimum value.

We propose two additional primal lower bound heuristics. The first heuristic we propose–which we call the 'basic' heuristic–works as follows: given a leader's mixed strategy $\bar{x}$, retrieved from the optimal solution of the master problem, one computes each follower's best response to $\bar{x}$. To do so, one first computes each follower's expected utility when committing to each pure strategy $j \in J$:

$$\mathrm{FEU}_j^k = \sum_{i \in I} C_{ij}^k \bar{x}_i.$$

Further one identifies, for each follower type $k \in K$, the pure strategy $j \in J$ for that follower type, denoted $j(k)$, which maximizes that follower's expected utility over all the pure strategies $j \in J$:

$$j(k) = \arg\max_{j \in J} \mathrm{FEU}_j^k.$$

One then constructs the follower's strategy q^* such that $q_j^{k*} = 1$ for $j \in J : j = j(k)$ and $q_j^{k*} = 0$ for $j \in J : j \neq j(k)$. Then $(\bar{x}, q^*)$ is a feasible solution to the integer problem with objective value $\sum_{k \in K} \sum_{i \in I} \sum_{j \in J} \pi^k R_{ij}^k \bar{x}_i q_j^{k*}$.

The second heuristic we propose–which we refer to as the 'advanced' heuristic–is an enhancement over the 'basic' heuristic just described. The 'basic' heuristic returns q^*, the follower's best response to a given leader's mixed strategy $\bar{x}$. The 'advanced' heuristic then optimizes the leader's strategy over all mixed strategies for which q^* is a best response by solving the following LP:

$$\text{(LP\_G)} \quad \text{Max} \quad \sum_{i \in I} \sum_{k \in K} \pi^k R^k_{ij(k)} x_i$$

$$\text{s.t.} \quad \sum_{i \in I} x_i = 1, \tag{5.5.1}$$

$$x_i \geq 0 \qquad \forall i \in I, \tag{5.5.2}$$

$$\sum_{i \in I} C^k_{ij(k)} x_i \geq \sum_{i \in I} C^k_{ij} x_i \qquad \forall k \in K, \forall j \in J : j \neq j(k). \tag{5.5.3}$$

Note that this LP will never be infeasible, since $(\bar{x}, q^*)$ is feasible for the LP by construction of q^* as a best response to $\bar{x}$. In Algorithm 4 we show the pseudocode for the advanced primal heuristic for $(\text{D2}_{x,q,s,f})$ just described.

```
{Input: Fractional solution from MASTER problem $(\bar{x}, \bar{q}, \bar{s}, \bar{d})$, Incumbent solution};
if iteration %( $p \in \mathbb{Z}_+$) then
    {Compute $q^*$, the follower's best response to $\bar{x}$};
    for $k \in K$ do
        for $j \in J$ do
            $FEU^k_j = \sum_{i \in I} C^k_{ij} \bar{x}_i$;
        end
    end
    {Identify for each $k \in K$, $j(k) = \arg\max_{j \in J} FEU^k_j$};
    {Construct $q^*$};
    for $k \in K$ do
        $q^{k*}_j = 1$ for $j \in J : j = j(k)$, $q^k_j = 0$ otherwise;
    end
    {Compute $\hat{x}$, leader's optimal mixed strategy for which $q^*$ is a best response by solving (LP_G)};
    {Compute $\hat{s}, \hat{d}$};
    for $k \in K$ do
        $\hat{s}^k = \sum_{i \in I} C^k_{ij(k)} \hat{x}_i$,   $\hat{d}^k = \sum_{i \in I} R^k_{ij(k)} \hat{x}_i$;
    end
    {$(\hat{x}, q^*, \hat{s}, \hat{d})$ is a feasible solution for (D2) with objective value $\sum_{k \in K} \pi^k \hat{d}_k$};
    if $\sum_{k \in K} \pi_k \hat{d}_k > value(incumbent)$ then
        {Update incumbent solution to $(\hat{x}, q^*, \hat{s}, \hat{d})$ and update $value(incumbent)$};
    end
end
```

Algorithm 4: 'Advanced' primal heuristic for $(\text{D2}_{x,q,s,f})$

The 'basic' and 'advanced' heuristics can be used to obtain an initial feasible point with which to warm start the optimization process by setting an initial uniform leader mixed strategy to which the heuristics can compute each follower's best response. Additionally, the heuristics can be used in the cut and branch process–and it is here that they will prove to be most beneficial–to iteratively improve the incumbent lower bound by computing follower best responses to the fractional solutions returned by MASTER.

Stackelberg security games

One can adapt the heuristic described in [Paruchuri et al., 2007] to the security setting in order to obtain an initial feasible solution when solving an SSG. In this case, one looks for an optimal coverage vector c. The heuristic proposed by [Paruchuri et al., 2007] discretizes the search space such that one need only consider coverage probabilities over targets that are a multiple of $\frac{1}{s}$ for some positive integer s. We thus consider (MIP-p-S_H), based on (MIP-p-$\text{S}_{c,q,y}$), to solve for an optimal vector c such that for each $j \in J$, c_j only takes values in $\{0, \frac{1}{s}, \ldots, 1\}$. (MIP-$p$-S_H) achieves this by including integer variables $r_j = sc_j$ for all $j \in J$ such that $r_j \in \{0, 1, \ldots, s\}$.

As in the previous section, we provide a 'basic' and 'advanced' heuristic for SSGs. The 'basic' heuristic takes as input a coverage strategy $\bar{c}$, retrieved from the optimal solution to the master problem, and computes q^*, the attacker's best response. The heuristic consists in first computing each attacker's expected utility when attacking each target $j \in J$:

$$AEU_j^k = A^k(j|c)\bar{c}_j + A^k(j|u)(1 - \bar{c}_j).$$

Further, for each attacker type $k \in K$, the heuristic identifies the target $j \in J$ for that attacker type, denoted $j(k)$, which maximizes that attacker's expected utility over all targets $j \in J$:

$$j(k) = \arg\max_{j \in J} AEU_j^k,$$

to construct an attacker's strategy q^* such that $q_j^{k*} = 1$ for $j \in J : j = j(k)$ and q_j^{k*} for $j \in J : j \neq j(k)$. Then, $(\bar{c}, q^*)$ is a feasible solution to the SSG with objective value $\sum_{k \in K} \sum_{j \in J} q_j^{k*} \pi^k (D^k(j|c)\bar{c}_j + D^k(j|u)(1 - \bar{c}_j))$.

The 'advanced' heuristic further optimizes over all the defender's coverage strategies for which q^* is a best response by solving the following LP:

(LP_S)

$$\begin{aligned}
\text{Max} \quad & \sum_{k \in K} \pi^k (D^k(j(k)|c)c_{j(k)} + \\
& \qquad D^k(j(k)|u)(1 - c_{j(k)})) \\
\text{s.t.} \quad & \sum_{j \in J} c_j \leq m, && (5.5.4) \\
& c_j \in [0,1] \qquad \forall j \in J, && (5.5.5) \\
& A^k(j(k)|c)c_{j(k)} + A^k(j(k)|u)(1 - c_{j(k)}) \geq \\
& \qquad A^k(j|c)c_j + A^k(j|u)(1 - c_j) \qquad \forall k \in K, \forall j \in J : j \neq j(k). && (5.5.6)
\end{aligned}$$

Also, note that (LP_S) will never be infeasible, since $(\bar{c}, q^*)$ is feasible for (LP_S) by construction of q^* as a best response to $\bar{c}$. Algorithm 5 provides pseudocode for the advanced primal heuristic.

```
{Input: Fractional solution from MASTER problem $(\bar{c}, \bar{q}, \bar{s}, \bar{d})$, Incumbent solution};
if $iteration\ \%(\ p \in \mathbb{Z}_+)$ then
    {Compute $q^*$, the attackers's best response to $\bar{x}$};
    for $k \in K$ do
        for $j \in J$ do
            $AEU_j^k = A^k(j|c)\bar{c}_j + A^k(j|u)(1 - \bar{c}_j)$;
        end
    end
    {Identify for each $k \in K$, $j(k) = \arg\max_{j \in J} AEU_j^k$};
    {Construct $q^*$};
    for $k \in K$ do
        $q_j^{k*} = 1$ for $j \in J : j = j(k)$, $q_j^k = 0$ otherwise;
    end
    {Compute $\hat{c}$, defender's optimal mixed strategy for which $q^*$ is a best response by solving (LP_S)};
    {Compute $\hat{s}, \hat{d}$};
    for $k \in K$ do
        $\hat{s}^k = A^k(j(k)|c)\hat{c}_{j(k)} + A^k(j(k)|u)(1 - \hat{c}_{j(k)})$,   $\hat{d}^k = D^k(j(k)|c)\hat{c}_{j(k)} + D^k(j(k)|u)(1 - \hat{c}_{j(k)})$;
    end
    {$(\hat{c}, q^*, \hat{s}, \hat{d})$ is a feasible solution for (ERASER) with objective value $\sum_{k \in K} \pi^k \hat{d}_k$};
    if $\sum_{k \in K} \pi_k \hat{d}_k > value(incumbent)$ then
        {Update incumbent solution to $(\hat{c}, q^*, \hat{s}, \hat{d})$ and update $value(incumbent)$};
    end
end
```

Algorithm 5: 'Advanced' primal heuristic for $(\text{ERASER}_{c,q,s,f})$

The 'basic' and 'advanced' heuristics can be used to obtain an initial feasible point with which to warm start the optimization process by setting an initial uniform defender coverage strategy to which the heuristics can compute each attacker's best response. As in the general case, the heuristics can be used in the cut and branch process to iteratively improve the incumbent lower bound by computing attacker best responses to the fractional solutions returned by MASTER.

5.5.3 Primal upper bound-enhancing heuristics

Even if one succeeds in quickly improving the quality of the lower bound, the efficiency of the cutting plane approaches is constrained, in addition, by the quality of the upper bound at the root node provided by the LP relaxation of the tight formulation in each setting. The following, very simplistic, heuristic shown in Algorithm 6 can be used to enhance the quality of the upper bound after having solved the root node with any of the cutting plane approaches proposed. The heuristic shown in Algorithm 6 corresponds to the general setting, but replacing (MIP-p-G$_{x,q,z}$) by (MIP-p-S$_{c,q,y}$) yields an equivalent heuristic to be used in the security setting.

```
1 {Input: Fractional solution from MASTER problem $(\bar{x}, \bar{q}, \bar{s}, \bar{d})$, Current UB};
2 if iteration %( $p \in \mathbb{Z}_+$) then
3     {Construct (rMIP-$p$-G$_{x,q,z}$), a relaxation of (MIP-$p$-G$_{x,q,z}$), by setting a subset of
      the continuous $q$ variables in ($\overline{\text{MIP-}p\text{-G}_{x,q,z}}$) to binary};
4     {Solve (rMIP-$p$-G$_{x,q,z}$)};
5     if v(rMIP-p-G$_{x,q,z}$)< UB then
6         {Add the following cut to MASTER: $\sum_{k\in K} \pi^k d^k \leq v(\text{rMIP-}p\text{-G}_{x,q,z})$};
7     end
8 end
```

Algorithm 6: UB primal heuristic for the GSG cutting plane approaches

One can use different criteria to select the subset of q variables to set to binary in the tight formulation for each setting. The more restrictive these criteria are, the better the quality of the upper bound provided will be, at the expense of more computational work. One disregards as candidate q variables to set to binary any q variables that have an integer value of 0 or 1 in the current fractional solution. We further consider different criteria to select a subset of the remaining q variables to be binary. We discuss the effect that these criteria, together with the number of times this UB heuristic is called, can have on the performance of the cutting plane approaches in the next section.

5.6 Computational experiments

We first study in Section 5.6.1 the effect that root node cut loop stabilization, the use of primal heuristics and interrupting the cut generation before the root node has been solved can have on the performance of the four cutting plane approaches studied–(C&B$_x$) and (C&B$_{x,q}$) for the general case and (C&B$_c$) and (C&B$_{c,q}$) for the security case–in order to determine the configuration that works best for each approach. Then, in Section 5.6.2 we compare the performance of the tuned cutting plane approaches against different solution methods in each case.

The general and security game instances used throughout this section are generated as follows. For the general games, leader and follower payoff values are uniform randomly generated between 0 and 10. For security games, penalties for the defender and the attacker, are uniform randomly generated between 0 and 5, and rewards for the defender and the attackers, are uniform randomly generated between 5 and 10. In both the general and security case, the follower's (attacker's) distribution over types π, is generated randomly in $[0, 1]^{|K|}$.

To adequately measure the scaling up capabilities of the proposed approaches, in both settings, with respect to the different parameters, we consider different sets of instances. For the general instances, we analyze how different solution methods perform against scale ups in the number of follower types and against scale ups in the number of leader and follower pure strategies. We therefore consider general instances with the following parameters:

- $I = J = \{5\}$, $K = \{25, 30, \dots, 95\}$,
- $I = J = \{5, 6, 7, 8, 9, 10\}$, $K = \{23\}$.

In either family of instances, for each instance size, we generate 5 uniform random instances, leading to 75 instances for the first family of instances and 30 instances for the second family of instances. For the security instances, we analyze how the methods proposed perform against scale ups in the number of attacker types and against scale ups in the number of targets. We always consider the number of security resources, m, to be $50\%|J|$. This ratio is chosen based on the so-called 'deployment to saturation' results in [Jain et al., 2012], where the authors indicate that a ratio of 0.5 accounts for the most computationally challenging problems. We thus consider security instances with the following parameters:

- $J = \{5\}$, $K = \{45, 50, \dots, 100\}$, $m = 50\%|J|$,
- $J = \{8, 10, 12, 14\}$, $K = \{25\}$, $m = 50\%|J|$.

Analogous to the general instance generation, in either family of security instances, for each instance size, we generate 5 uniform random instances, leading to 60 instances for the first family of instances and 20 instances for the second family of instances. We further consider an additional set of instances to analyze the behavior of the solution methods when there are very few attacker types but a very large number of targets:

- $J = \{25, 50, \dots, 175\}$, $K = \{4\}$, $m = 50\%|J|$.

Again, for a given instance size we generate 5 uniform random instances, leading to 35 instances to be solved.

5.6.1 Configuring the cutting plane approaches

In this section, we study the effect of root node cut loop stabilization and that of primal bound heuristics on the proposed cutting plane approaches. We further study the effects of interrupting the cut generation before the root node has been solved.

Impact of the root node cut loop stabilization

We present computational results that show the effect on the performance of our general and security cutting plane procedures of different configurations of λ in the cut loop stabilization. This parameter regulates the aggressiveness of the stabilization. We show how different values of λ affect the solution time of the integer problem, the time it takes to solve the root node and the number of nodes explored in the subsequent branch and bound process.

General case. We consider the 65 GSG instances described above corresponding to parameters $I = J = \{5\}$ and $K = \{25, 30, \dots, 85\}$ (the results for the other family of instances is comparable) to configure the parameter λ in a cut loop stabilization for the GSG approaches (C&B$_x$) and (C&B$_{x,q}$). We consider the performance of these methods for $\lambda \in \{0.2, 0.5, 0.7, 1\}$, where $\lambda = 0.2$ ensures an aggressive stabilization, while $\lambda = 1$ means stabilization is disabled. We show in Table 5.6.1 the average solution time over the instances for the different values of λ. One can see that stabilization is helpful in decreasing

Method-(λ)	Av.Time (s.)	Method-(λ)	Av.Time (s.)
C&B$_x$(0.2)	940.89	**C&B$_{x,q}$(0.2)**	**491.53**
C&B$_x$(0.5)	**881.80**	C&B$_{x,q}$(0.5)	535.06
C&B$_x$(0.7)	913.20	C&B$_{x,q}$(0.7)	552.67
C&B$_x$(1)	938.53	C&B$_{x,q}$(1)	538.75

Table 5.6.1: Average IP solution time, for different values of λ, over instances where $I = J = \{5\}$ and $K = \{25, 30, \dots, 85\}$

the average solution time over the set of instances explored. A value of $\lambda = 0.5$ works best for (C&B$_x$), whereas a more aggressive stabilization with $\lambda = 0.2$ further decreases the average solution time for (C&B$_{x,q}$). Also, note that, in general, stabilizing results in a gain in solution time.

In Table 5.6.2, we report the average LP solution time across the instances. Note that if one does not stabilize, all iterations at the root node produce cuts, whereas the more aggressively one stabilizes, the likelier that an iteration will not produce cuts and the interior point in the stabilization procedure will need to be readjusted, therefore taking longer to solve the LP. However, the additional solution time the LP requires, can result in producing better cuts that lead to sparser branch and bound trees, which in turn can result in faster IP solution times.

Next, we report in Table 5.6.3 the average number of branch and bound nodes explored across the instances. One can observe that the stabilization values which account for the

Method-(λ)	Av. Time (s.)	Method-(λ)	Av. Time (s.)
C&B$_x$(0.2)	3.20	C&B$_{x,q}$(0.2)	12.51
C&B$_x$(0.5)	3.65	C&B$_{x,q}$(0.5)	8.20
C&B$_x$(0.7)	2.77	C&B$_{x,q}$(0.7)	6.87
C&B$_x$(1)	**1.62**	**C&B$_{x,q}$(1)**	**6.12**

Table 5.6.2: Average LP solution time, for different values of λ, over instances where $I = J = \{5\}$ and $K = \{25, 30, \dots, 85\}$

smallest average solution time across the instances, are also responsible for the smallest average number of nodes.

	Nodes		Nodes
C&B$_x$(0.2)	1300203	**C&B$_{x,q}$(0.2)**	**438030**
C&B$_x$(0.5)	**1177943**	C&B$_{x,q}$(0.5)	467630
C&B$_x$(0.7)	1211479	C&B$_{x,q}$(0.7)	456678
C&B$_x$(1)	1250099	C&B$_{x,q}$(1)	484540

Table 5.6.3: Average number of nodes, for different values of λ, over instances where $I = J = \{5\}$ and $K = \{25, 30, \dots, 85\}$

One can conclude that, for the instances considered, the best stabilization configuration for (C&B$_x$) is to set $\lambda = 0.5$. Similarly, for (C&B$_{x,q}$), a value of $\lambda = 0.2$ works best.

Security case. Similarly, we consider the 60 SSG instances corresponding to parameters $J = \{5\}$, $m = 2$ and $K = \{45, 50, \dots, 100\}$ (the results for the other families of instances are comparable) to configure the parameter λ in the cut loop stabilization for the SSG approaches (C&B$_c$) and (C&B$_{c,q}$). We consider the performance of these methods for $\lambda \in \{0.2, 0.5, 0.7, 1\}$. We show in Table 5.6.4 the average solution time over the instances for the different values of λ.

One can see, that for both methods, a mild stabilization, accomplished by setting $\lambda = 0.7$, provides the best average solution time across the instances solved. In Table 5.6.5, we report the average LP time across instances for the different values of λ.

As before, for both methods, when no stabilization is applied, there are no iterations where no cuts are generated and the interior feasible point needs to be updated. Therefore, in these cases the LP relaxation is solved faster than when stabilization is applied.

Next, we report the average number of branch and bound nodes explored across instances in Table 5.6.6. Not stabilizing results, for these instances, in smaller branch and bound trees.

Method - λ	Av. Time (s.)	Method - λ	Av. Time (s.)
C&B$_c$(0.2)	318.16	C&B$_{c,q}$(0.2)	213.31
C&B$_c$(0.5)	311.85	C&B$_{c,q}$(0.5)	201.59
C&B$_c$(0.7)	**283.18**	**C&B$_{c,q}$(0.7)**	**173.13**
C&B$_c$(1)	292.10	C&B$_{c,q}$(1)	179.93

Table 5.6.4: Average IP solution time, for different values of λ, over instances where $J = \{5\}, m = 2$ and $K = \{45, 50, \ldots, 100\}$

Method - λ	Av. Time (s.)	Method - λ	Av. Time (s.)
C&B$_c$(0.2)	8.43	C&B$_{c,q}$(0.2)	24.72
C&B$_c$(0.5)	7.57	C&B$_{c,q}$(0.5)	16.32
C&B$_c$(0.7)	6.14	C&B$_{c,q}$(0.7)	13.42
C&B$_c$(1)	**2.78**	**C&B$_{c,q}$(1)**	**8.25**

Table 5.6.5: Average LP solution time, for different values of λ, over instances where $J = \{5\}, m = 2$ and $K = \{45, 50, \ldots, 100\}$

Method-(λ)	Av. Nodes	Method-(λ)	Av. Nodes
C&B$_c$(0.2)	285307	C&B$_{c,q}$(0.2)	167448
C&B$_c$(0.5)	278715	C&B$_{c,q}$0.5)	149631
C&B$_c$(0.7)	250658	C&B$_{c,q}$(0.7)	126627
C&B$_c$(1)	**248330**	**C&B$_{c,q}$(1)**	**117259**

Table 5.6.6: Average number of nodes, for different values of λ, over instances where $J = \{5\}, m = 2$ and $K = \{45, 50, \ldots, 100\}$

Remark that the configuration that corresponds to the smallest node exploration in the branch and bound tree is not the configuration that solves the integer problem the fastest. From Tables 5.6.4 and 5.6.6, we note that the configurations with $\lambda = 1$ and $\lambda = 0.7$ show very similar behavior in terms of solving time and nodes explored. Further, either configuration behaves better than those corresponding to $\lambda = 0.2$ and $\lambda = 0.5$. A first natural conclusion is that for the instances tested, a mild stabilization works better than a more aggressive one. Further, Table 5.6.4 shows that a mild stabilization accounts for somewhat faster solving times on average over the unstabilized approaches and is, thus, the preferred configuration.

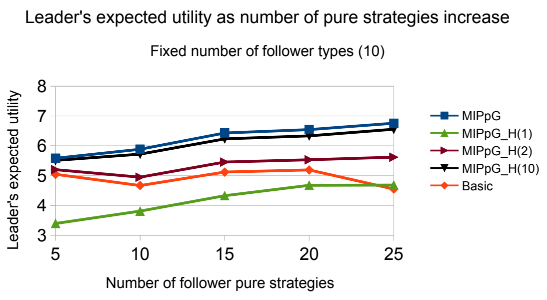
Primal heuristic effect

As discussed in Section 5.5.2, lower bound primal heuristics can provide an initial feasible solution and significantly improve the quality of the incumbent lower bound during the solving process leading to efficient pruning of the branch and bound tree, but this can require excessive computational effort and thus hinder the overall performance of an approach. If one is also interested in the amount of memory consumed when solving an instance, a lower bound enhancing approach can efficiently prune a branch and bound tree, leading to significant memory savings. Also, as discussed in Section 5.5.3, improving the upper bound can also lead to closing the optimality gap faster and having to explore a much smaller search tree. We study next the effect of the lower and upper bound primal heuristics on general and security instances.

General case In Section 5.5.2 we discussed two alternatives to obtain initial feasible solutions for (D2$_{x,q,s,f}$): solving (MIP-p-G_H) to produce an s-uniform leader mixed strategy and it's corresponding follower best response, or employing the 'basic' heuristic to calculate the follower's best response to an initial leader mixed strategy (a uniform distribution over his pure strategies, for example).

Note that the efficiency of (MIP-p-G_H) can be tuned by selecting an adequate value of $s \in \mathbb{Z}_+$, the parameter which determines how much one discretizes over the space of leader mixed strategies. The larger the value of s, the closer the solution provided by the heuristic is to the integer optimum but the higher the computational effort incurred in to obtain the solution. The lower the value of s, the quicker this heuristic MIP solves but the worse the lower bound becomes. We compare the solution quality, against the optimum value as computed by (MIP-p-G$_{x,q,z}$), and solution time of (MIP-p-G_H) for different values of $s \in \{1, 2, 10\}$ on GSG instances where $I = J = \{5, 10, \dots, 25\}$ and $K = \{10\}$ (For each instance size, we construct 5 uniform random instances leading to 25 instances in total). We further compare against using the 'basic' heuristic approach to generate a feasible solution for the GSG. The results are shown in Figure 5.6.1. Note that no solution time is reported for the 'basic' heuristic as the time taken is negligible. One can observe that for values of s different from 1, the solution time of (MIP-p-G_H) is worse than that of the optimal solution-providing MIP, (MIP-p-G$_{x,q,z}$), and for $s = 1$, the time taken is still quite considerable while the solution quality is much worse than using the 'basic' heuristic approach which is very fast. One can thus conclude that for the instance sizes tested, the (MIP-p-G_H) approach is not competitive and the 'basic' approach should be preferred.

Next, we analyze the effect that the lower bound-enhancing 'basic' and 'advanced' heuris-



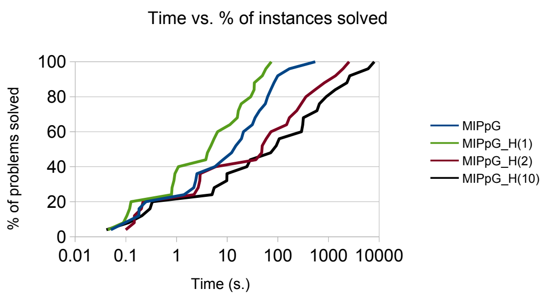


Figure 5.6.1: Quality of the lower bound and solution time using (MIPpG_H)

tics have on the GSG cutting plane approaches, (C&B$_x$) and (C&B$_{x,q}$). We compare different configurations of the 'basic' and 'advanced' heuristics against solving the cutting plane approaches without any bound enhancing heuristics.

We run both cutting plane algorithms over the 65 GSG instances with parameters $I = J = \{5\}$ and $K = \{25, 30, \ldots, 85\}$ (similar conclusions can be drawn from the set of instances that scales player pure strategies). We denote by (C&B$_{x,q}$(NO)) and (C&B$_x$(NO)) the cutting plane approaches that do not use any bound enhancing heuristics and by (C&B$_{x,q}$(H)) and (C&B$_x$(H)), those that do. We consider several settings:

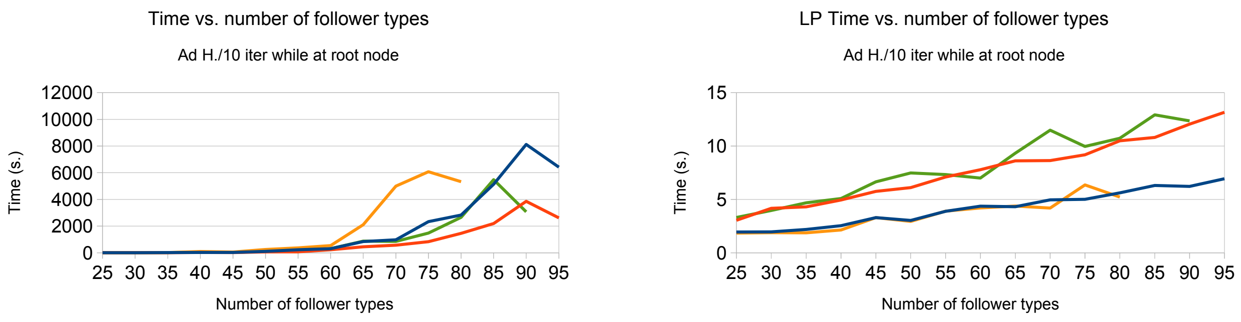
1. Only the 'basic' heuristic is run every 10 iterations.

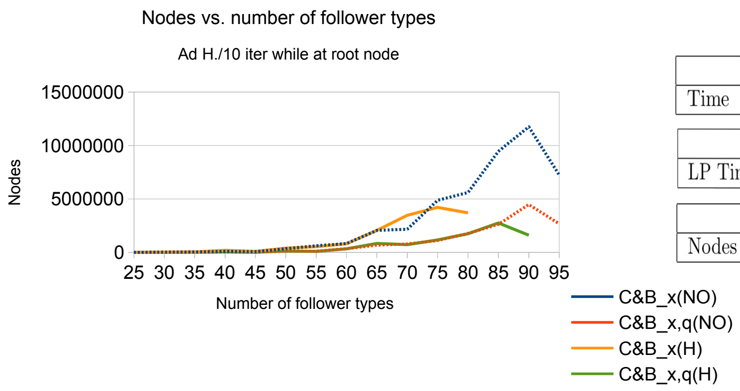
2. 'Basic' heuristic is run every 20 iterations, except when 'Advanced' heuristic is run. 'Advanced' heuristic is run every 100 iterations while the number of iterations is below 50000.

3. 'Advanced' heuristic is run every 10 iterations, but only while at the root node.

In Figure 5.6.2, we present the results for Configuration 3 with respect to solution time of the integer problem, solution time of the linear program and number of nodes explored in the branch and bound solution scheme.

The tables shown in the figure report the average values of integer solution time, LP solution time and number of branching nodes across instances, respectively, for the different methods. Note that the effect of the heuristic configuration is negative on all three counts. Including the primal heuristic results in fewer instances being solved to optimality within the considered time limit. The LP time is also worsened by the heuristic configuration. Branch and bound trees for (C&B$_{x,q}$(H)) are larger, on average, to the trees explored by (C&B$_{x,q}$(NO)). The trees become slightly smaller, on average, when the heuristic configuration is used on (C&B$_x$). Results for the other two heuristic configurations are reported in Figure A.4.1 in Section A.4 of the appendix. Unfortunately, none of the other configurations provide positive results either.





	C&B_x(NO)	C&B_x,q(NO)	C&B_x(H)	C&B_x,q(H)
Time	502	411	1530	735

	C&B_x(NO)	C&B_x,q(NO)	C&B_x(H)	C&B_x,q(H)
LP Time	3.58	7.11	3.41	7.65

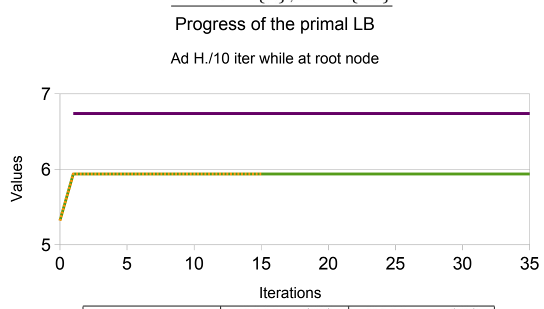
	C&B_x(NO)	C&B_x,q(NO)	C&B_x(H)	C&B_x,q(H)
Nodes	1084055	517154	1222786	508295

Figure 5.6.2: Effect of heuristic Config. 3 on gen. instances $I = J = \{5\}$, $K = \{25, \ldots, 85\}$

In Figure 5.6.3, we compare the evolution of the LB at the root node for some specific general instances solved by (C&B$_x$(H)) and (C&B$_{x,q}$(H)) under the heuristic Configuration 3 and under a heuristic configuration where the advanced heuristic is called at every root node iteration. The tables below the graphs report the LB percentage gap with respect to the integer optimum at the root. As one can see, Configuration 3 has very little effect on the LB. Calling the advanced heuristic at every root node iteration shows some bound improvement but this leads to an overall excessive computational effort which results in poor solution times. This suggests adapting our heuristic configuration so that it is called repeatedly early on and then discarded. From the tests we have run, we have seen that the LB behavior is erratic and instance dependent, making it hard to identify the best heuristic configuration that balances obtaining a good LB and a good computational performance.

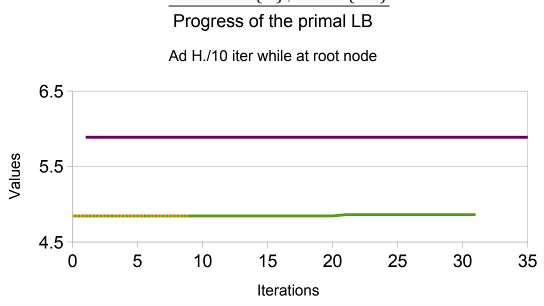
Figure 5.6.4 reports the LB percentage gap at the root node with respect to the integer optimal solution across the instances tested under the heuristic Configuration 3. One can see, that the gap remains consistently high. Opposite the graph, the table shows that under both cutting plane algorithms the average LB gap percentage is around 11%. Figure 5.6.5 shows the upper bound gap at the root node for the two sets of GSG instances considered. Note that the instances tackled have a relatively large UB gap, so it is to be expected that the solution methods will require computational effort to close the gap. The upper bound heuristic was implemented on the cutting plane approaches such that the heuristic is called

$I = J = \{5\}, K = \{55\}$

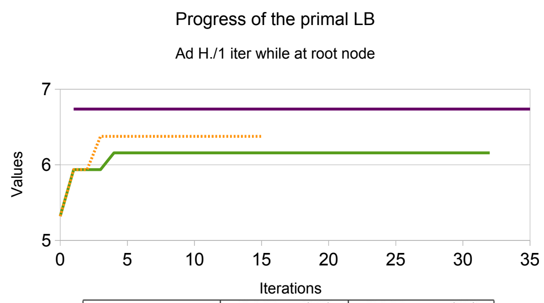


	C&B_x(H)	C&B_x,q(H)
LB GAP %	11.90	11.90

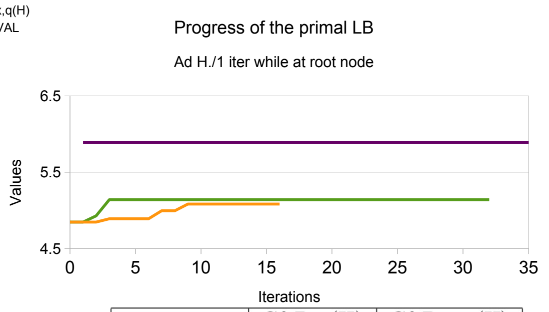
$I = J = \{5\}, K = \{70\}$



	C&B_x(H)	C&B_x,q(H)
LB GAP %	17.70	17.42

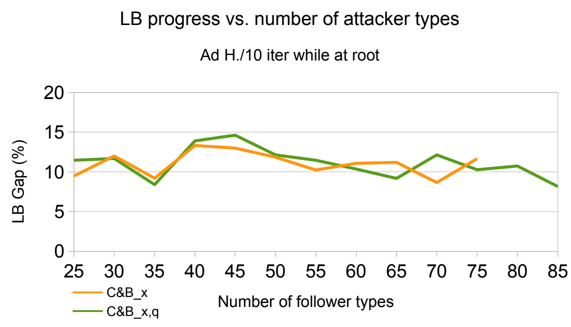


	C&B_x(H)	C&B_x,q(H)
LB GAP %	5.38	8.62



	C&B_x(H)	C&B_x,q(H)
LB GAP %	13.69	12.73

Figure 5.6.3: LB progress of (C&B$_x$) and (C&B$_{x,q}$) under different heuristic configurations



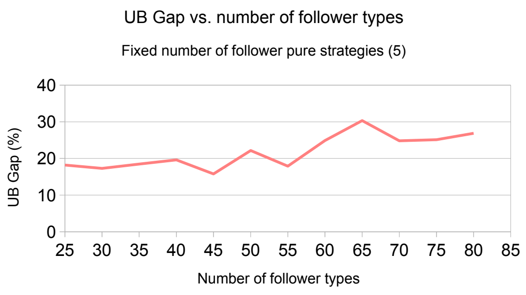
	C&B$_x$(H)	C&B$_{x,q}$(H)
LB Gap (%)	11.06	11.11

a) LB Gap at root node

b) Average LB Gap at root node

Figure 5.6.4: LB gap at the root node across instances where $I = J = \{5\}, K = \{25, 50, \dots, 85\}$

once after solving the root node. One then constructs (rMIP-p-G$_{x,q,z}$) as the relaxation of (MIP-p-G$_{x,q,z}$) where the subset of q variables that are set as binary are determined by the fractional solution recovered from the root node as follows: all q variables that have a fractional value between 0.01 and 0.25 are set as binary in (rMIP-p-G$_{x,q,z}$). We further tried setting all q variables that were sufficiently close to either 0 or 1, $0.01 \leq q \leq 0.1$ and $0.9 \leq q \leq 0.99$. In both cases one obtains a drop in the UB gap of 2-3% on average, but the added computation time required to solve (rMIP-p-G$_{x,q,z}$) makes it preferable not to invoke the upper bound heuristic for the GSG instances tested.



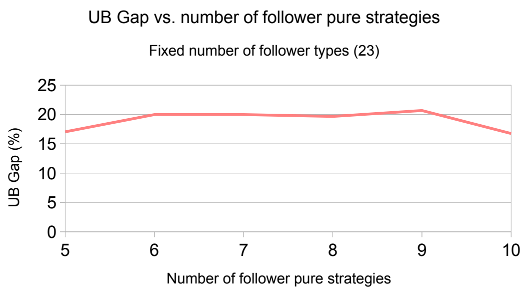


Figure 5.6.5: Upper bound gap at the root node over the two sets of GSG instances

It should be noted that neither the LB nor the UB heuristics enhance the performance of the GSG cutting plane approaches. However, this is not the case in the security setting where they aid in both reducing the solution time and in pruning the search tree. A reason for this could very well be that SSGs are highly structured while GSGs are not.

Security case Analogous to the general setting described above, one has two alternatives to obtain an initial feasible solution for (ERASER$_{c,q,s,f}$): solving (MIP-p-S_H) in order to obtain an s-uniform coverage strategy and it's corresponding attacker best response, or to employ the security 'basic' heuristic to calculate the attacker's best response to a defender uniform random coverage strategy over the targets.

The same experiments recorded in the previous section, adapted to the security setting and omitted here, show that the 'basic' security heuristic should be preferred over (MIP-p-S_H), for any value of s, to produce an initial feasible solution for (ERASER). The 'basic' approach provides a feasible solution in negligible time.

Next, we report the behavior of the cutting plane approaches (C&B$_{c,q}$) and (C&B$_c$) when lower bound primal heuristics are included. As before, we try different configurations of the 'basic' and 'advanced' heuristics and compare against not using any primal heuristics during the optimization process of the algorithms.

We run the cutting plane algorithms over the 60 SSG instances with parameters $J = \{5\}, K = \{45, \dots, 100\}$ and $m = 2$. Exploring the set of instances where we scale the number of targets leads to similar conclusions. We denote by (C&B$_c$(NO)) and (C&B$_{c,q}$(NO)) the cutting plane approaches that do not use bound enhancing heuristic and by (C&B$_c$(H)) and (C&B$_{c,q}$(H)) those that do. We consider several settings:

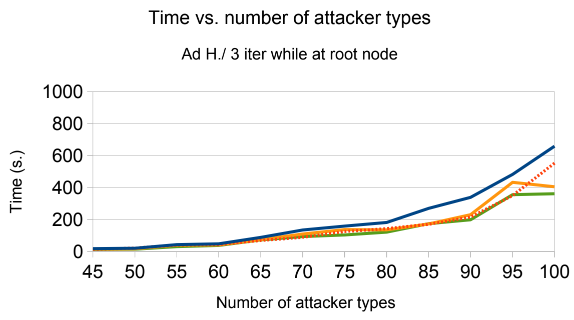
1. Only the 'basic' heuristic is run every 10 iterations.

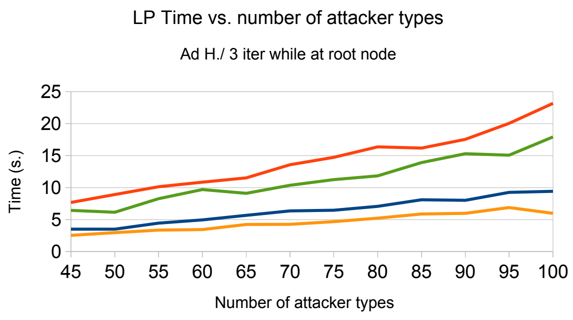
2. 'Basic' heuristic is run every 10 iterations, except when 'Advanced' heuristic is run. 'Advanced' heuristic is run every 100 iterations while under 50000 iterations.

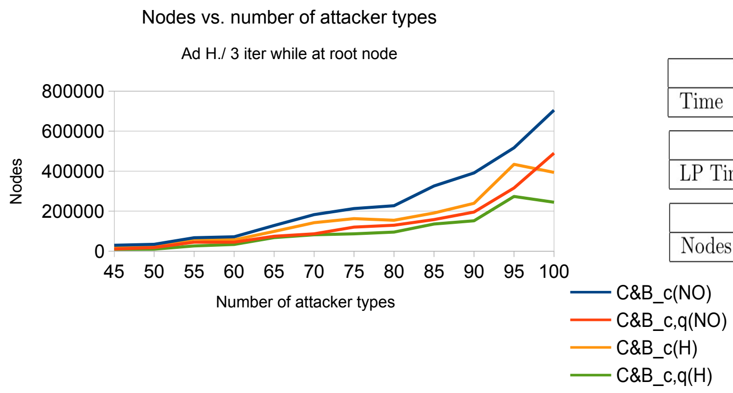
3. ‘Advanced’ heuristic is run every 5 iterations, but only while still at the root node.

4. ‘Advanced’ heuristic is run every 3 iterations while at the root node.

In Figure 5.6.6, we present the results for Configuration 4 with respect to solution time of the integer problem, solution time of the linear program and number of nodes explored in the branch and bound solution scheme. Configuration 4 is the best performing configuration for both (C&B$_c$) and (C&B$_{c,q}$). The tables shown report the average values of integer solution







	C&B_c(NO)	C&B_c,q(NO)	C&B_c(H)	C&B_c,q(H)
Time	204	153	138	132

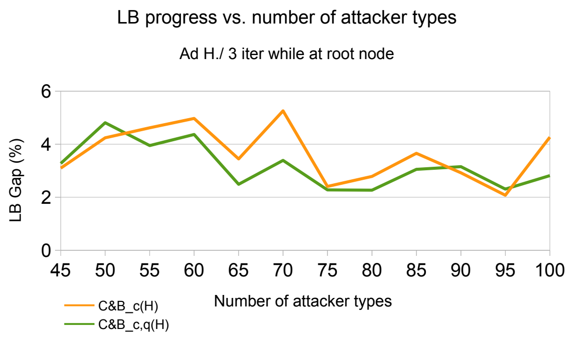
	C&B_c(NO)	C&B_c,q(NO)	C&B_c(H)	C&B_c,q(H)
LP Time	6.39	14.23	4.54	11.27

	C&B_c(NO)	C&B_c,q(NO)	C&B_c(H)	C&B_c,q(H)
Nodes	241161	140882	153120	101525

Figure 5.6.6: Effect of heur. Config. 4 on sec. instances $J = \{5\}, K = \{45, \dots, 100\}, m = 2$

time, LP solution time and number of branching nodes across instances, respectively, for the different methods. Note that unlike in the general case, in the security case, the use of primal lower bound heuristics has a major effect on pruning down the branch and bound tree under all the configurations tested. In terms of time, it outperforms the other configurations for both cutting plane algorithms and leads to an enhancement in the solution time with respect to the original algorithms. One can also observe, that employing Configuration 4 also diminishes the solution time of the root node and significantly reduces the size of the search tree. (C&B$_c$(H)) has a search tree which is, on average, 36% smaller than (C&B$_c$(H)) under Configuration 4. (C&B$_{c,q}$(H)) leads to a search tree under the same configuration which is, on average, 28% smaller than the tree explored by (C&B$_{c,q}$(NO)). The results for the other three heuristic configurations are reported in Figure A.5.1 in Section A.5 of the appendix.

Additionally, we report in Figure 5.6.7 the evolution of the gap between the lower bound at the root node and the optimal solution across the instances tested. On average the LB gap is around 3%. This suggests that it suffices to call the lower bound heuristic only at the root node. Devoting computational power to further increase the bound leads to a worsened performance of the methods.



	C&B$_c$(H)	C&B$_{c,q}$(H)
LB Gap (%)	3.61	3.16

a) LB Gap at root node b) Average LB Gap at root node

Figure 5.6.7: Lower bound gap at the root node across instances where $J = \{5\}, K = \{45, 50, \dots, 100\}$ and $m = 2$

In Figure 5.6.8, one can see the evolution of the UB gap across the different sets of instances considered. As one can see, for the first two sets of instances the gap is between 10% and 16%. The UB heuristic discussed in Section 5.5.3, could thus be beneficial in decreasing the UB gap. We implement the heuristic on both SSG cutting plane approaches and call it once after solving the root node. In the UB heuristic, one constructs (rMIP-p-S$_{c,q,y}$), a relaxation of (MIP-p-S$_{c,q,y}$), by selecting a subset of the q variables to be binary. To select an appropriate subset of these variables, we consider the fractional solution returned at the root node and set to binary the q variables in (rMIP-p-S$_{c,q,y}$) such that the corresponding fractional q's returned at the root node are between 0.01 and 0.25. We then solve the relaxation and add a cut to the master problem bounding the optimal solution from above.

For (C&B$_{c,q}$(H)), when the UB heuristic is used in conjunction with the best performing LB heuristic, it provides a decrease in the upper bound in excess of 1% and leads to faster solution time and a smaller search tree. For (C&B$_c$(H)), no further improvement is found by using the upper bound heuristic. We report the behavior of the methods with respect to solution time and UB quality after the root node in Figure 5.6.9. We would like to note that the upper bound heuristic proposed is very naïve and it is therefore comprehensible that the results are not astonishing. However, the use of the heuristic does in fact provide an improved performance for (C&B$_{c,q}$)–which is the most competitive of the two cutting

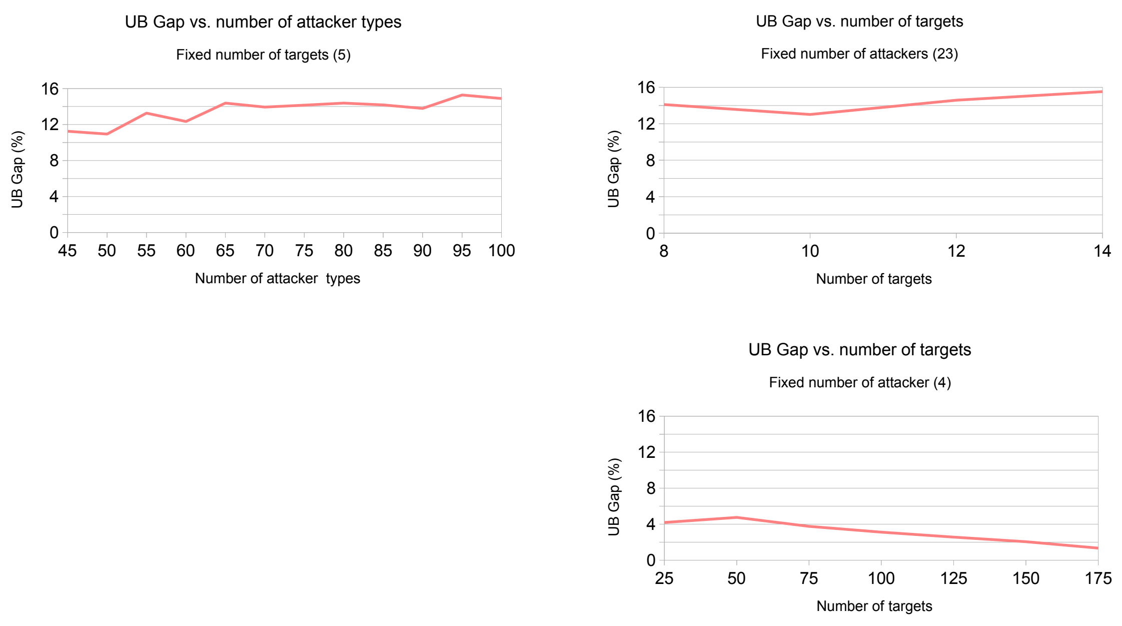
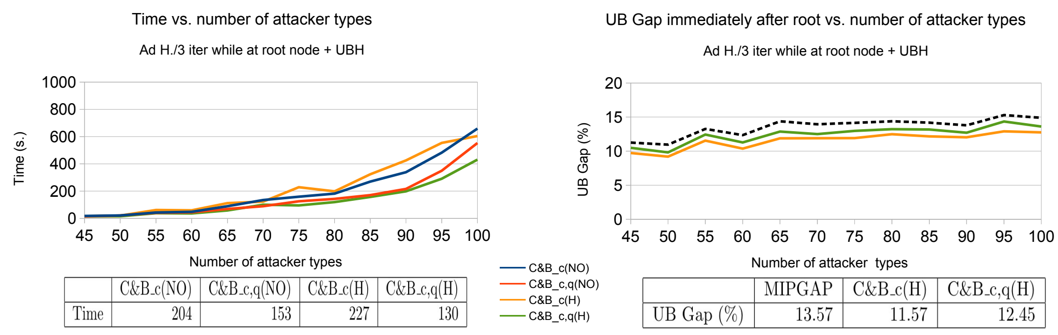


Figure 5.6.8: Upper bound gap at the root node across SSG instances

planes explored–with respect to not using any UB heuristic, and as such should be taken into consideration.



	C&B_c(NO)	C&B_c,q(NO)	C&B_c(H)	C&B_c,q(H)
Time	204	153	227	130

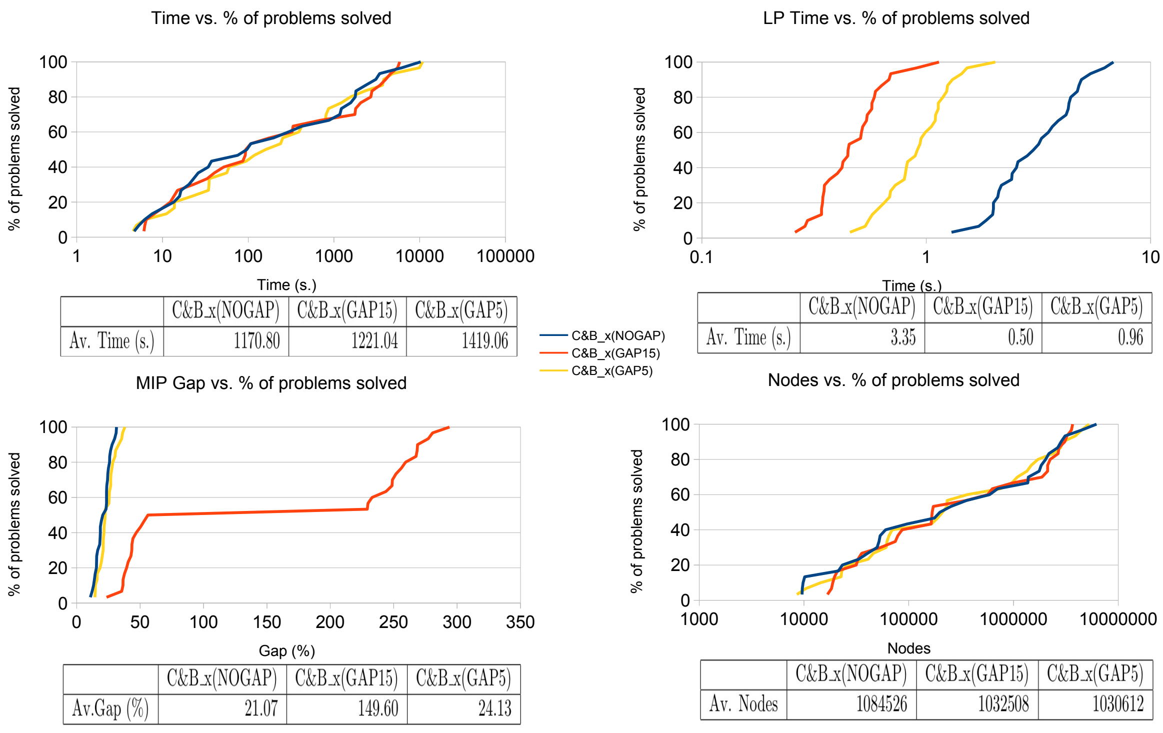
	MIPGAP	C&B_c(H)	C&B_c,q(H)
UB Gap (%)	13.57	11.57	12.45

Figure 5.6.9: Upper bound gap at the root node across SSG instances

Interrupting the cut generation at the root node

In order to improve the efficiency of the cutting plane methods, both in the general and in the security case, we study the effects of interrupting the cut generation procedure before the root node is solved. We compare solution time, LP solution time, strength of the upper bound provided by solving the LP relaxation and number of nodes explored in our original methods (where cuts are generated until the root node is solved), against an implementation of the methods where cut generation stops within a certain threshold of the root node's solution.

General case We run experiments on the 55 GSG instances corresponding to parameters $I = J = \{5\}$ and $K = \{25, 30, \ldots, 75\}$ (similar conclusions can be drawn from the set of instances that scale player pure strategies). We consider three different settings: (C&B$_x$(NOGAP)) and (C&B$_{x,q}$(NOGAP)) represent the standard methods where cuts are generated until the root node is solved; (C&B$_x$(GAP15)) and (C&B$_{x,q}$(GAP15)) are the cutting plane methods where cut generation is interrupted whenever the gap percentage between the upper bound and the dual lower bound at the root node drops below 15%. Similarly, (C&B$_x$(GAP5)) and (C&B$_{x,q}$(GAP5)) are the corresponding cutting plane methods which generate cuts until the upper bound-dual lower bound gap at the root node drops below 5%. Figures 5.6.10 and 5.6.11 show the performance of (C&B$_x$) and (C&B$_{x,q}$), respectively, under the three configurations. In each figure, the tables below the graphs show the average values of the studied parameters across the instances solved.



	C&B_x(NOGAP)	C&B_x(GAP15)	C&B_x(GAP5)
Av. Time (s.)	1170.80	1221.04	1419.06

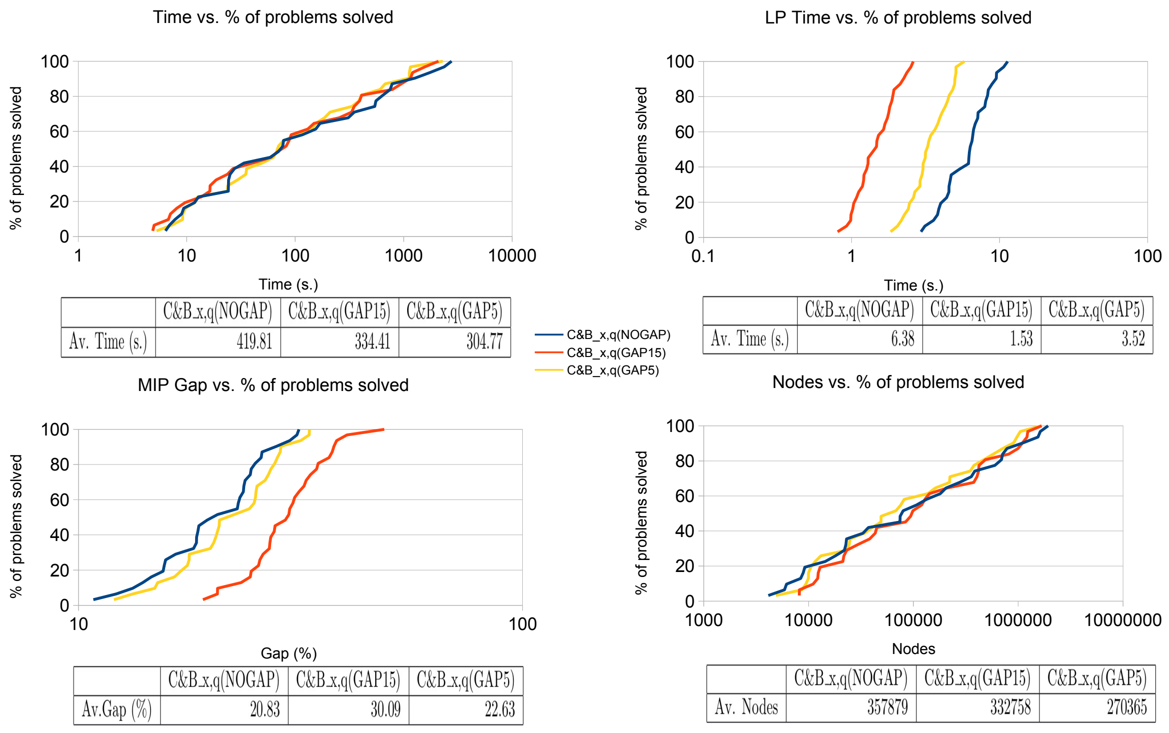
	C&B_x(NOGAP)	C&B_x(GAP15)	C&B_x(GAP5)
Av. Time (s.)	3.35	0.50	0.96

	C&B_x(NOGAP)	C&B_x(GAP15)	C&B_x(GAP5)
Av.Gap (%)	21.07	149.60	24.13

	C&B_x(NOGAP)	C&B_x(GAP15)	C&B_x(GAP5)
Av. Nodes	1084526	1032508	1030612

Figure 5.6.10: Effects on (C&B$_x$) of stopping cut generation during root node solve. Instances $I = J = \{5\}$ and $K = \{25, \ldots, 75\}$

Stopping the cut generation prior to the solution of the root node leads to fewer valid cuts being generated and as a result to a weaker upper bound on the optimal solution returned by the corresponding linear relaxation. From Figures 5.6.10 and 5.6.11, one observes that the LP solution time is smaller when the cut generation is interrupted as is the number of nodes explored. In the case of (C&B$_x$), interrupting the cut generation at the root node worsens the overall solution time by around 250 seconds on average. In the case of (C&B$_{x,q}$),



	C&B_x,q(NOGAP)	C&B_x,q(GAP15)	C&B_x,q(GAP5)
Av. Time (s.)	419.81	334.41	304.77

	C&B_x,q(NOGAP)	C&B_x,q(GAP15)	C&B_x,q(GAP5)
Av. Time (s.)	6.38	1.53	3.52

	C&B_x,q(NOGAP)	C&B_x,q(GAP15)	C&B_x,q(GAP5)
Av.Gap (%)	20.83	30.09	22.63

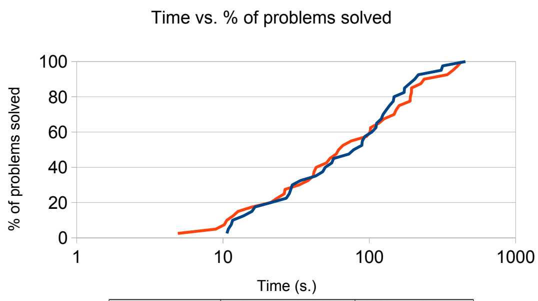
	C&B_x,q(NOGAP)	C&B_x,q(GAP15)	C&B_x,q(GAP5)
Av. Nodes	357879	332758	270365

Figure 5.6.11: Effects on (C&B$_{x,q}$) of stopping cut generation during root node solve. Instances $I = J = \{5\}$ and $K = \{25, \ldots, 75\}$

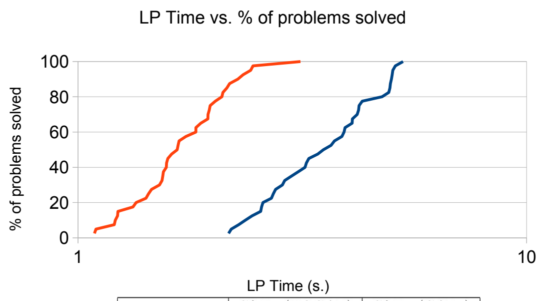
however, stopping the cut generation when the upper bound-dual lower bound gap at the root node drops below 5%, has a positive effect on the overall solution time, saving, on average, over 100 seconds of computation time. Thus, interrupting the cut generation at the root node can be advantageous for the cutting plane procedure (C&B$_{x,q}$).

Security case We run experiments on the 40 SSG instances corresponding to parameters $J = \{5\}$, $m = 2$, $K = \{45, \ldots, 80\}$ (similar conclusions can be drawn from the set of instances that scale the number of targets). We consider two different settings: (C&B$_c$(NOGAP)) and (C&B$_{c,q}$(NOGAP)) represent the standard cutting plane approaches where valid cuts are generated until the root node is solved and (C&B$_c$(GAP5)) and (C&B$_{c,q}$(GAP5)) represent the corresponding cutting plane approaches where valid cuts are generated while the upper bound-dual lower bound gap percentage at the root node remains over 5%. Figures 5.6.12 and 5.6.13 show the performance of (C&B$_c$) and (C&B$_{c,q}$), respectively, under both configurations. The tables under the graphs indicate the average values of the studied parameters across the instances solved.

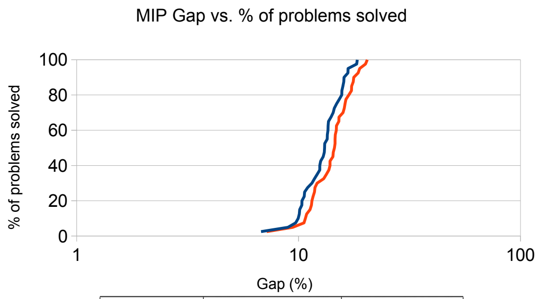
As in the general case, stopping the cut generation before the root node is solved leads to fewer cuts being generated and as a result to obtaining a weaker upper bound on the optimal solution. LP solution time is smaller for both methods when the cut generation



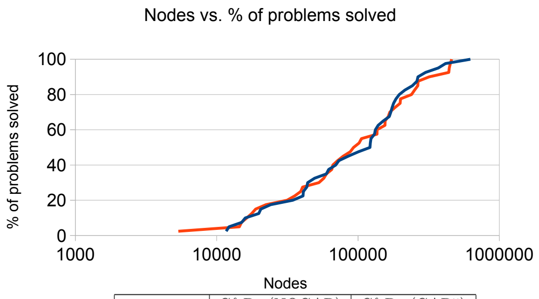
	C&B_c(NOGAP)	C&B_c(GAP5)
Av. Time (s.)	102.62	112.75



	C&B_c(NOGAP)	C&B_c(GAP5)
Av. Time (s.)	3.63	1.75

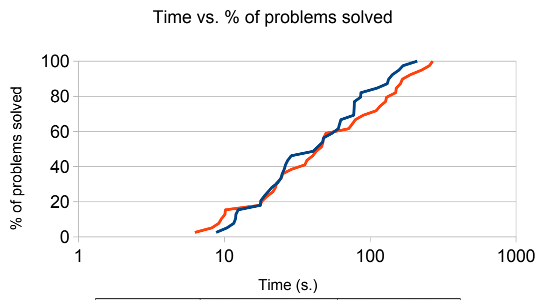


	C&B_c(NOGAP)	C&B_c(GAP5)
Av.Gap (%)	13.09	14.35

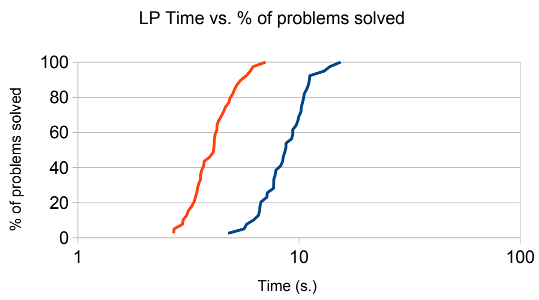


	C&B_c(NOGAP)	C&B_c(GAP5)
Av. Nodes	137700	143080

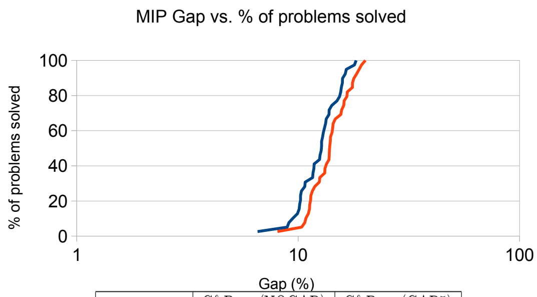
Figure 5.6.12: Effects on (C&B$_c$) of stopping cut generation during root node solve. Instances $J = \{5\}$, $m = 2$, $K = \{45, \ldots, 80\}$



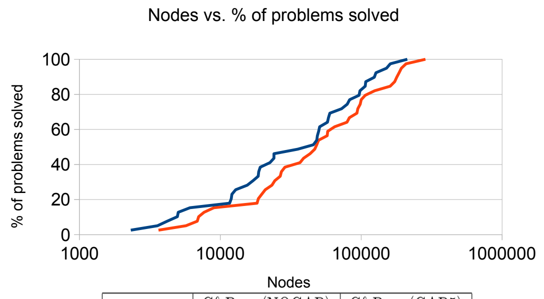
	C&B_c,q(NOGAP)	C&B_c,q(GAP5)
Av. Time (s.)	58.90	76.00



	C&B_c,q(NOGAP)	C&B_c,q(GAP5)
Av. Time (s.)	8.95	4.18



	C&B_c,q(NOGAP)	C&B_c,q(GAP5)
Av.Gap (%)	12.74	14.16



	C&B_c,q(NOGAP)	C&B_c,q(GAP5)
Av. Nodes	54117	73272

Figure 5.6.13: Effects on (C&B$_{c,q}$) of stopping cut generation during root node solve. Instances $J = \{5\}$, $m = 2$, $K = \{45, \ldots, 80\}$

is interrupted before the root node is solved. In this case, however, for both methods, interrupting the cut generation leads to then having to explore larger branch and bound trees and this leads to overall higher solution times. It would thus appear that for security games it is not worthwhile to interrupt the cut generation procedure until the root node has been solved.

5.6.2 Comparing the performance of the cutting plane approaches

In the general case, the best stabilization, heuristic configurations and cut generation strategies for the cutting plane methods, as determined in the previous section, are the following:

- (C&B$_x$)–Stabilization: $\lambda = 0.5$, Heur. Config.: No LB or UB heuristic. Cut generation: Until root is solved.
- (C&B$_{x,q}$)–Stabilization: $\lambda = 0.2$, Heur. Config.: No LB or UB heuristic. Cut generation: While UB-Dual LB gap percentage at root node remains over 5%.

We compare the performance of (C&B$_x$) and (C&B$_{x,q}$), under the above configurations, against the following GSG solution methods:

- (MIP-p-G$_{x,q,z}$), the best performing GSG MILP,
- CPLEX 12.7's automatic Benders decomposition of (MIP-p-G$_{x,q,z}$) where the continuous x variables and the binary q variables remain in the master problem and the continuous z variables are relegated to $|K|$ separation subproblems,
- (B&C$_{x,q}$), the branch and cut version of (C&B$_{x,q}$) where cuts are separated throughout the entire solving tree and not just at the root node.

The branch and cut algorithm (B&C$_{x,q}$) is run under the same configuration in terms of stabilization and primal heuristics as that of its cut and branch counterpart, (C&B$_{x,q}$). In the branch and cut algorithm, valid cuts are identified beyond the root node, extending the iterative process of cut generation to all the nodes explored in the branch and bound solution procedure. At every new node, the dual lower bound is reinitialized and cuts are generated until the gap between the the master problem solutions and the dual lower bound obtained from the separation problems is closed. In the branch and cut algorithm more valid cuts are generated than in the cut and branch algorithm. We study whether or not the strengthening effect of these extra cuts on the formulation accounts for faster solution times.

Similarly, in the security setting, the best stabilization, heuristic configurations and cut generation strategies for the cutting plane methods, as determined in the previous section, are the following:

- (C&B$_c$)–Stabilization: $\lambda = 0.7$, Heur. Config.: LB Advanced heuristic called every 3 iterations while at root node. No UB heuristic. Cut generation: Until root is solved
- (C&B$_{c,q}$)–Stabilization: $\lambda = 0.7$, Heur. Config.: LB Advanced heuristic called every 3 iterations while at root node. UB heuristic called once after solving the root node. Cut generation: Until root is solved

We compare the performance of (C&B$_c$) and (C&B$_{c,q}$), under the above configurations, against the following SSG solution methods:

- (MIP-p-S$_{c,q,y}$), the best performing SSG MILP,
- CPLEX 12.7's automatic Benders decomposition of (MIP-p-S$_{c,q,y}$) where the continuous c variables and the binary q variables remain in the master problem and the continuous y variables are relegated to $|K|$ separation subproblems,
- (B&C$_{c,q}$), the branch and cut version of (C&B$_{c,q}$) where cuts are separated throughout the entire solving tree and not just at the root node.

As in the general case, the branch and cut algorithm (B&C$_{c,q}$) is run under the same configuration in terms of stabilization and primal heuristics as that of its cut and branch counterpart, (C&B$_{c,q}$). In the branch and cut algorithm, cut generation is extended down the branch and bound tree leading to more valid cuts being generated in the solution process as compared to only generating cuts at the root node. We study whether or not the strengthening effect of these extra cuts on the formulation accounts for faster solution times than those of the cut and branch methods.

To measure the performance of the solution methods against scaling of the different game parameters (follower pure strategies/targets and number of follower types/attacker types), we use the sets of instances described at the beginning of Section 5.6. We report the solution times of the different methods across the instances considered. We further plot performance profile graphs to observe the behavior of the cutting plane methods with respect to LP relaxation solution time and number of nodes in the branch and bound tree. Since for the cutting plane approaches the LP time also includes the generation of the violated cuts that are added to the formulations, we also analyze the time it takes to solve the very last LP before the root node is solved, after all violated cuts have been added, and compare this time against the time it takes to solve the LP of the best performing general or security MILP. This gives an indication of how much faster it is to solve the lighter LP produced by our cutting plane approaches.

General case Figure 5.6.14 shows the running time of the different methods for both sets of instances considered. We set a solution time limit of 10800 seconds.

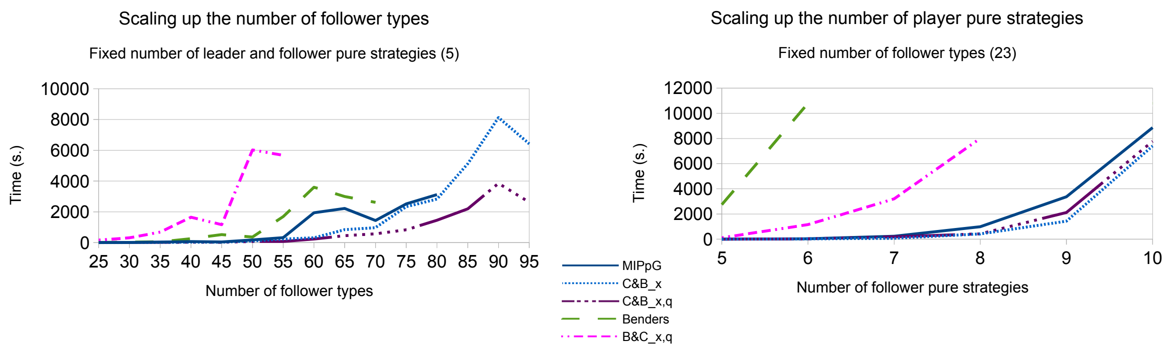


Figure 5.6.14: Solution time of the different GSG methods over different sets of instances

One can observe that for both sets of instances, (C&B$_x$) and (C&B$_{x,q}$) are the fastest solution methods. As one can see, (C&B$_{x,q}$) is the fastest solution method when we consider a large number of follower types and (C&B$_x$) performs faster when we scale the number of targets. Recall that (C&B$_{x,q}$) adds up to K feasibility or optimality cuts at each iteration at the root node and therefore performs well when the value of K is high.

In Figures 5.6.15 and 5.6.16, we plot performance profile graphs for both sets of GSG instances to measure solution time, LP solution time, solution time of the last linear program after all root cuts have been generated and number of nodes against the percentage of problems solved.

Across both sets of instances, (C&B$_x$) and (C&B$_{x,q}$) are faster than the other solution methods. When scaling up the number of follower types, (C&B$_{x,q}$) is able to solve the most instances to optimality within the time limit. When scaling the number of player pure strategies all but the automatic Benders decomposition and the full branch and cut approach are able to solve 95% of the instances within the time limit. Automatic Benders and the full branch and cut approaches are consistently the worst performing approaches with respect to solution time across the instances tested.

In addition, note that since the time devoted to generating cuts for (C&B$_x$) and (C&B$_{x,q}$) at the root node is included in their corresponding LP time, (MIP-p-G$_{x,q,z}$) has the overall fastest LP solution time. However, when we compare the solution time of the last linear program after all cuts produced by our approaches have been added to the correspond-

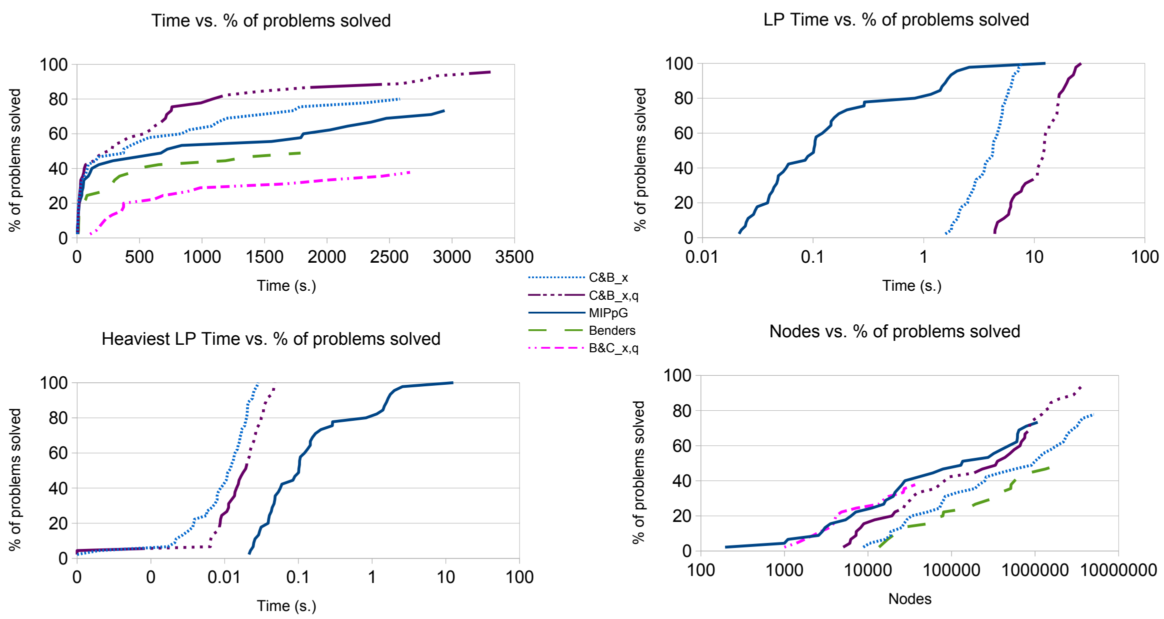


Figure 5.6.15: Performance profile graphs over instances $I = J = \{5\}, K = \{25, 30, \ldots, 95\}$

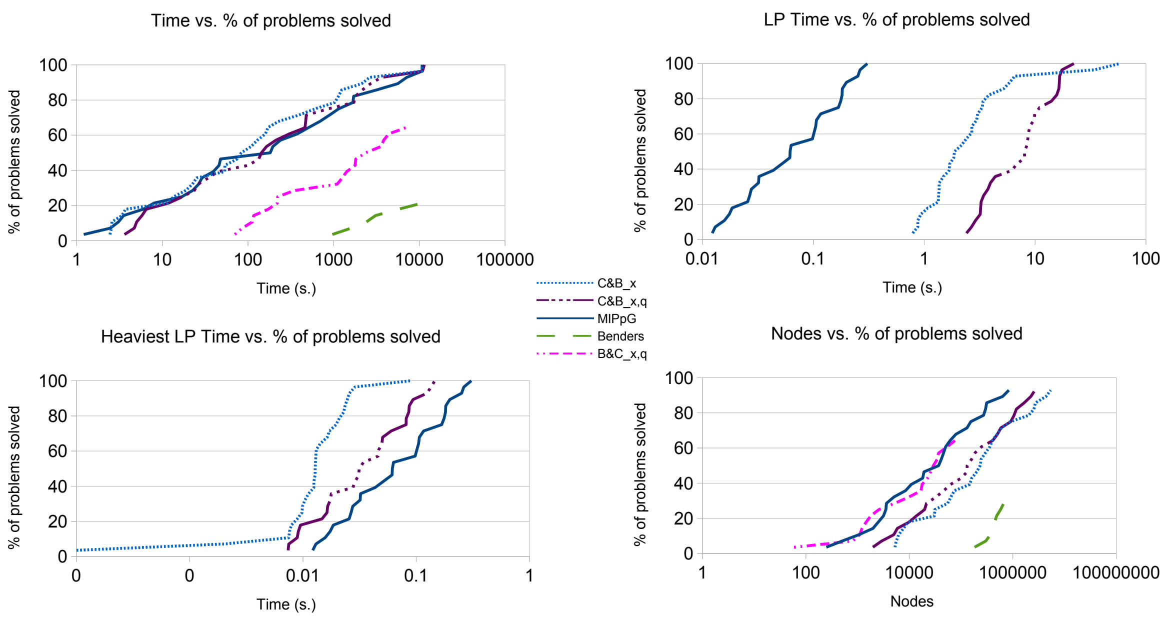


Figure 5.6.16: Performance profile graphs over instances I=J={5,6,...,10}, K={23}

ing master problem, we see that our approaches produce thinner relaxations which solve faster than the LP relaxation of (MIP-p-G$_{x,q,z}$). We do not explicitly show the LP solution time for (B&C$_{x,q}$) as it coincides with that of its cut and branch counterpart, neither do we plot the LP solution time for the automatic Benders approach. In terms of nodes in the branch and bound solving scheme, we observe that the full branch and cut approach explores the fewest nodes, followed by (MIP-p-G$_{x,q,z}$). Our approaches require to explore more nodes than the MILP, with (C&B$_x$) exploring larger trees than (C&B$_{x,q}$). CPLEX's automatic Benders decomposition requires the most nodes for the sets of instances tested. Both (C&B$_x$) and (C&B$_{x,q}$) suppose a significant improvement–in terms of solving speed and successfully handling scale ups–over the other methods tested over the two sets of instances described.

Security case Figure 5.6.17 shows the running time of our approaches across the different instance sets considered. A solution time limit of 3600 seconds is set for the first two sets of instances and a solution time limit of 10800 seconds is set for the last set of instances.

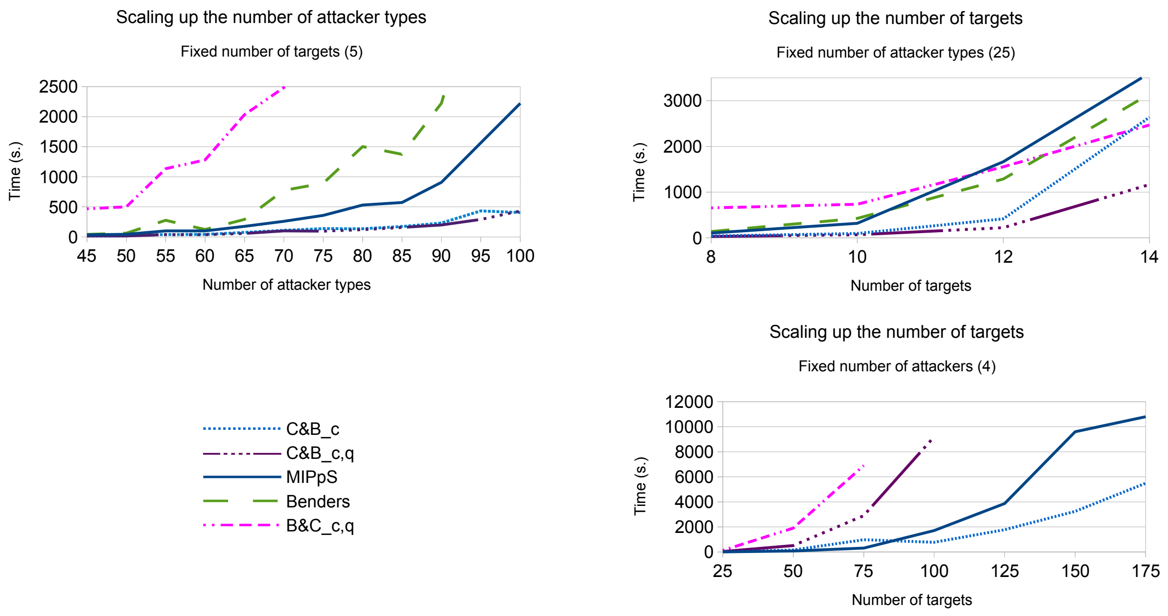


Figure 5.6.17: Solution time of the different SSG methods over different sets of instances

For the first two sets of instances, corresponding to scale ups in attacker types and targets respectively, (C&B$_{c,q}$) is the fastest approach followed closely by (C&B$_c$). When scaling up attacker types, (B&C$_{c,q}$) is the slowest approach, but when scaling up targets, the branch and cut approach becomes faster than the MILP, its automatic Benders decomposition and (C&B$_c$) for instances with 14 targets. For the third family of instances, where the number of targets can grow very large, (C&B$_{c,q}$) and it's branch and cut counterpart perform very poorly, not being able to scale above 75-100 targets. For these instances, (C&B$_c$) is the fastest method, significantly outperforming the MILP. We do not consider automatic Benders in this setting as preliminary results showed it performed very poorly for even the smallest instances.

We plot performance profile graphs for the SSG instances considered in Figures 5.6.18, 5.6.19 and 5.6.20. We measure the total solution time, the LP solution time, the solution time of the last LP in our cutting plane approaches after all cuts have been added at the root node and the number of nodes in the branch and bound scheme against the percentage of problems solved.

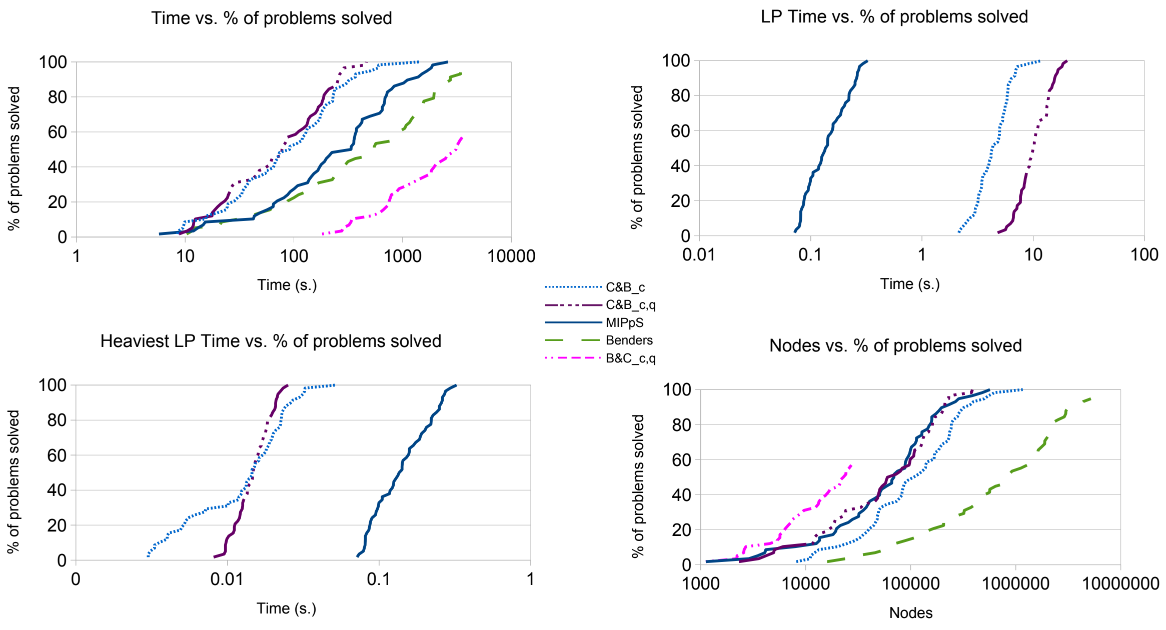


Figure 5.6.18: Profile graphs over instances $J = \{5\}, K = \{45, 50, \ldots, 100\}, m = 2$

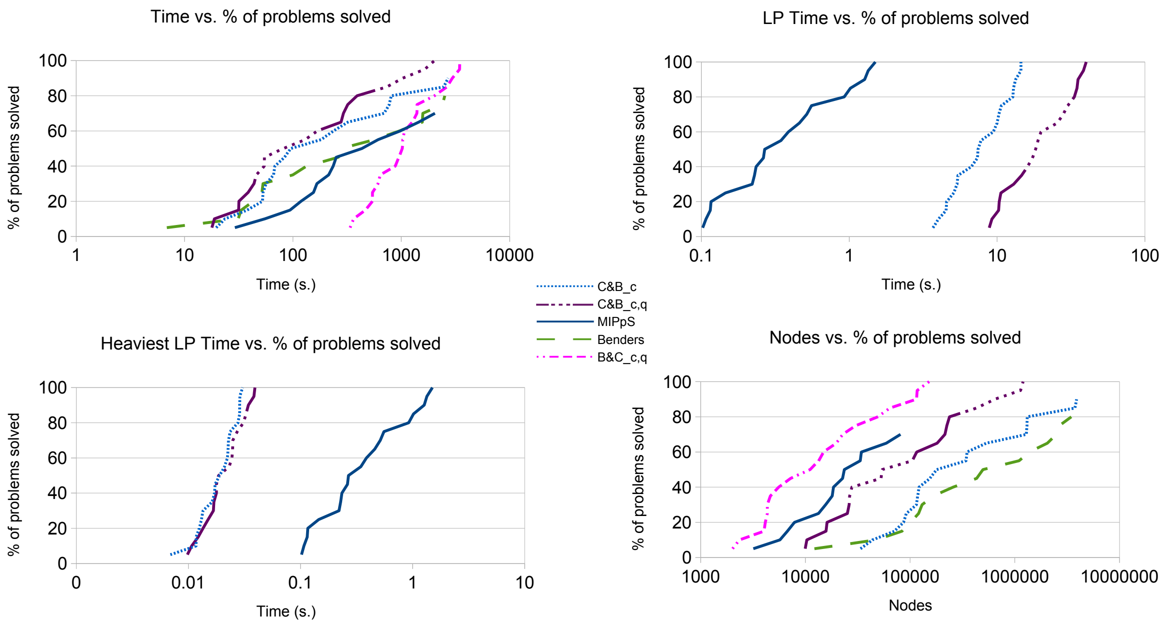


Figure 5.6.19: Profile graphs over instances $J = \{8, 10, \ldots, 14\}, K = \{25\}, m = 50\%|J|$

Consider the performance of the solution methods over the sets of instances that scale up attacker types and targets in Figures 5.6.18 and 5.6.19, respectively. One can see a similar behavior in the performance of (C&B$_{c,q}$) and (C&B$_c$) compared to that of their GSG counterparts. Our cutting plane approaches are faster than competing approaches and have an LP solution time which is worse than that of (MIP-p-S$_{c,q,y}$). However, when

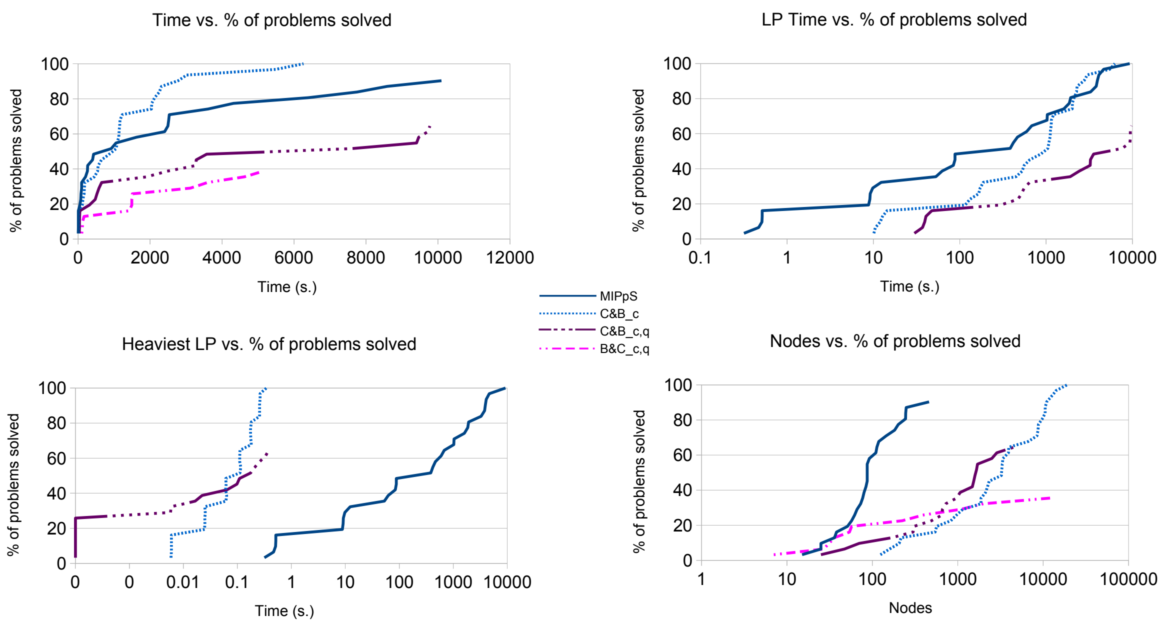


Figure 5.6.20: Profile graphs over instances $J = \{25, 50, \dots, 175\}, K = \{4\}, m = 50\%|J|$

one compares the solution time of the last linear program in (C&B$_c$) and (C&B$_{c,q}$) after all root cuts have been added, to the solution time of the linear relaxation of (MIP-p-S$_{c,q,y}$), the linear programs in our approaches solve faster than the LP relaxation of (MIP-p-S$_{c,q,y}$). Also, with respect to nodes in the branch and bound scheme, (MIP-p-S$_{c,q,y}$) generally employs fewer nodes than our approaches, with (C&B$_x$) exploring larger search trees than (C&B$_{c,q}$). When scaling the number of attacker types, (C&B$_{c,q}$) tends to explore even fewer nodes than those explored by (MIP-p-S$_{c,q,y}$). The full branch and cut approach, (B&C$_{c,q}$) performs better when moderately scaling the number of targets, being able to solve all instances within the time limit in this case, and leads to the smallest branch and bound trees. CPLEX's automatic Benders of (MIP-p-S$_{c,q,y}$) is the worst performing approach across all sets of instances considered.

Consider the performance of the solution methods over the set of instances where there is a small number of attacker types but a very large number of targets in Figure 5.6.20. The approach (C&B$_{c,q}$) and its branch and cut version perform very poorly only being able to solve around 60% and 30% of the instances, respectively, within the time limit. In addition, the branch and cut approach explores search trees which are considerably larger than those explored by the MILP, which is the approach that explores the fewest nodes. The MILP also falls short of solving all instances to optimality, being able to solve around 90% within the specified time limit. The cutting plane approach (C&B$_c$) is able to solve all instances to optimality and is faster than the competing approaches even though it explores more nodes in the branch and bound scheme than its competitors. For the harder instances, it solves the root node faster than the MILP even when considering the time devoted to generating the Benders cuts at the root node.

As in the general case, our cutting plane approaches for the security case (C&B$_c$) and (C&B$_{c,q}$) represent a significant improvement over the other solution methods considered in this section. (C&B$_{c,q}$) should be preferred to (C&B$_c$) when facing a scale up in the number of attacker types and when facing a moderate scale up in the number of targets. When there are very few attackers but a large number of targets, (C&B$_c$) should be the solution approach of choice.

5.7 Conclusions

We have exploited the fact that for GSGs and SSGs we have two sets of valid MILP formulations. We saw in Chapter 4 that the formulations with a sparse LP relaxation allow for quick resolution speeds on smaller instances while the formulations with a heavier LP relaxation ensure a good quality bound on the optimal solution. The proposed decomposition approaches have thus consisted in strengthening the LP relaxation of the weaker formulations by embedding optimality and/or feasibility cuts obtained from different Benders decompositions on the LP relaxations of the stronger formulations. We have further enhanced the decomposition approaches by considering primal upper and lower bound heuristics, dual lower bound stabilization and interruption of the cut generation at the root node prior to root node resolution. The first conclusion that one can draw is that the work developed in this chapter is only partially dependent on the problem being tackled. The methodology proposed could be extended to other domains where one has valid MILP formulations for a given problem and one such formulation has a tight LP relaxation that accounts for poor resolution speed but good bound quality while another formulation has a sparser LP relaxation with good resolution speed and poor bound quality. This is the case, for example, in the domains of Vehicle Routing Problems and Network Pricing Problems, to name but two.

With regards to the decomposition methods proposed in this chapter, some follow-through work should be explored. In our work, we have tested the impact of mildly or aggressively stabilizing the separation point in our cutting plane approaches by tuning the stabilization parameter $\lambda \in \{0, 0.2, 0.5, 0.7, 1\}$. It is likely that a finer tuning of the parameter λ might result in an improved performance of our approaches. Further, the average upper bound gap over both the general and security instances remains around 10-20%. A thorough polyhedral study of (MIP-p-G$_{x,q,z}$) and (MIP-p-S$_{c,q,y}$) could be conducted to identify strong facet-defining inequalities that might further decrease the UB gap. In addition, a less naïve UB heuristic than the one proposed in this chapter could be implemented into the cutting plane approaches to aid in closing the optimality gap quicker. Finally, we have attributed the sub-par performance of the stabilization and heuristic add-ons to the GSG cutting plane approaches to a lack of structure in GSGs. Perhaps a more in-depth look at these games would reveal some structure that the add-ons could benefit from, leading to an improved performance of the GSG cutting plane approaches studied in this chapter.

Chapter 6

Stackelberg security games with combined defender strategies

6.1 Introduction

In this chapter, we propose a special type of SSG on a network where the mixed strategy the defender commits to consists of two coverage distributions. A first coverage distribution over edges and a second coverage distribution over the targets, which are located at the nodes. The defender can only select m edges–pair $2m$ nodes–and defend m targets. Further, coverage on a target can only take place if the node at which the target is located, is incident to a covered edge. Once the defender has committed to a mixed strategy, an attacker of type $k \in K$, who plays the game with probability π^k, attacks the target that maximizes his payoff.

We propose a compact MILP formulation, (FENCE$_{c,z,q,s,f,g}$)-(**F**rontier **E**nforcement and **N**eutralization of **C**riminal **E**nterprise) that solves for the optimal coverage distributions over edges and targets. The formulation proposed has a polynomial number of variables and an exponential number of constraints.

We further provide two sampling methods that recover an implementable strategy for the defender given the two optimal coverage distributions. The first sampling method is exact, in that the implementable strategy recovered agrees with the optimal coverage probabilities. The second sampling method is a simpler two-stage sampling method. The method first samples over the edges, and then samples over the targets. Because of these two sampling stages, the implementable strategy recovered is an approximation in the sense that it may not fully comply with the optimal coverage distributions.

The game modeled in this chapter is applicable to real life situations where a defender has to create patrol plans that combine the importance of a globally orchestrated strategy with the intricacies of local patrols within sectors. For example, the nodes in the network might represent undermanned police precincts and the edges could represent the adjacency of the precincts. A central planner might have to pair adjacent precincts and then deploy a security resource from each precinct pair to protect a target within the territory of the paired pair of precincts. Precisely this scenario is tackled in Chapter 7.

Our computational results show that the approximate two-stage sampling method produces estimated coverage distributions over edges and targets which are close to the optimal coverage distributions returned by (FENCE$_{c,z,q,s,f,g}$). Further, we run computational tests to compare the resolution speed of (FENCE$_{c,z,q,s,f,g}$) to that of the GSG formulation (D2$_{x,q,s,f}$), where all the exponentially many defender pure strategies are explicitly included in the formulation. Our results over the instances tested indicate that (FENCE$_{c,z,q,s,f,g}$) allows for faster resolution times than (D2$_{x,q,s,f}$), being able, in addition, to solve larger instances, which are out of the scope of (D2$_{x,q,s,f}$).

The rest of the chapter is divided as follows. In Section 6.2 we formally define the problem we study. In Section 6.3 we present our formulation and provide sampling methods to retrieve an implementable solution. In section 6.4, we run computational experiments to compare the quality of the sampling methods and to evaluate the performance of the proposed formulation. In Section 6.5, we provide closing remarks and suggest future work.

6.2 Problem definition

Given a graph G=(V, E), let V be the set of nodes and E the set of edges that provide an allowed pairing of nodes. We denote by $\delta(v) \subset E$ the set of edges incident to node $v \in V$. Similarly, for any $U \subset V$, $\delta(U) \subset E$ denotes the edges between U and $V \setminus U$ and $E(U) \subset E$ denotes the edges between nodes in U. Further, $\overline{E(U)} = E(U) \cup \delta(U)$. Let $\mathcal{M}_m$ be the set of all possible matchings of size m in G:

$$\mathcal{M}_m := \left\{ w \in \{0,1\}^{|E|} : \sum_{e \in E} w_e = m, \sum_{e \in \delta(v)} w_e \leq 1 \quad \forall v \in V \right\}$$

For every node $v \in V$, let J_v be the set of targets inside that node. Note that $\{J_v\}_{v \in V}$ is a partition of the set of targets J, *i.e.*, $\cup_{v \in V} J_v = J$ and $J_u \cap J_v = \emptyset$ for all $u \neq v$. A pure strategy for the defender selects a matching from $\mathcal{M}_m$ and for each edge $e = (u, v)$ in the selected matching, further selects one target to protect in $J_e = J_u \cup J_v$. It follows that the set I of defender pure strategies can be expressed as

$$I = \left\{ (y, w) \in \{0,1\}^{|J|} \times \{0,1\}^{|E|} : w \in \mathcal{M}_m, \sum_{j \in \cup_{v \in U} J_v} y_j \leq \sum_{e \in \overline{E(U)}} w_e \, \forall U \subseteq V, \sum_{j \in J} y_j = m \right\},$$

For $(y, w) \in I$, the variable y_j indicates whether the target $j \in J$ is protected and the variable w_e indicates whether edge e is selected. The first condition indicates that a matching of size m is selected. The second condition guarantees that the coverage provided to any given subset of nodes is bounded by the coverage on all incident edges to this subset of nodes. The third condition enforces that total coverage on targets be equal to the available number of resources.

The GSG defined by all defender strategies $i = (y, w) \in I$ and follower strategies $j \in J$ can be solved by explicitly enumerating all the strategies and using the GSG MILP $(\text{D2}_{x,q,s,f})$ to obtain the optimal mixed strategy for the defender.

$(\text{D2}_{x,q,s,f})$

$$\begin{aligned}
\text{Max} \quad & \sum_{k \in K} \pi^k f^k && && (6.2.1)\\
& \sum_{(y,w) \in I} x_{(y,w)} = 1, && && (6.2.2)\\
& x_{(y,w)} \geq 0 && \forall (y,w) \in I, && (6.2.3)\\
& \sum_{j \in J} q_j^k = 1 && \forall k \in K, && (6.2.4)\\
& q_j^k \in \{0,1\} && \forall j \in J, \forall k \in K, && (6.2.5)\\
& 0 \leq s^k - \sum_{(y,w) \in I} C^k_{(y,w)j} x_{(y,w)} \leq (1 - q_j^k) M^2 && \forall j \in J, \forall k \in K, && (6.2.6)\\
& f^k \leq \sum_{(y,w) \in I} R^k_{(y,w)j} x_{(y,w)} + (1 - q_j^k) M^1 && \forall j \in J, \forall k \in K. && (6.2.7)
\end{aligned}$$

This approach, however, is highly intractable as the set I is of exponential size and this leads to a MILP with exponentially many variables and constraints. Further, recall the payoff relation between GSGs and SSGs adapted to our current notation:

$$R^k_{(y,w)j} = \begin{cases} D^k(j|c) & \text{if } (y,w) : y_j = 1 \\ D^k(j|u) & \text{if } (y,w) : y_j = 0 \end{cases} \quad (6.2.8) \qquad C^k_{ij} = \begin{cases} A^k(j|c) & \text{if } (y,w) : y_j = 1 \\ A^k(j|u) & \text{if } (y,w) : y_j = 0 \end{cases} \quad (6.2.9)$$

In the next section, we propose a compact SSG MILP formulation with a polynomial number of variables and exponentially many constraints.

6.3 Solving the problem

In Section 6.3.1 we first provide the formulation for the security problem we study. In Section 6.3.2 we provide two sampling methods to recover an implementable defender mixed strategy from an optimal solution to our compact model.

6.3.1 The formulation: (FENCE$_{c,z,q,s,f,g}$)

The formulation we propose, following the logic of (ERASER$_{c,q,s,f}$), is based on the observation that if payoffs for the players are given by (6.2.8) and (6.2.9), then the payoffs for each player only depend on whether or not the attacked target is covered. The coverage on a target $j \in J$ can be expressed by summing the individual contributions over all the pure strategies that allocate coverage to that target. Namely,

$$c_j = \sum_{(y,w) \in I : y_j = 1} x_{(y,w)} \tag{6.3.1}$$

We further consider the variables

$$z_e = \sum_{(y,w) : w_e = 1} x_{(y,w)} \qquad \forall e \in E, \tag{6.3.2}$$

where z_e represents the coverage on edge $e \in E$. Constraint (6.3.2) expresses this coverage as the sum of the contributions over the pure strategies that allocate coverage to edge $e \in E$. Further, consider the following variables:

$$g_{ej} = \sum_{(y,w) \in I : y_j = 1, w_e = 1} x_{(y,w)} \qquad \forall e \in E, j \in J_e. \tag{6.3.3}$$

Note that g_{ej} represents the combined coverage on edge $e \in E$ and on target $j \in J_e = J_u \cup J_v$ and it can be expressed as the contribution over all pure strategies that assign coverage to both edge $e \in E$ and target $j \in J_e$.

We thus propose the following MILP formulation employing polynomially many variables and exponentially many constraints:

(FENCE$_{c,z,q,s,f,g}$)

$$\text{Max} \quad \sum_{k \in K} \pi^k f^k \tag{6.3.4}$$

$$\text{s.t.} \quad \sum_{j \in J} q_j^k = 1 \qquad \forall k \in K, \tag{6.3.5}$$

$$q_j^k \in \{0,1\} \qquad \forall j \in J, \forall k \in K, \tag{6.3.6}$$

$$\sum_{j \in J} c_j = \sum_{e \in E} z_e = m, \tag{6.3.7}$$

$$\sum_{e \in \delta(v)} z_e \leq 1 \qquad \forall v \in V, \tag{6.3.8}$$

$$\sum_{e \in E(U)} z_e \leq \frac{|U| - 1}{2} \qquad \forall U \subseteq V,\ |U| \geq 3, |U| \text{ odd} \tag{6.3.9}$$

$$\sum_{e \in E_j} g_{e,j} = c_j \qquad \forall j \in J \tag{6.3.10}$$

$$\sum_{j \in J_e} g_{e,j} = z_e \qquad \forall e \in E \tag{6.3.11}$$

$$f^k \leq D^k(j|c)c_j + \\ D^k(j|u)(1-c_j) + (1-q_j^k)\cdot M^1 \qquad \forall j \in J, \forall k \in K, \tag{6.3.12}$$

$$0 \leq s^k - A^k(j|c)c_j - \\ A^k(j|u)(1-c_j) \leq (1-q_j^k)\cdot M^2 \qquad \forall j \in J, \forall k \in K \tag{6.3.13}$$

$$c_j \in [0,1] \qquad \forall j \in J, \tag{6.3.14}$$

$$g, z \geq 0, \tag{6.3.15}$$

$$s, f \in \mathbb{R}^K. \tag{6.3.16}$$

Constraints (6.3.5) and (6.3.6) ensure that the each attacker $k \in K$ attacks a single target $j \in J$ with probability 1. Constraint (6.3.7) indicates that the defender uses all his resources in a feasible solution and that in order to form his resources he pairs up nodes without exceeding the number of resources he wants to deploy. Constraint (6.3.8) indicates that a node's contribution to a pairing cannot exceed 1. Constraints (6.3.9), introduced in [Edmonds, 1965], are known as *Odd Set Inequalities*, and together with (6.3.7) and (6.3.8) enforce that the coverage probabilities on the edges belong to the convex hull of the matching polytope of size m. Constraints (6.3.10) and (6.3.11) enforce the conservation between marginal coverages in nodes and edges. Finally, Constraints (6.3.12) and (6.3.13) are the same as in (ERASER$_{c,q,s,f}$) and ensure that c and q are mutual best responses. The objective function in the formulation, maximizes the defender's expected utility.

Discussion To ensure the correctness of the formulation (FENCE$_{c,z,q,s,f,g}$) we need to be able to recover variables $x_{(y,w)}$ for $(y,w) \in I$–that represent the probability distribution over the defender pure strategies–from an optimal solution (c,z,q,s,f,g) to (FENCE$_{c,z,q,s,f,g}$). In particular, we need to find variables $x \in [0,1]^{|I|}$ that satisfy Constraint (6.3.2). Remark that the odd set inequalities are necessary. Figure 6.3.1 shows a solution for $m = 2$ where the z variables violate the odd set inequalities and one cannot write this solution as a convex combination of pure matchings of the nodes making it impossible to retrieve an implementable defender strategy x.

Similarly, Constraints (6.3.10) and (6.3.11) also play a vital role in that they establish a link between the coverage variables on the edges and on the targets. This becomes much more apparent if one applies Farkas' Lemma [Farkas, 1902] on the linear system defined by (6.3.10), (6.3.11) and $g \geq 0$ to understand which conditions on c and z guarantee feasibility of the system. Applying Farkas provides the following necessary conditions on c and z which offer a more direct interpretation:

$$\sum_{e \in E: e \in \overline{E}(U)} z_e \geq \sum_{j \in \cup_{u \in U} J_u} c_j \;\; \forall U \subseteq V, \tag{6.3.17}$$

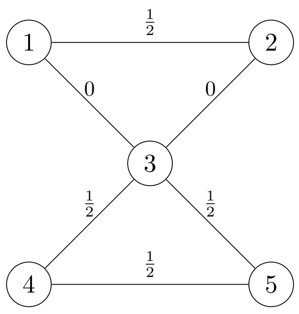


Figure 6.3.1: A solution which violates the odd set inequalities for $m = 2$

$$\sum_{j\in J: j\in\cup_{u\in V: e\in\delta(u)\cap E'} J_u} c_j \geq \sum_{e\in E'} z_e \; \forall E' \subseteq E. \tag{6.3.18}$$

Constraint (6.3.17) states that given a subset of nodes, the coverage provided on all targets inside these nodes cannot exceed the weight of the edges incident to these nodes. Constraint (6.3.18) indicates that given a fixed set of edges, the weights on those edges does not suffice to protect all the targets in nodes to which those edges are incident, *i.e.*, it is necessary to consider, in addition, the other edges which are incident to those nodes. Figure 6.3.2 shows a solution that satisfies all but Constraints (6.3.10) and (6.3.11). The numbers on the nodes represent total coverage on targets in that node, $\sum_{j\in J_u} c_j$, and the numbers on the edges represent the coverage probabilities on the edges, z_e. The solution in Figure 6.3.2 violates (6.3.17) for $U = \{1, 2\}$. It is also not possible to find in this example an implementable defender strategy $x \in [0, 1]^{|I|}$ related to these variables z and c.

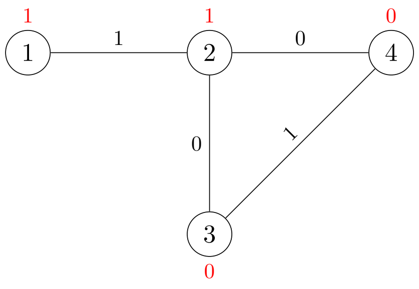


Figure 6.3.2: A solution which violates (6.3.17) for $U = \{1, 2\}$ and $m = 2$

It can in fact be proven that (FENCE$_{c,z,q,s,f,g}$) is a valid formulation for the SSG by showing that it is equivalent to (D2$_{x,q,s,f}$) in the sense that a feasible solution to one leads to a feasible solution to the other with the same objective value and viceversa. Given a feasible solution to (D2$_{x,q,s,f}$), one can construct a feasible solution to (FENCE$_{c,q,z,s,f,g}$), with same objective value, through Constraints (6.3.1)-(6.3.3). Conversely, given a feasible solution to (FENCE$_{c,q,z,s,f,g}$), a feasible solution to (D2$_{x,q,s,f}$) can be obtained relying on the fact that the cardinality constrained matching polytope is integral, [Schrijver, 2003]. The formal proof of the validity of $(\mathrm{FENCE}_{c,z,q,s,f,g})$ is omitted here. Next, we discuss the recovery of an implementable defender strategy x from an optimal solution to $(\mathrm{FENCE}_{c,q,z,s,f,g})$ in more detail.

6.3.2 Recovering an implementable defender mixed strategy

Given $(c^*, z^*, q^*, s^*, f^*, g^*)$ an optimal solution to $(\mathrm{FENCE}_{c,z,q,s,f,g})$, we want to recover an implementable defender mixed strategy x^* which complies with the optimal solution $(c^*, z^*, q^*, s^*, f^*, g^*)$. We propose two sampling methods in this section.

First sampling method First, note that from Constraints (6.3.8) and (6.3.9), $z^* \in conv(\mathcal{M}_m)$. The cardinality constrained matching polytope is integral, [Schrijver, 2003]. Therefore, there is a finite set of integer m-matchings $\mathrm{M}_{z^*} \subseteq \mathcal{M}_m$ such that we can express $z^* = \sum_{w \in \mathrm{M}_{z^*}} \lambda_w w$, where λ_w, $w \in \mathrm{M}_{z^*}$ are the weights in a convex combination of the integer m-matchings $w \in \mathrm{M}_{z^*}$ so that $\sum_{w \in \mathrm{M}_{z^*}} \lambda_w = 1$ and $\lambda_w \geq 0 \; \forall w \in \mathrm{M}_{z^*}$. Given this decomposition of z^*, we perform the following construction much in the spirit of Algorithm 1 discussed in Section 2.2.1.

Step 1. For every edge $e \in E$, consider a column of height 1. Divide each column into $|\mathrm{M}_{z^*}|$ horizontal segments–one for each integer m-matching $w \in \mathrm{M}_{z^*}$. The corresponding width of the segment across the columns, associated to matching $w \in \mathrm{M}_{z^*}$ is λ_w that matching's weight in the convex combination described above. For each edge $e \in E$, block out the segments on that edge's column that correspond to m-matchings that do not involve edge $e \in E$ (marked 'NO' in Figure 6.3.3).

Step 2. For every edge $e \in E$, further subdivide the area in its column that has not been marked 'NO' according to the values g^*_{ej} for all $j \in J_e$. Since $\sum_{j \in J_e} g^*_{ej} = z^*_e = \sum_{w \in \mathrm{M}_{z^*}} \lambda_w w_e$ from (6.3.11) and the decomposition of z^*, it follows that the described partition uses all the area in the column of e that has not been blocked out.

Step 3. Define x^* by identifying all the minimum constant-width horizontal sections after drawing horizontal lines across all the columns along each segment's subdivisions. Each horizontal section, of corresponding width $x^*_{(y,w)}$, is contained in a matching $w \in \mathrm{M}_{z^*}$ and for each edge e in the matching ($w_e = 1$), includes part of some $g^*_{ej_e}$. The indicator vector $y \in \{0,1\}^{|J|}$ in $x^*_{(y,w)}$, refers to the protected targets $\{j_e\}_{\{e:w_e=1\}}$ in that strategy.

For example, in Figure 6.3.3, we notice that the matching w^2 includes edges $e_1, e_3, \ldots e_{|E|}$. Furthermore, a constant-width horizontal section (shaded) can be identified using portions of $g^*_{14}, g^*_{33}, \dots, g^*_{|E|11}$. This shaded horizontal section corresponds to the strategy that implements pure matching w^2 and offers protection to targets $4, 3, \dots, 11$ with a probability equal to the width of the segment $x^*_{(y,w^2)}$.

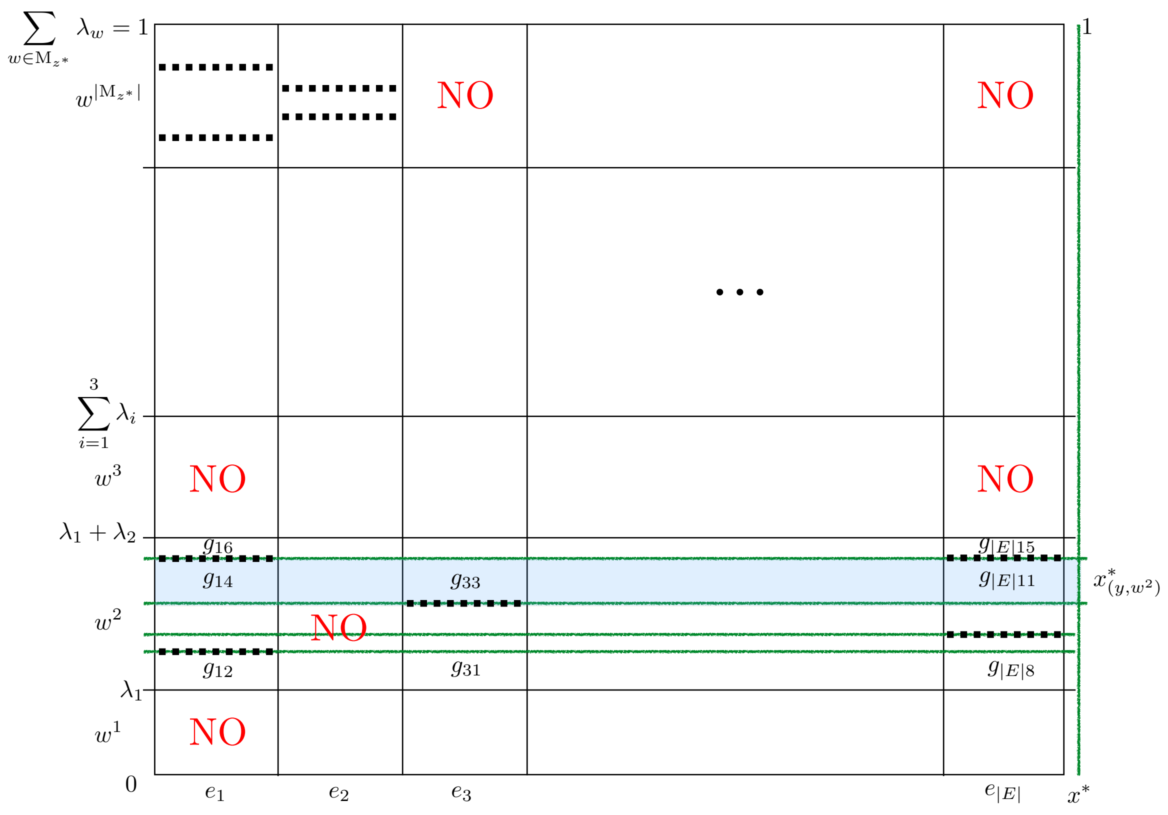


Figure 6.3.3: Box method to retrieve an implementable mixed strategy x^*

Lemma 6.3.1. *The box procedure described above is well defined.*

Proof. First, we show that all the (y, w) identified in Step 3, satisfy $(y, w) \in I$. By construction, we have that $w \in \mathcal{M}_m$. In addition, for each edge $e = (u_e, v_e) \in E$ used in w, we identify a single target $j_e \in J_{u_e} \cup J_{v_e}$ so that $y_{j_e} = 1$. Therefore, the vector y considers exactly m targets to be protected. Further, the m targets considered are distinct, *i.e.*, there can be no overlap between g^* variables involving the same target in different columns. If there were an overlap, it would imply that a node has at least two incident edges in an m-matching, which is not the case. Finally, for any $U \subseteq V$, we have that

$$\sum_{j \in \cup_{v \in U} J_v} y_j \leq \sum_{\substack{j_e \in J_{u_e} \cup J_{v_e} \\ \{u_e, v_e\} \cap U \neq \emptyset}} y_j = \sum_{e : \{u_e, v_e\} \cap U \neq \emptyset} w_e = \sum_{e \in E(U) \cup \delta(U)} w_e.$$

This proves that $(y, w) \in I$. Also, by construction, the horizontal segments corresponding to defender strategies (y, w), cover the box completely since the entire area of a column associated to an edge $e \in E$ that is not blocked out is covered by some $g^*_{e j_e} \geq 0$. This implies that Step 3 can be done at any height of the box and, therefore, that the values $x^*_{(y,w)}$ constructed in Step 3 are non-negative and add up to 1. They thus form a probability distribution over $(y, w) \in I$. ■

A small example detailing the box method procedure on a small instance can be found in Section A.6 in the appendix.

The box method construction relies heavily on being able to decompose z^*, a fractional matching of size m, as a convex combination of pure matchings of size m. [Schrijver, 2003] guarantees that it can be done because the cardinality constrained matching polytope is integral. Further, Carathédory's convex hull theorem guarantees that one needs at most $|E| + 1$ elements from $\mathcal{M}_m$ to define the decomposition of z^*.

In order to construct the decomposition, we provide an algorithmic Dantzig Wolfe approach, [Dantzig and Wolfe, 1960]. Let $\mathcal{M}^t_m$ be the set of matchings considered at step t of the procedure. The master problem can be stated as:

$$(MP) \quad \text{Min} \quad \sum_{e \in E} Y_e \tag{6.3.19}$$

$$\text{s.t.} \quad Y_e + \sum_{i \in \mathcal{M}^t_m} \theta^i_e \lambda_i = z^*_e \qquad \forall e \in E \tag{6.3.20}$$

$$\sum_{i \in \mathcal{M}^t_m} \lambda_i = 1 \tag{6.3.21}$$

$$\lambda_i \geq 0 \qquad \forall i \in \mathcal{M}^t_m, \tag{6.3.22}$$

$$Y_e \geq 0 \qquad \forall e \in E, \tag{6.3.23}$$

where Y_e are auxiliary variables to minimize. The parameters θ^i_e for all $e \in E$ and $i \in \mathcal{M}^t_m$ indicate whether or not edge e is present in the i-th matching. Let π_e and σ the optimal dual variable associated to constraint (6.3.20) and (6.3.21) respectively. Then, the reduced costs of any new column generated is given by:

$$r_i = 0 - \sum_{e \in E} \pi_e \theta^i_e - \sigma \tag{6.3.24}$$

So the problem of adding a new column can be stated as a maximum weight matching problem where the matchings are of size m and the weights are $\{\pi_e\}_{e \in E}$. If the optimal cost is greater than $-\sigma$ a new matching θ^i is added to $\mathcal{M}^{t+1}_m$. The algorithm stops when there is no new column to be added or the objective function of (MP) is equal to zero (which will happen in at most $|E|+1$ iterations), and $(\lambda, \{\theta^i\}_{i \in \mathcal{M}_{t^*}})$ are the weights and matchings that allow z to be decomposed as a convex combination. We implement a warm start using a greedy type algorithm that, in many cases, provides an optimal decomposition.

The box method described above, is an exact method to recover an implementable mixed strategy for the defender. It is exact in that the recovered mixed strategy complies with the optimal coverage probabilities returned by (FENCE$_{c,z,q,s,f,g}$). Obtaining the decomposition of the fractional matching z–a crucial part of the algorithm–is not without some computational effort. We propose next a simpler approximate sampling method which allows to recover an implementable mixed strategy x^* given optimal coverage probabilities (c^*, z^*).

Second sampling method We propose a two-stage approximate sampling method to recover a defender strategy. In Section 6.4, we discuss the accuracy of the method. Given z^*, we discard all the edges in E such that $z_e = 0$. We then select m distinct edges according to a uniform random variable $U(0, m)$. This, in itself, could provide edges that do not form a matching. Hence, we need to be a bit more subtle. Let M be the set of m edges sampled. Now, we solve the following optimization problem:

$$\begin{aligned}
\text{Max} \quad & \sum_{e \in M} z_e^* w_e \\
\text{s.t.} \quad & \sum_{e \in M} w_e = m, \\
& \sum_{e \in \delta(v)} w_e \leq 1 && \forall v \in V, \\
& w \in \{0,1\}^{|E|}.
\end{aligned}$$

Out of all possible matchings of size m, the objective function guarantees that we pick a maximum weight matching. The optimization problem either returns an optimal solution, in which case the edges in M admit a matching of size m, or, the problem is infeasible and such a matching cannot be constructed. If the problem is infeasible, we sample a new edge which we add to the set M and we reoptimize the optimization problem above. We proceed in this iterative fashion until we construct a matching of size m. This algorithm will produce a matching in at most $|E| - m$ iterations, since the original graph admits a matching of size m. The sampled matching respects the optimal coverages on edges if the procedure returns the required matching after one iteration. Otherwise, the matching deviates from the optimal coverage.

Having obtained M^*, the sampled matching of size m, the second stage of our sampling consists in sampling an allocation of resources to targets that satisfies the optimal coverage probability on the targets returned by our formulation. We discard targets j that belong to unpaired nodes. For each target j that belongs to a paired pair of nodes, say u and v, we normalize their coverage probability by the weight of the total coverage provided by the optimal coverage vector c^* in the two nodes u and v that are paired and denote it by $\bar{c}_j^*$:

$$\bar{c}_j^* = \frac{c_j^*}{\sum_{j \in J_u \cup J_v} c_j^*} \qquad \forall (u,v) = e \in M.$$

This way, we ensure that one resource is available per paired pairs of nodes. The implementable strategy is then composed by sampling over the newly constructed $\bar{c}^*$.

6.4 Computational experiments

In this section we run computational experiments to explore the quality of the approximate two-stage sampling method in Section 6.3.2 which produces an implementable defender strategy given an optimal solution to (FENCE$_{c,z,q,s,f,g}$). Further, we analyze the performance of the proposed formulation (FENCE$_{c,z,q,s,f,g}$) against solving the game that results from explicitly enumerating the defender pure strategies with GSG MILP formulation (D2$_{x,q,s,f}$).

6.4.1 Performance of the two-stage sampling method

In Section 6.3.2 we present two sampling methods that return a defender implementable strategy. The first method is exact in that the strategy returned agrees with the optimal coverage distributions on edges and targets returned by (FENCE$_{c,z,q,s,f,g}$). The second sampling method, is a two-stage approximate sampling method as it first samples from the edge coverage distribution and then, based on this first stage sampling, samples from the coverage distribution on the targets. The first sampling method, although exact, incurs on more computational load than the two-stage sampling method. In this section we study the accurateness of the two-stage sampling method.

Consider the optimal coverage distribution over the edges of the graph G, z^*. Now, construct an estimated edge coverage distribution $\hat{z}$ as follows. Sample $i = 1, \ldots, N$ matchings of size m according to the first stage of the two-stage sampling method. In our experiments $N = 1000$. For each edge $e \in E$, the estimated coverage on that edge is given by $\hat{z}_e = \frac{1}{N} \sum_{i=1}^{N} z_e^i$, where $z_e^i \in \{0,1\}$ depending on whether or not edge $e \in E$ was sampled in sampling $i \in \{1, \ldots, N\}$.

We use the Kullback-Leibler divergence [Kullback and Leibler, 1951] to measure the closeness of the two distributions, z^* and $\hat{z}$ over instances with n nodes where $n \in \{5, 25, 50, 100\}$. For each instance size, we generate 30 estimations $\hat{z}$. The results are shown as box plots in Figure 6.4.1.

Observe that the Kullback-Leibler distance between z^* and $\hat{z}$ is very small, below 0.2 over all instances, which is a good indicator that $\hat{z}$ is a good estimator for z^*. In particular

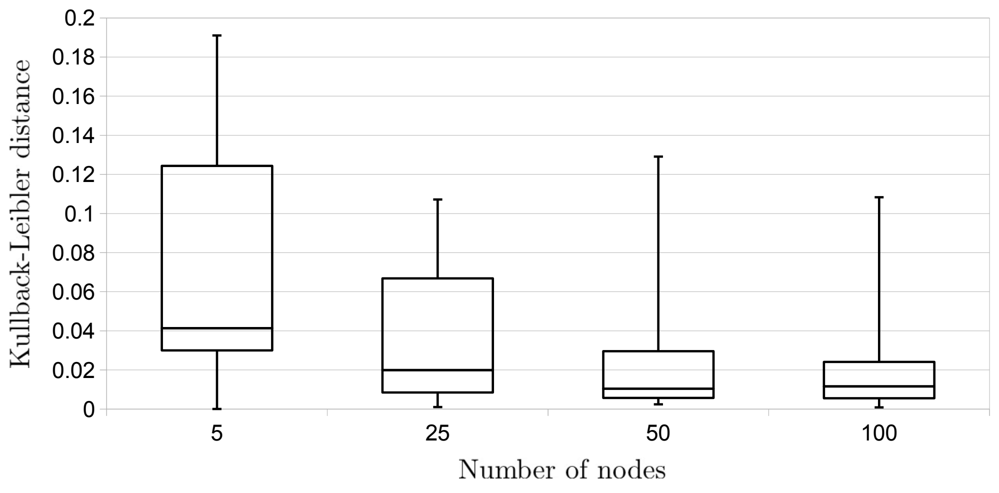


Figure 6.4.1: Comparison of the estimated distribution $\hat{z}$ produced by the two-stage sampling method to the optimal distribution z^* returned by (FENCE$_{c,z,q,s,f,g}$)

we observe that the larger the set of nodes in an instance the better an estimator $\hat{z}$ appears to be. For instances with 100 nodes, most of the $\hat{z}$ have a Kullback-Leibler distance to z^* which is below 0.02.

The same analysis to measure the closeness of the optimal coverage distribution over targets, c^*, to an estimated distribution $\hat{c}$ obtained from multiple samplings in the second stage of the two-stage sampling method, leads to analogous conclusions.

Therefore, we can safely conclude that the implementable strategy returned by the two-stage sampling approach proposed is a good approximation to the optimal coverage probabilities on edges and targets returned by (FENCE$_{c,z,q,s,f,g}$).

6.4.2 Performance of (FENCE$_{c,z,q,s,f,g}$)

We study the performance of the proposed formulation (FENCE$_{c,z,q,s,f,g}$) on randomly generated instances against solving the GSG that results in explicitly enumerating all the defender pure strategies with (D2$_{x,q,s,f}$).

The instances considered for these experiments are generated as follows. We generate random graphs with n nodes, where $n = \{5, 6, \dots, 22\}$ and edges such that the graphs are connected and that, on average, each node has degree three. Further, we consider four targets inside each node. The set of targets, J, is thus of size $|J| = 4n$. We consider three types of attackers in these games. We then uniformly generate payoff values for the defender and each attacker type by considering for each player, rewards $D^k(j|c)$ and $A^k(j|u)$ for all $k \in K$ and $j \in J$ in the range $[0, 100]$ and penalties $D^k(j|u)$ and $A^k(j|c)$ for all $k \in K$ and $j \in J$ in the range [-100,0].

In Figure 6.4.2 we show the running time of the different solution methods over the generated instances. On the left hand side, we consider instances where the number of pairings is two, *i.e.*, four nodes need to be paired. On the right hand side, we consider instances where the number of pairings is three, *i.e.*, six nodes need to be paired, and we only go up to graphs with 20 nodes. In both plots, for each instance size we record the average solution time of 30 randomly generated instances.

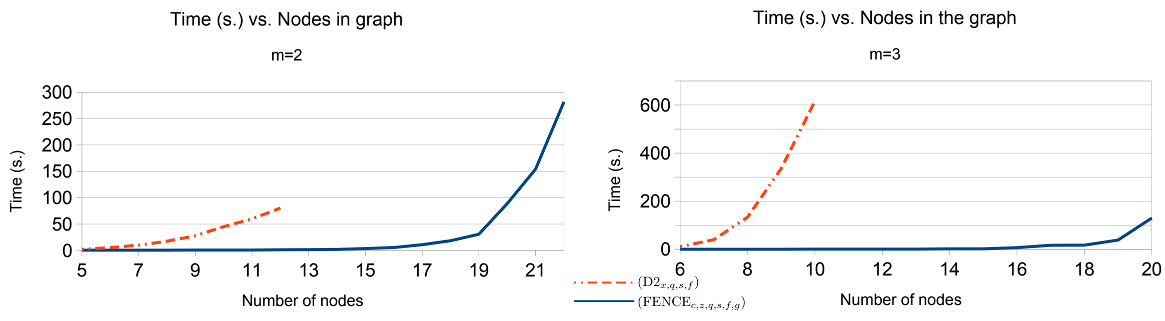


Figure 6.4.2: $(\mathrm{FENCE}_{c,z,q,s,f,g})$ and $(\mathrm{D2}_{x,q,s,f})$ on randomly generated instances

As one can see, $(\mathrm{FENCE}_{c,z,q,s,f,g})$ solves the instances much faster than $(\mathrm{D2}_{x,q,s,f})$. The set of defender strategies grows exponentially and $(\mathrm{D2}_{x,q,s,f})$ can only explicitly enumerate these strategies for very small graphs of less than 12 nodes. It is also interesting to note that our full compact formulation, $(\mathrm{FENCE}_{c,z,q,s,f,g})$, behaves remarkably well for graphs of up to 18 nodes.

6.5 Conclusions

The computational results shown in the previous section can only be considered preliminary based on the sizes of the instances being considered. In order to make $(\mathrm{FENCE}_{c,q,z,s,f,g})$ more efficient, one must separate the exponentially many odd-set inequalities (6.3.9). This issue has been tackled in the Ph.D. thesis [Bucarey, 2017], where the author implements a separation procedure detailed in [Padberg and Rao, 1982]. Separating the odd-set inequalities in $(\mathrm{FENCE}_{c,q,z,s,f,g})$, allows to efficiently solve instances over graphs with up to 35 nodes where 10 edges are selected and 10 targets patrolled within the designated pairs. The solution times for these instances are consistently under 100 seconds, showing an enhanced performance over $(\mathrm{FENCE}_{c,q,z,s,f,g})$ without the cut separation.

Further, as in Chapter 4, linearizing variables that represent the product of the c and the q variables could be introduced in $(\mathrm{FENCE}_{c,z,q,s,f,g})$ to avoid having to include Constraints (6.3.12). Also Constraints (6.3.13) could be strengthened as in Chapter 4, eliminating the need for big M constants in the formulation, *i.e.*, the network SSG formulation could be built on (MIP-p-S$_{q,y}$) rather than on (ERASER$_{c,q,s,f}$). This would lead to a new MILP formulation with a heavier LP relaxation but a better quality LP bound than (FENCE$_{c,q,z,s,f,g}$). It would then be possible to explore the possibility of adapting the decomposition methodology described in Chapter 5 to the different network SSG formulations. One would expect these techniques to allow for a sizeable scaling in the sizes of the instances that could be solved efficiently.

In addition, one could think of meaningful extensions to the underlying problem. For example, the proposed formulation only allows one security resource per paired pair of nodes. This is a way of modelling the limited availability of security resources in a network where all nodes are equally important. A natural extension is to consider that different pairings of nodes have different numbers of security resources available to protect targets within those pairings. This extension would account, for example, for different levels of importance among the nodes in the network.

Chapter 7

Case Study: Border Protection

7.1 Introduction

A problem faced by many countries is that of securing their national borders. The United States Department of Homeland Security states as a primary objective that of "protecting [the] borders from the illegal movement of weapons, drugs, contraband, and people, while promoting lawful entry and exit" claiming it is "essential to homeland security, economic prosperity, and national sovereignty" [Department of Homeland Security of the United States, 2016].

Recent events have led to uncontrolled massive migrant flows into the European Union originating from areas such as Western and Southern Asia, Africa and the Western Balkans, [Financial Times, 2015]. The European Union responded to this crisis by creating the European Border and Coast Guard in October 6th, 2016, whose main task is to "provide integrated border management of [the EU's] external borders [so as to] ensure effective management of migration flows and provide a high level of security for the EU" [Council of the European Union, 2016].

Additionally, transnational crime involving the illicit flow of drugs, illegal entry of people and smuggling of contraband also hits countries hard. In Chile, a country with 7,801 km of land borders, transnational crime is a scourge. The Central Intelligence Agency marks Chile as a transshipment country for cocaine destined for Europe and indicates that "domestic cocaine consumption is rising, making Chile a significant consumer of cocaine", [Central Intelligence Agency, 2008]. Illegal entry of people is a highly lucrative crime for organized groups who make up to US$2000 per illegal alien. The year 2015 saw an increase of 124.5% in illegal entry crimes with respect to 2014, [El Mercurio On-Line, 2016, Fiscalía de Chile, Ministerio Público de Chile, 2014]. Further, over the years 2010-2014 contraband estimated at over US$ 4 million and just shy of a 1000 stolen motor vehicles were intercepted at the

Chilean borders, [Fiscalía de Chile, Ministerio Público de Chile, 2015].

Vast stretches of land to be thoroughly surveilled and limited manpower are among the chief problematics faced by border patrol agencies throughout the globe. One way to overcome the lack of manpower in a given border region is to combine resources from different locations in a joint effort to coordinate a global patrol plan. It is crucial to balance the effectiveness of a global border plan with the cost and difficulty of locally coordinating resources in undermanned areas. The European Border and Coast Guard lists as one of its prime objectives "organising joint operations and rapid border interventions to strengthen the capacity of the member states to control the external borders, and to tackle challenges at the external border resulting from illegal immigration or cross-border crime".

The contributions of this chapter involve developing a Stackelberg methodology to provide strategic security patrols to Carabineros de Chile, the Chilean national police, along the borders of Chile's northernmost region, Arica y Parinacota. We propose a parameter estimation technique to construct the payoffs of an SSG based on past crime data as well as on other geographical and social factors, and build a software for Carabineros that implements the parameter estimation and solves an SSG to ultimately determine the optimal actions that Carabineros should implement. We perform a sensitivity analysis on the optimal solutions of our Stackelberg formulation to show its robustness against minor changes in the parameter estimation technique and discuss an adequate evaluation of the deployed security system.

This chapter is divided as follows. In Section 7.2, we formally describe the problem. In Section 7.3, we discuss the methodology used to generate the parameters of the game so as to best model the real life problem. In Section 7.4, we discuss the software developed for Carabineros and in Section 7.5, we perform some computational experiments to show the robustness of our resolution strategy against minor changes in the game parameters. In Section 7.6, we discuss the evaluation of the deployed security system. We close the chapter off in Section 7.7 with some conclusions.

Figure 7.1.1: A Carabinero surveils

Figure 7.1.2: Harsh border landscape

7.2 The border patrol problem

In the XV region of Chile, Arica y Parinacota, Carabineros are faced, primarily, with three different types of crime, namely, drug trafficking, contraband and illegal entry. In order to minimize the free flow of these types of crime across their borders, Carabineros organizes both day shift patrols and night shift patrols along the border, following different patterns and satisfying different requirements.

We are concerned with the specific actions that Carabineros can take during night shift patrols. The XV region is divided into several police precincts. Due to the vast expanses of harsh landscape that needs to be covered and the lack of manpower to constantly cover the entire border, for the purpose of the defender actions under consideration, a number of these precincts–those considered as border precincts–are paired up at the beginning of the month. Further, Carabineros are aware of twenty locations along the border of the region that can serve as vantage points from where to conduct surveillance with technical equipment such as night goggles and heat sensors. A night shift action consists in deploying a joint detail with personnel from two paired precincts to conduct vigilance from 22h00 to 04h00 at a vantage point located within the territory of the paired precincts. Due to logistical constraints Carabineros deploys a joint detail from every precinct pair to a surveillance location once a week.

Carabineros wants a schedule indicating the optimal deployment of details to vantage points for the night shifts, on a given week. Figure 7.2.1 depicts a defender strategy in a game with 3 pairings and 10 locations. Table 7.2.1 shows a tabular representation of the implemented strategy.

Pairing\Outpost	1	2	3	4	5	6	7	8	9	10
Pairing 1		M								
Pairing 2					M					
Pairing 3										Su

Table 7.2.1: Tabular representation for the feasible schedule in Figure 7.2.1

It becomes immediately clear that this problem fits the setting described in Chapter 6 where we have a graph $G = (V, E)$ with V, the set of police precincts, E indicates the allowed pairings of precincts, which in this case is fixed by Carabineros, and there is a set of targets $j \in J$ inside the police precincts which correspond to the vantage points that need to be manned. In this setting, K corresponds to the number of crime types: drug trafficking, contraband and illegal entry. Note that since the pairings are fixed, the game is separable

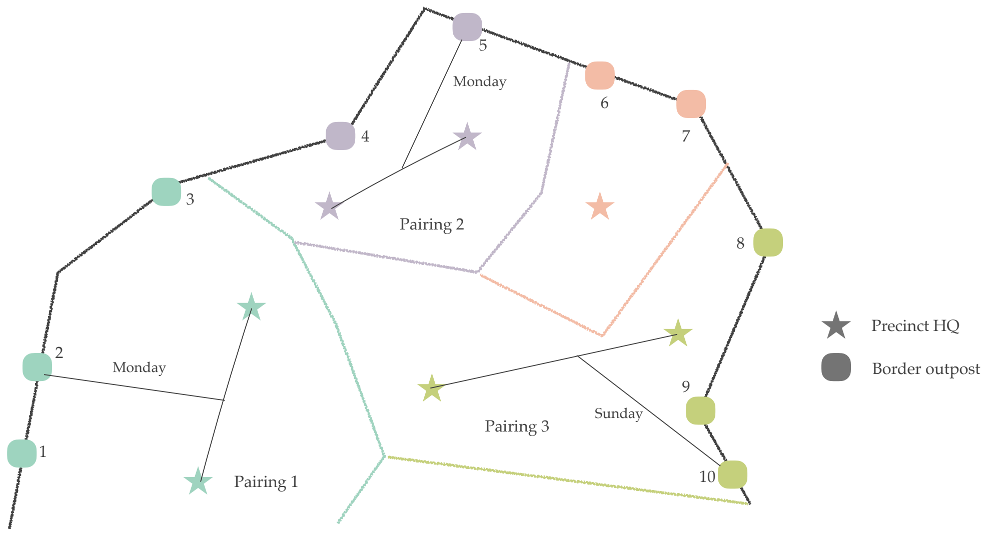


Figure 7.2.1: Feasible schedule for a week

into a different standard SSG within each paired pair of precincts. Within a paired pair of precincts, the defender has a single resource and his pure strategies consist in covering one target on a given day of the week. Given a coverage strategy over the targets in a given paired pair of precincts, a criminal of type $k \in K$ plays the game with probability π^k and tries to cross the border through the vigilance outpost $j \in J$ and on the day of the week which maximizes his payoff. We rely on the SSG MILP formulation (ERASER$_{c,q,s,f}$) to solve the game at each paired pair of precincts because it is the best performing Stackelberg security formulation for small sized instances as discussed in Chapter 4.

It remains to construct the payoffs of the game for the problem described. To that end, Carabineros supplied us with arrest data in the region between 1999 and 2013 as well as other relevant data. We explore a payoff generation methodology next.

7.3 Parameter generation: Constructing the game

An accurate estimation of the payoffs for the players is one of the most crucial factors in building a Stackelberg model to solve a real life problem. For each target in the game, we need to estimate 12 different values corresponding to a reward and penalty for Carabineros and the attacker for each type of crime $k \in K$.

We tackle this problem in several steps. First, we use QGIS [QGIS, 2009], an open source geographic information system, to determine what we call action areas around each vantage point provided by Carabineros, based on the visibility range from each outpost. Such an action area represents the range of a detail stationed at a vantage point, *i.e.*, the area within which the detail will be able to observe and intercept a criminal. For further information on how the action areas are constructed, refer to Section A.7 in the appendix.

Further, consider, for each type of crime $k \in K$, a network $G^k(V^k, E^k)$ that models that type of crime's flow from some nodes outside the border to some nodes inside the border, crossing the border precisely through the action areas previously defined. As nodes of origin for the different types of crime, we considered several main cities in Perú and Bolivia and as destination nodes we considered the locations inside Chile where Carabineros has performed an arrest of that type of crime. In order to have a more manageable sized network, we considered a clustering of these destination nodes. We later show that our methodology is robust versus small changes in the number of cluster nodes.

Specifically, for a crime of type $k \in K$, let us define $S^k \subset V^k$ as the nodes of origin situated outside the borders, $F^k \subset V^k$ as the nodes of destination and J as the set of action areas along the border. Each destination node, $f \in F^k$, resulting from a clustering process is then assigned a demand $b(f)$ which corresponds to the number of destination nodes which are contracted into f. For each $k \in K$, the edge set E^k is constructed as follows. All nodes of origin are linked to all action areas. These areas are then linked to all of the destination nodes for crime $k \in K$. Figure 7.3.1 is a representation of such a network. The nodes to the right represent the points of origin of crime and the three nodes to the left are clusters of destination nodes for those crime flows. Note that crime enters the country through the four action areas marked as squares along the border.

We propose the following attractiveness parameter for a given action area $j \in J$ for a criminal of type $k \in K$ attempting to move from node $s \in S^k$ to node $f \in F^k$ through action area j:

$$U_{sf}^{j} = \frac{\text{Kilometers of roads inside action area j}}{d_{sj} + d_{jf}},$$

where d_{uv} is the distance in kilometers between nodes u and v. This attractiveness parameter is proportional to the total length of roads inside a given action area and it is inversely proportional to how much an attacker moving from s^k to f^k has to travel in order to cross the border through area j. Similar techniques are used in the domain of competitive location, [Suárez-Vega et al., 2011].

We model the flow of crime $k \in K$ through a single route from $s \in S^k$ to $f \in F^k$ passing

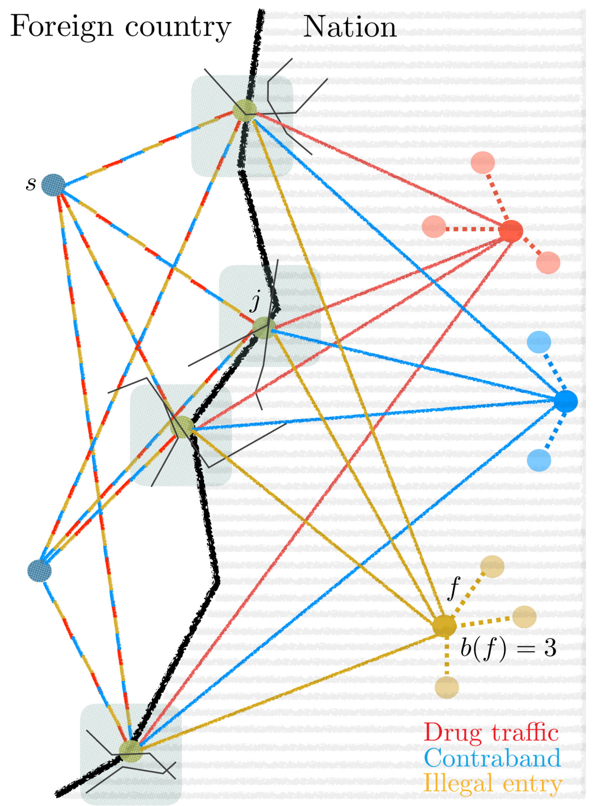


Figure 7.3.1: Three crime flow networks, one per type of crime

through $j \in J$ as follows:

$$x(s,j,f,k) = \frac{e^{\lambda U^j_{sf}}}{\sum_{s' \in S^k} \sum_{j' \in J} e^{\lambda U^{j'}_{s'f}}} \cdot b(f).$$

The flow of crime $k \in K$ through a route (s,j,f) is expressed as a proportion with respect to the flow of crime $k \in K$ through all routes leading into the same destination point $f \in F^k$. The parameter $\lambda \in \mathbb{R}_+$ provides a proxy of how the defender expects crime to behave. A value of $\lambda = 0$ means that crime $k \in K$ between any node of origin and destination would distribute itself evenly among the different action areas. A high value of λ, however, would be consistent with a flow of that type of crime through those action areas $j \in J$ with a higher attractiveness parameter U^j_{sf}. It follows that the total flow of crime of type $k \in K$ through $j \in J$ can be computed by summing over all origin nodes $s \in S^k$ and all destination nodes $f \in F^k$:

$$x(j,k) = \sum_{s \in S^k} \sum_{f \in F^k} \frac{e^{\lambda U^j_{sf}}}{\sum_{s' \in S^k} \sum_{j' \in J} e^{\lambda U^{j'}_{s'f}}} \cdot b(f) \quad \forall j \in J, \forall k \in K.$$

Based on this parameter, we propose the following values for the players' payoff values:

$$\begin{aligned}
A^k(j|u) &= x(j,k)\cdot AG(k) && \forall j \in J, \forall k \in K,\\
A^k(j|c) &= -x(j,k)\cdot OC(k) && \forall j \in J, \forall k \in K,\\
D^k(j|c) &= 0 && \forall j \in J, \forall k \in K,\\
D^k(j|u) &= -x(j,k)\cdot AG(k) && \forall j \in J, \forall k \in K,
\end{aligned}$$

where $AG(k)$ denotes the average gain of successfully committing crime $k \in K$, and $OC(k)$ the opportunity cost of being captured while attempting to perpetrate a crime $k \in K$. Note that the reward Carabineros perceives when capturing a criminal is 0, irrespective of the crime. Carabineros will only be penalized when a crime is successfully perpetrated on their watch. These values where calculated following open source references [Comisión Económica para América Latina y el Caribe, 2000, Aduanas de Chile, 2016, Ministerio del Trabajo y Previsión Social, 2016] and were then vetted by Carabineros to ensure that our estimates were realistic.

7.4 Building software for Carabineros

We provide Carabineros with a graphical user interface developed in PHP to determine optimal weekly schedules for the night shift actions for a set of border precincts in the XV region of Chile.

The software provided for Carabineros is divided into two parts: a first part, devoted to the parameter generation of the game according to the indications of the previous section, and a second part, which solves for the optimal deployment of resources. We discuss the two parts separately.

7.4.1 Parameter estimation software

The objective of the parameter estimation software is to construct the payoff matrices for the Stackelberg security game. This software allows for the matrices to be updated when new criminal arrests are recorded in Carabineros' database. The input for this software is a csv file with arrest data which is uploaded to the software. The main screen of the software has a map of the region to the left and the following options to the right:

- *Crimes*: Shows all criminal arrests in the area, color-coded according to the type of crime.
- *Nodes of origin*: Shows the nodes of origin used in the networks constructed to determine the crime flow through the action areas.

- *Action areas*: Shows the different action areas considered.

- *Cluster*: Clusters the criminal arrest points and constructs the crime flow networks joining nodes of origin, action areas and the clustered arrest points, which are the destination nodes for the different types of crime. It displays the payoff matrices for the different action areas.

- *Input file and update*: Allows to upload a csv file with arrest data. One then re-clusters to obtain new destination points and to construct the new crime flow networks that lead to new payoff matrices.

7.4.2 Deployment generation software

The deployment generation software is the part of the software that optimizes the SSG and returns the implementable patrol strategy for Carabineros. The user is faced with a main screen that has a map of the region with the action areas along the border to the left and several options to the right. The action areas are color-coded so that the user sees which action areas belong to which pair of precincts. Clicking on an action area automatically shows the payoff values for that area. The values can be modified on-screen although this is not advised. The user can additionally select the number of resources in a given pair of precincts. An increased number of resources can be used to model that a joint detail can perform a night-shift patrol more than once a week. Further, the user can select the number of weekly schedules sampled from a given optimal coverage strategy, allowing to extend the weekly schedule to, say, a monthly schedule. Once all the parameters are set, clicking on 'solve' returns the desired patrol schedule like the one shown in Table 7.2.1.

Once a patrol strategy has been returned, the user can perform several actions. If the patrol is not to the liking of the planner, he can re-sample. This produces a different patrol strategy from the same optimal coverage probabilities returned by the SSG formulations. The user can further impose different types of constraints on each paired pair of precincts. For a pair of paired precincts, the user can force a deployment to an action area and can force a deployment on a given day of the week. He can similarly impose that an action area not be protected or that no deployments happen on a given day of the week. In addition, he can impose that at least one of a subset of targets is protected on a given day of the week or that a deployment to a specific target happens on at least one of a subset of days of the week. Sampling under these constraints will produce a deployment strategy that satisfies the user's demands.

7.5 Computational tests

We study the robustness of the solutions produced by our software to variations in the payoff matrices. Specifically, we study the robustness of our method against variations of two key parameters in the payoff generation methodology: λ which models the defender's belief on how crime flows across the border and $b(f)$ which indicates the number of nodes clustered into a given destination node f. Equivalently, one can consider variations in a vector $h = (h_1, h_2, h_3)$ which determines the number of cluster nodes for crimes 1, 2 and 3 respectively. We study the effects of variations in the parameter λ and in the vector of cluster nodes h separately.

As a base case, we generate payoffs for the players by setting $\overline{\lambda} = 50$ and $\overline{h} = (6, 6, 6)$. This appears reasonable given the size of the problem and distribution and number of arrests per type of crime in the XV region. Let $\lambda \in \Lambda = \{0.5\overline{\lambda}, 0.75\overline{\lambda}, 1.25\overline{\lambda}, 1.5\overline{\lambda}\}$ and $h \in H = \{(h_1, h_2, h_3) \in \mathbb{N}^3 : h_t = \overline{h}_t \pm s, t \in \{1, 2, 3\}, s \in \{0, 1, 2, 3\}\}$. We denote by $c(\lambda, h)$, the optimal coverage probabilities on the targets when the payoffs have been defined according to λ and h. Given two vectors $p, q \in \mathbb{R}_+^{|J|}$, we consider the usual distance function between them:

$$d(p, q) = \sqrt{\sum_{j \in J} (p_j - q_j)^2}.$$

We identify $\lambda^* \in \arg\max\{d(c(\overline{\lambda}, \overline{h}), c(\lambda, \overline{h}))\}$ and $h^* \in \arg\max\{d(c(\overline{\lambda}, \overline{h}), c(\overline{\lambda}, h))\}$ and plot $c(\overline{\lambda}, \overline{h})$, $c(\lambda^*, \overline{h})$ and $c(\overline{\lambda}, h^*)$.

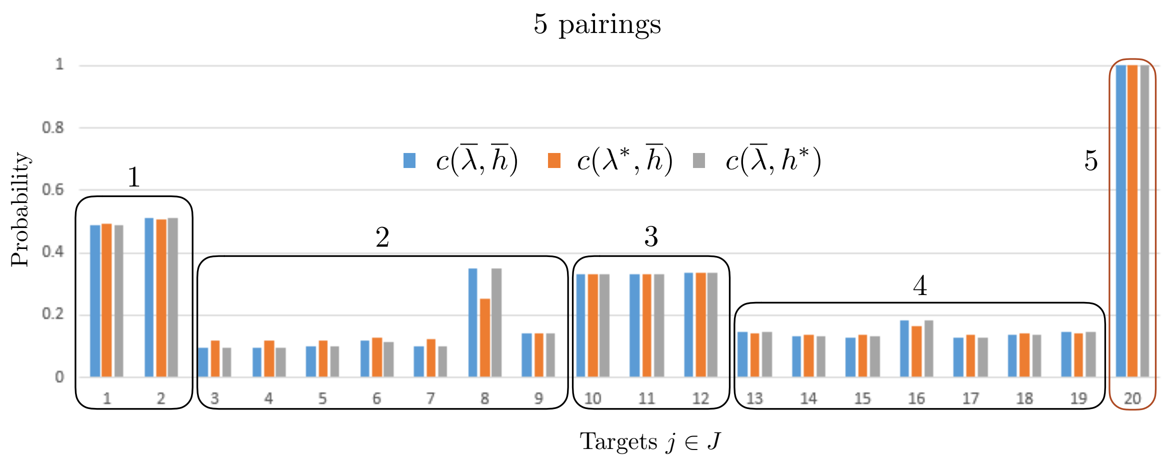


Figure 7.5.1: Robustness of our solution method to variations in the parameters λ and h

Figure 7.5.1 shows the optimal coverage probabilities $c(\overline{\lambda}, \overline{h})$, $c(\lambda^*, \overline{h})$ and $c(\overline{\lambda}, h^*)$ for the game with five paired police precincts and twenty targets on a single day of the week. One can see that the optimal probabilities are robust towards variations in the number of clusters. As one could expect, they are less robust to variations in the parameter λ.

Recall that a low value of λ constructs the payoff matrices under the assumption that crime distributes itself uniformly among the different action areas $j \in J$. It is therefore understandable that the optimal coverage probabilities reflect this by trying to cover the targets uniformly. On the other hand the optimal coverage probabilities tend to be more robust for higher values of λ.

7.6 Evaluating our deployed system

The final phase of a project whose aim it is to deliver a security system is normally an evaluation phase. A comprehensive evaluation of the software as a whole needs to be addressed so as to satisfy the client that the software created performs quantifiably better than whatever plausible alternative exists. Evaluating a deployed system, particularly in the security field, can be tricky for a number of reasons. A pertinent reference is [Tambe, 2011].

To comprehensively evaluate our security system, we break down the analysis of the system as a whole into smaller analyses of different aspects of said system. We discuss the adequateness of both the methodology and the software in Sections 7.6.1 and 7.6.2.

7.6.1 Evaluating the mathematical model

The following questions are meaningful:

1. How well does the proposed methodology model the real life problem? Is the solution concept employed realistic in this setting?
2. Is the proposed formulation simple to solve from an algorithmic/computational point of view? Is it apt?
3. How robust is the proposed formulation to changes in its key parameters?
4. What can be said about the quality of the proposed strategy?

To answer question 1, the SSG paradigm seems to be a good fit based on the strategic nature of the game that is being modeled. Carabineros acts first and criminals react to Carabineros' actions. An assumption is made that criminals can observe how Carabineros deploys its units over time and they will acquire perfect information on the probabilities of a unit being deployed to a specific outpost. Criminals are strategic and will try to maximize the payoff associated with their actions. Carabineros is also highly strategic and will try to influence the criminal's behavior by selecting the defensive strategy that provides Carabineros with the highest payoff, anticipating that the criminals will act on the knowledge of how Carabineros has deployed its units over time. It seems like a reasonable assumption that players have complete information on how they themselves value their different actions and also how their opponent values their own actions. Further, if the defender commits to a uniform random deployment of his resources, he is, by definition, unpredictable and no assumptions are necessary as to how the attacker values his actions but it is known that if the defender plays a Stackelberg equilibrium strategy, which is a weighted randomization that takes advantage of each player's valuations, he stands to gain a higher payoff.

In addition, note that it is appropriate that the border patrol problem is not zero-sum, *i.e.*, the amount by which one of the players wins doesn't have to coincide with the amount by which the other loses, since the criminal may derive some utility from trying to commit a crime, such as forcing the defender to incur in some operational costs, even if he is caught. Also, the parameter estimation methodology, which is crucial in generating games that capture the reality of the border patrol setting, was carefully vetted by domain experts.

Many aspects of the problem, such as operational costs or personnel available at the different precincts, have been left out of the model. Modelling a real life problem requires abstraction which inevitably leads to some loss of information. We strive to minimize the loss of precision by discerning which information is more relevant to the job at hand. Also, one risks poor solvability by overcomplicating a model.

To answer question 2, recall that the MILP SSG formulation used in our border patrol software is ($\text{ERASER}_{c,q,s,f}$). We saw in Chapter 4 that ($\text{ERASER}_{c,q,s,f}$) is a very fast formulation when solving smaller sized instances like the ones we tackle in the border patrol problem, making it an ideal choice for the patrol software.

To answer question 3, we conduct a sensitivity analysis of our solution method against variations in some of the parameters of the payoff estimation methodology. Unfortunately, it is well known that game theoretic models can be quite sensitive to payoff noise and arbitrary changes in the payoffs can lead to arbitrary changes in the optimal strategy. Carabineros were reassured, however, from our computational tests that our solution method is not highly volatile against minor changes in the payoff estimation parameters.

To answer question 4, first consider that the implementable strategy that one recovers from the optimal coverage probabilities through a sampling algorithm such as the box method (See Algorithm 1) agrees with the optimal coverage returned by the optimization model. Measuring the quality of a deployment schedule is not immediate. One might want to compare it against previous deployment schedules and consider differences and similarities between them. A schedule might give Carabineros the impression of being better because resources are used more often and are spread more across the map. A schedule will ultimately be considered better if it results in a higher arrest rate than others.

7.6.2 Evaluating the software

Our proposed software is appealing, simple to use but powerful and it presents output in a very clear and visual manner. Our software is as simple as possible, leaving all the technical and mathematical tools to be executed behind the scenes. It is also flexible, allowing the end-user to enforce constraints and update the arrests database, leading to a redefinition of the payoff matrices. Carabineros' appraisal of our software has been very satisfactory.

7.7 Conclusions

A crucial aspect in the evaluation procedure of a deployed system is its track record after being deployed. Our border patrol system has not yet undergone an on-the-field performance evaluation. Providing an evaluation mechanism for such a deployed system is challenging work in its own right. The authors in [Shieh et al., 2012], for example, conducted a so-called 'red'-team exercise to evaluate their patrol planner in a port scenario. A 'red'-team exercise consists in evaluating the performance of the security software by having a team of security professionals attempt a successful attack against the optimal patrols provided by the software. Such an analysis might help to better understand defender-attacker interactions which could lead to a more sophisticated patrol planner.

Further, our payoff estimation methodology could be enhanced in different ways. Temporal weighing of crime data would increase the relative importance of the more recent crimes. Our estimation methodology currently builds the attractiveness of the action areas for a certain type of criminal based on road density around the action area and distances to be travelled by the criminals from source to destination. Other environmental factors such as maximum altitude along a route, availability of shelter along a route or distance of settlements from a route could be taken into account to compute a more realistic attractiveness of an outpost.

In addition, it would be interesting to compare the solutions provided by the security software, where the pairings between the police precincts is predetermined by Carabineros, to the solutions provided by the compact MILP formulation ($\text{FENCE}_{c,z,q,s,f,g}$) presented in Chapter 6, where pairings between nodes are determined by decision variables in the model. One would expect the latter to provide a higher return for the defender. It could be useful to consider an intermediate situation where pairings are not predetermined, but rather than being free between adjacent districts, comply with some further meaningful restrictions.

Chapter 8

Conclusions and future work

In this dissertation we have studied Stackelberg games, both in a general and in a security setting. Stackelberg games are a game theoretical paradigm that models a situation of conflict between two players. It has been established that such games can be efficiently tackled through mathematical optimization. Stackelberg games are very versatile in that their use has extended to a wide variety of domains such as telecommunications, transportation, theory of incentives and most prominently, security. The field of Stackelberg game theory has thus received much attention from the research community over the last 10-15 years. A significant part of the research in this field of knowledge has been driven towards developing efficient solution methods to tackle real-life, and oftentimes large-scale, problems. This thesis represents a step forward in that direction.

8.1 Summary of main results

In Chapter 1 we have motivated the interest in Stackelberg games, as well as provided some context for the remaining chapters by briefly overviewing two important fields of knowledge which are deeply connected to this thesis: Game Theory and Bilevel Programming.

In Chapter 2 we have provided a formal description of GSGs and SSGs and show how they can be naturally modeled by means of Bilevel Programming. We have further provided an algorithm to recover an implementable GSG solution from a given SSG solution.

In Chapter 3 we have performed a review of the related literature with a particular interest in studies regarding the complexity of the games, the main computational challenges encountered when solving said games as well as the main solution methods used. We have further discussed extensions and ramifications of the Stackelberg methodology which have sparked interest among the research community.

Chapters 4 through 7 encompass the main contributions of the thesis. In Chapter 4, we have studied previously existing MILP formulations for general Stackelberg games and we have conducted a comparative study of the formulations with respect to the strength of their LP relaxation solutions. We have further specified a theoretical link based on variable projections which establishes a formal connection between general and security Stackelberg games. This link has been exploited to extend our study to Stackelberg security game formulations and to deduce a new tight MILP formulation for the security setting which outperforms competing formulations. The new formulation we have presented is the strongest, in the sense that the bound on the optimal value provided by its LP relaxation is the tightest, and we have seen that it can handle scale-ups in the size of the security instances much better than other security game formulations. We have experienced that, even though the tightest MILP formulation in each setting outperforms competing formulations, they are limited in the size of the instances they can solve because of their heavy LP relaxations.

In Chapter 5, we have addressed the challenge encountered in Chapter 4. We have thus exploited the problem structure to develop decomposition algorithms for the general and security settings which embed Benders cuts from the heavy but strong LP relaxations of the tightest formulations into a Cut and Branch solving scheme based on much sparser and weaker equivalent formulations. We have fine-tuned our decomposition algorithms by implementing different cut-loop stabilization procedures as well as different primal bound heuristics designed to enhance the solution process. We have further tested whether or not interrupting the cut generation in our cutting plane approaches before the root node is solved, might lead to faster solution times. We have compared the performance of our tuned decomposition algorithms against other solution methods from the literature, and have tested their scaling-up capabilities. Our methods have proven to scale better than competing state of the art approaches both for general and security games, allowing us to solve much larger instances efficiently in either setting.

In Chapter 6 we have analyzed a particular type of Stackelberg security game, defined on a network. The defender's strategy now needs to take into account a more global planning (a selection of m edges of the network) as well as more local planning (the deployment of the m resources to targets located at the nodes of the network) which is dependent on the edge selection. We have provided a valid MILP formulation for the problem with a polynomial number of variables and exponentially many constraints. We have further presented two different sampling methods, one exact and one approximate, that recover an implementable defender strategy given the optimal coverage distributions over edges and targets. Our computational tests have shown that the estimated coverages on edges and targets that are constructed by the approximate sampling method are close to the real coverages returned by the optimal solution provided by the formulation. Further, our tests have shown that the presented MILP formulation performs much better than attempting to solve the problem as a general bi-matrix game where the exponentially many defender pure strategies are explicitly specified.

In Chapter 7 we have presented a direct application of the Stackelberg problem described in Chapter 6 to tackle a real-life border patrol problem along the Chilean border. We have developed and implemented software to provide Carabineros with a Stackelberg-based tool with which to schedule the weekly deployment of security resources to discourage illegal border crossings. We have further provided a parameter estimation methodology, specific to the problem at hand, to automatically generate the game parameters so that our modelization of the problem captures its key aspects. Finally, based on our hands-on experience in this border patrol problem, we have provided some insights on thoroughly evaluating a deployed system such as the one presented in this chapter.

8.2 Perspectives

In this thesis we have made some progress in being able to efficiently solve large scale instances. The decomposition methods presented in Chapter 5 allow us to solve GSG instances with up to 95 5×5 payoff bimatrices as well as GSG instances with 23 10×10 bimatrices within a three hour time limit. In the security domain, within the same time limit, we can efficiently take on SSG instances where a defender has to secure 175 targets against 4 attacker types with 87 resources or SSG instances where a defender has to secure 5 targets against 100 attacker types with 2 security resources.

It is our contention that the proposed decomposition algorithms could handle even larger GSG and SSG instances efficiently after some finer tuning is performed on their enhancing add-ons. A finer tuning of the stabilization parameter λ and some further tweaking of the cut generation interruption criteria could lead to substantial savings in resolution time of all of the decomposition approaches proposed. Further, we have seen that the reported integer gap of the decomposition procedures over the families of instances tested is in the vicinity of 10-20%. A thorough polyhedral study of the tight formulations (MIP-p-G$_{q,z}$) and (MIP-p-S$_{q,y}$) could yield strong facet-defining inequalities that could further decrease this integer gap. Another avenue of improvement involves exploring primal bound heuristics. In particular, a better upper bounding procedure could help in closing the optimality gap sooner and thus prune the branch and bound tree which could lead to performance improvements for the different approaches.

The performance of the network SSG formulation (FENCE$_{c,z,q,s,f,g}$) could be improved by implementing the polynomial-time separation procedure for the exponentially many odd set inequalities as in [Bucarey, 2017]. In addition, a stronger formulation could be obtained from (FENCE$_{c,z,q,s,f,g}$) by applying the strengthening techniques described in Chapter 4. Also, decomposition approaches like the ones described in Chapter 5 could be applied in this setting. Further, it would be interesting to consider extensions of the problem being modelled. For instance, a natural extension would be to allow for different pairings of nodes to account for different numbers of available resources, as opposed to the current setting where only a single resource is available at each pairing of nodes. In this new setting, different nodes in the network would have different degrees of importance.

Finally, much work could still be done on our real-life implementation of the Stackelberg-based software for patrol scheduling. To begin with, different parameter estimation methodologies could be considered to obtain improved modelizations of the real-world problem tackled. Currently, the payoff estimation for the players along the different targets is independent of the day of the week considered. Since the strategies for the players take the day of the week into account, it would make sense to consider payoffs which vary throughout the week.

We have pinpointed the evaluation of a deployed system as a sensitive phase. Computer-run simulations cannot provide the same level of evaluation as monitoring a deployed system. Further, we believe that conducting 'red' team exercises along the border after deployment of the software could yield important information about defender-attacker interactions. This could lead to a more precise parameter estimation and ultimately to a more sophisticated border patrol planner.

It would also be interesting to compare the optimal coverage strategies returned by our algorithm when i) the pairings between the different nodes are fixed; and ii) the pairings are part of the optimization. One would expect strategies with lower payoffs when pairings are fixed but it would be interesting to study their respective structure to understand how target coverage is distributed when edge distribution is part of the optimization. An intermediate option could be studied where the pairings are not fixed, but not all adjacent pairings are permitted in compliance with some meaningful restrictions relating to the nature of a specific real-life problem.

8.3 Closing remarks

Stackelberg games have sparked great interest among researchers and have a proven track record in tackling real life security problems. One of the main shortcomings experienced in efficiently solving these hard problems is scalability. Real life problems tend to be intricate and large in scale making naïve solution approaches inappropriate. Mathematical programming approaches have contributed greatly to the field, both from a modelling and from a resolution perspective.

In this thesis we have studied how to improve solution algorithms to tackle large scale games. It is fairly standard in the literature to exploit a problem's underlying structure in order to devise constraint and variable generation techniques to allow for an efficient handling of large instances. It is equally as important to study the formulations from a modelling perspective to understand how best to encode certain problem requirements as constraints in the formulations.

Even though the use of Stackelberg game formulations to provide solutions to real life problems is widespread, given the complexity of the Stackelberg paradigm, it is possible to encounter real life problems which are too large in size to be efficiently tackled by the exact methods analyzed in this thesis. In such cases, the key to finding strategic solutions lies in developing Stackelberg-inspired heuristics and metaheuristics.

The Stackelberg game theory literature has been undergoing a constant evolution since its inception in an attempt to address conceptual weaknesses in the assumptions, mathematical or otherwise, made in the Stackelberg framework. Behavioral game theory addresses the limitation of playing against a fully rational follower, repeated games address the limitation of playing a one-shot game which does not allow any of the players to adapt to the other's actions and learn over time. Researchers have tackled meaningful extensions to the base game where, for example, the follower may only partially observe the leader's mixed strategy and therefore the leader cannot fully anticipate the follower's response.

Throughout the different methodological approaches based on Stackelberg games, regardless of how complicated they might be, they all rely on an adequate estimation of the payoffs associated to the different actions that the players can take. In the security domain, in particular, we face two challenging problems. First, understanding what the possible actions might be for the different players in a given problem. Second, adequately estimating the payoffs associated to the different actions. Most of this work generally relies on discussions between the researcher and a domain expert. Even though a parameter estimation methodology, to a large extent, is highly problem dependent, taking into account the overall importance of a parameter estimation procedure on the validity of a Stackelberg framework, endowing the parameter estimation process with standard guidelines and mathematical rigor would enhance the applicability of the Stackelberg paradigm to tackle real life problems.

Bibliography

[CPL, 2017] (2017). IBM ILOG CPLEX Optimizer, V12.7.
url https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[Aduanas de Chile, 2016] Aduanas de Chile (2016). https://www.aduana.cl/importaciones-de-productos/aduana/2007-02-28/161116.html.

[Albers, 2015] Albers, C. (2015). Región de Arica y Parinacota coberturas sig. `http://www.rulamahue.cl/mapoteca/fichas/chile_geo/ficha15geo.html`.

[Alpern et al., 2011] Alpern, S., Morton, A., and Papadaki, K. (2011). Patrolling games. *Operations Research*, 59(5):1246–1257.

[Anandalingam and Friesz, 1992] Anandalingam, G. and Friesz, T. L. (1992). Hierarchical optimization: An introduction. *Annals of Operations Research*, 34(1):1–11.

[Barnhart et al., 1998] Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329.

[Ben-Ameur and Neto, 2007] Ben-Ameur, W. and Neto, J. (2007). Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17.

[Benders, 1962] Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numer. Math.*, 4(1):238–252.

[Bloem et al., 2007] Bloem, M., Alpcan, T., and Başar, T. (2007). A Stackelberg game for power control and channel allocation in cognitive radio networks. In *Proceedings of the 2Nd International Conference on Performance Evaluation Methodologies and Tools*, ValueTools '07, pages 4:1–4:9, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Bouza, 2006] Bouza, G. (2006). *Mathematical Programs with Equilibrium Constraints: Solution Techniques from Parametric Optimization.* PhD thesis, Universiteit Twente.

[Bracken and McGill, 1973] Bracken, J. and McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44.

[Bucarey, 2017] Bucarey, V. (2017). *Addressing Problem Size in Stackelberg Security Games*. PhD thesis, Universidad de Chile.

[Camerer, 2011] Camerer, C. (2011). *Behavioral Game Theory: Experiments in Strategic Interaction*. The Roundtable Series in Behavioral Economics. Princeton University Press.

[Cardinal et al., 2009] Cardinal, J., Labbé, M., Langerman, S., and Palop, B. (2009). Pricing geometric transportation networks. *Int. J. Comput. Geometry Appl.*, 19(6):507–520.

[Central Intelligence Agency, 2008] Central Intelligence Agency (2008). The world factbook. https://www.cia.gov/library/publications/the-world-factbook/geos/ci.html.

[Cheeseman et al., 1991] Cheeseman, P., Kanefsky, B., and Taylor, W. M. (1991). Where the really hard problems are. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'91, pages 331–337, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Comisión Económica para América Latina y el Caribe, 2000] Comisión Económica para América Latina y el Caribe (2000). Costo económico de los delitos, niveles de vigilancia y políticas de seguridad ciudadana en las comunas del Gran Santiago. http://www.cepal.org/es/publicaciones/7258-costo-economico-de-los-delitos-niveles-de-vigilancia-y-politicas-de-seguridad.

[Conitzer and Korzhyk, 2011] Conitzer, V. and Korzhyk, D. (2011). Commitment to correlated strategies. In Burgard, W. and Roth, D., editors, *AAAI*. AAAI Press.

[Conitzer and Sandholm, 2006] Conitzer, V. and Sandholm, T. (2006). Computing the optimal strategy to commit to. In ACM, editor, *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 82–90, New York, NY, USA. ACM.

[Correa et al., 2014] Correa, J. R., Harks, T., Kreuzen, V. J. C., and Matuschke, J. (2014). Fare evasion in transit networks. *CoRR*, abs/1405.2826.

[Council of the European Union, 2016] Council of the European Union (2016). http://www.consilium.europa.eu/en/press/press-releases/2016/09/14-european-border-coast-guard/.

[Cournot et al., 1897] Cournot, A., Bacon, N., and Fisher, I. (1897). *Researches Into the Mathematical Principles of the Theory of Wealth*. Economic classics. Macmillan.

[Dantzig and Eaves, 1973] Dantzig, G. B. and Eaves, C. B. (1973). Fourier-Motzkin Elimination and Its Dual. *J. Comb. Theory, Ser. A*, 14(3):288–297.

[Dantzig and Wolfe, 1960] Dantzig, G. B. and Wolfe, P. (1960). Decomposition principle for linear programs. *Oper. Res.*, 8(1):101–111.

[DeNegre, 2011] DeNegre, S. (2011). *Interdiction and Discrete Bilevel Linear Programming.* PhD thesis, Lehigh University.

[Department of Homeland Security of the United States, 2016] Department of Homeland Security of the United States (2016). https://www.dhs.gov/border-security.

[Edmonds, 1965] Edmonds, J. (1965). Maximum matching and a polyhedron with 0, l-vertices. *J. Res. Nat. Bur. Standards B*, 69:125–130.

[El Mercurio On-Line, 2016] El Mercurio On-Line (2016). Tráfico de inmigrantes en Chile: Bandas realizan hasta tres ingresos ilegales por semana. http://www.emol.com/noticias/Nacional/2016/05/13/802673/Trafico-de-Inmigrantes-Bandas-realizan-ingresos-ilegales-de-extranjeros-hasta-tres-veces-por-semana.html.

[Farkas, 1902] Farkas, J. (1902). Theorie der einfachen ungleichungen. *Journal fr die reine und angewandte Mathematik*, 124:1–27.

[Financial Times, 2015] Financial Times (2015). https://www.ft.com/topics/themes/EU_immigration.

[Fiscalía de Chile, Ministerio Público de Chile, 2014] Fiscalía de Chile, Ministerio Público de Chile (2014). Estadísticas. http://www.fiscaliadechile.cl/Fiscalia/estadisticas/index.do.

[Fiscalía de Chile, Ministerio Público de Chile, 2015] Fiscalía de Chile, Ministerio Público de Chile (2015). Informe 2015: Observatorio del narcotráfico en Chile. http://www.fiscaliadechile.cl/observatoriodrogaschile/documentos/informe_final_2015.pdf.

[Fischetti et al., 2016a] Fischetti, M., Ljubic, I., Monaci, M., and Sinnl, M. (2016a). Interdiction games and monotonicity. Technical report, DEI, University of Padova.

[Fischetti et al., 2016b] Fischetti, M., Ljubic, I., and Sinnl, M. (2016b). Redesigning Benders decomposition for large-scale facility location. Technical report, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal.

[Flegel and Kanzow, 2005a] Flegel, M. L. and Kanzow, C. (2005a). On m-stationary points for mathematical programs with equilibrium constraints. *Journal of Mathematical Analysis and Applications*, 310(1):286 – 302.

[Flegel and Kanzow, 2005b] Flegel, M. L. and Kanzow, C. (2005b). On the Guignard constraint qualification for mathematical programs with equilibrium constraints. *Optimization*, 54(6):517–534.

[Fortz et al., 2013] Fortz, B., Labbé, M., and Violin, A. (2013). Dantzig-wolfe reformulation for the network pricing problem with connected toll arcs. *Electronic Notes in Discrete Mathematics*, 41:117–124.

[Gent and Walsh, 1996] Gent, I. P. and Walsh, T. (1996). The TSP phase transition. *Artificial Intelligence*, 88(1):349 – 358.

[Ghare et al., 1971] Ghare, P. M., Montgomery, D. C., and Turner, W. C. (1971). Optimal interdiction policy for a flow network. *Naval Research Logistics Quarterly*, 18(1):37–45.

[Harsanyi and Selten, 1972] Harsanyi, J. C. and Selten, R. (1972). A Generalized Nash Solution for Two-Person Bargaining Games with Incomplete Information. *Management Science*, 18(5).

[Haskell et al., 2014] Haskell, W. B., Kar, D., Fang, F., Tambe, M., Cheung, S., and Denicola, E. (2014). Robust protection of fisheries with compass. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2978–2983.

[Hochbaum et al., 2014] Hochbaum, D. S., Lyu, C., and Ordóñez, F. (2014). Security routing games with multivehicle chinese postman problem. *Networks*, 64(3):181–191.

[Howard, 1960] Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* MIT Press, Cambridge, MA.

[Israeli, 1999] Israeli, E. (1999). *System Interdiction and Defense.* PhD thesis, Naval Postgraduate School.

[Jain et al., 2010a] Jain, M., Kardes, E., Kiekintveld, C., Ordóñez, F., and Tambe, M. (2010a). Security games with arbitrary schedules: A branch and price approach. In Fox, M. and Poole, D., editors, *AAAI.* AAAI Press.

[Jain et al., 2011a] Jain, M., Kiekintveld, C., and Tambe, M. (2011a). Quality-bounded solutions for finite bayesian Stackelberg games: Scaling up. In *The 10th International*

Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '11, pages 997–1004, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Jain et al., 2011b] Jain, M., Korzhyk, D., Vaněk, O., Conitzer, V., Pěchouček, M., and Tambe, M. (2011b). A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, pages 327–334, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Jain et al., 2012] Jain, M., Leyton-Brown, K., and Tambe, M. (2012). The deployment-to-saturation ratio in security games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1362–1370. AAAI Press.

[Jain et al., 2010b] Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., and Ordóñez, F. (2010b). Software assistants for randomized patrol planning for the LAX Airport Police and the Federal Air Marshal Service. *Interfaces*, 40(4):267–290.

[Jeroslow, 1985] Jeroslow, R. G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical Programming*, 32(2):146–164.

[Kanzow and Schwartz, 2010] Kanzow, C. and Schwartz, A. (2010). Mathematical programs with equilibrium constraints: Enhanced Fritz John-conditions, new constraint qualifications and improved exact penalty results. *SIAM Journal on Optimization*, 20:2730–2753.

[Kar et al., 2016] Kar, D., Fang, F., Fave, F. M. D., Sintov, N., Tambe, M., and Lyet, A. (2016). Comparing human behavior models in repeated Stackelberg security games: An extended study. *Artificial Intelligence*, 240:65 – 103.

[Karp, 1972] Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.

[Kelley, 1960] Kelley, J. E. (1960). The cutting plane method for solving convex programs. *Journal of the SIAM*, 8:703–712.

[Kiekintveld et al., 2009] Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., and Tambe, M. (2009). Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents* *and Multiagent Systems - Volume 1*, AAMAS '09, pages 689–696, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Kolstad, 1985] Kolstad, C. (1985). A review of the literature on bi-level mathematical programming. Technical report, Los Alamos Nat. Lab.

[Korzhyk et al., 2010] Korzhyk, D., Conitzer, V., and Parr, R. (2010). Complexity of computing optimal stackelberg strategies in security resource allocation games. In *In Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 805–810.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

[Labbé et al., 1998] Labbé, M., Marcotte, P., and Savard, G. (1998). A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12-part-1):1608–1622.

[Labbé and Violin, 2016] Labbé, M. and Violin, A. (2016). Bilevel programming and price setting problems. *Annals of Operations Research*, 240(1):141–169.

[Laffont and Martimort, 2009] Laffont, J. and Martimort, D. (2009). *The Theory of Incentives: The Principal-Agent Model.* Princeton University Press.

[Laffont and Tirole, 1993] Laffont, J. and Tirole, J. (1993). *A Theory of Incentives in Procurement and Regulation.* MIT Press.

[Leitman, 1978] Leitman, G. (1978). On generalized Stackelberg strategies. *J. Optim. Theory Appl.*, 26(4):637–643.

[Marecki et al., 2012] Marecki, J., Tesauro, G., and Segal, R. (2012). Playing repeated Stackelberg games with unknown opponents. In *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS-12).*

[McMasters and Mustin, 1970] McMasters, A. W. and Mustin, T. M. (1970). Optimal interdiction of a supply network. *Naval Research Logistics Quarterly*, 17(3):261–268.

[McNaughton, 1959] McNaughton, R. (1959). Scheduling with deadlines and loss functions. *Management Science*, 6(1):1–12.

[Ministerio del Trabajo y Previsión Social, 2016] Ministerio del Trabajo y Previsión Social (2016). http://www.leychile.cl/Navegar?idLey=20763.

[Moerenhout, 2012] Moerenhout, J. (2012). Formulation et résolution du jeu de securité. Technical report, Département d'Informatique, Université Libre de Bruxelles.

[Möller, 1948] Möller, H. (1948). Heinrich Freiherr von Stackelberg und sein beitrag fur die Wirtschaftswissenchaft. *Journal of Institutional and Theoretical Economics*, 105(3):395–428.

[Morais et al., 2016] Morais, V., Salles Da Cunha, A., and Mahey, P. (2016). A Branch-and-cut-and-price algorithm for the Stackelberg Minimum Spanning Tree Game. *Electronic Notes in Discrete Mathematics*, 52:309 – 316.

[Nash, 1950] Nash, J. F. (1950). Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36:48–49.

[Padberg and Rao, 1982] Padberg, M. W. and Rao, M. R. (1982). Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7(1):67–80.

[Papadaki et al., 2016] Papadaki, K., Alpern, S., Lidbetter, T., and Morton, A. (2016). Patrolling a border. *Operations Research*, 64(6):1256–1269.

[Paruchuri et al., 2008] Paruchuri, P., Pearce, J. P., Marecki, J., Tambe, M., Ordóñez, F., and Kraus, S. (2008). Playing games for security: An efficient exact algorithm for solving bayesian Stackelberg games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '08, pages 895–902, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Paruchuri et al., 2007] Paruchuri, P., Pearce, J. P., Tambe, M., Ordóñez, F., and Kraus, S. (2007). An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, pages 181:1–181:8, New York, NY, USA. ACM.

[Pita et al., 2008] Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., and Kraus, S. (2008). Deployed armor protection: The application of a game theoretic model for security at the Los Angeles International Airport. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track*, AAMAS '08, pages 125–132, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Pita et al., 2009] Pita, J., Jain, M., Ordóñez, F., Tambe, M., Kraus, S., and Magori-Cohen, R. (2009). Effective solutions for real-world Stackelberg games: When agents must deal with human uncertainties. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09, pages 369–376, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Pita et al., 2011] Pita, J., Tambe, M., Kiekintveld, C., Cullen, S., and Steigerwald, E. (2011). Guards: game theoretic security allocation on a national scale. In Sonenberg, L., Stone, P., Tumer, K., and Yolum, P., editors, *AAMAS*, pages 37–44. IFAAMAS.

[QGIS, 2009] QGIS (2009). *QGIS Geographic Information System*. Open Source Geospatial Foundation.

[Roh et al., 2011] Roh, H., Park, H., Jung, C., and Lee, W. (2011). A Stackelberg game for spectrum investment and pricing in cooperative cognitive radio networks. In *Proceedings of The ACM CoNEXT Student Workshop*, CoNEXT '11 Student, pages 8:1–8:2, New York, NY, USA. ACM.

[Salanié, 2005] Salanié, B. (2005). *The Economics of Contracts: A Primer*. MIT Press.

[Savard, 1989] Savard, G. (1989). *Contirbutions à la programmation mathématique a deux niveaux*. PhD thesis, École Polytechnique, Université de Montreal.

[Schrijver, 2003] Schrijver, A. (2003). *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

[Sherali and Adams, 1994] Sherali, H. D. and Adams, W. P. (1994). A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, 52(1):83 – 106.

[Shieh et al., 2012] Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., and Meyer, G. (2012). Protect: A deployed game theoretic system to protect the ports of the united states. In Richland, S., editor, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, volume 1 of *AAMAS '12*, pages 13–20, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[Sinha et al., 2015] Sinha, A., Nguyen, T. H., Kar, D., Brown, M., Tambe, M., and Jiang, A. X. (2015). From physical security to cybersecurity. *J. Cybersecurity*, 1(1):19–35.

[Slaney and Walsh, 2002] Slaney, J. and Walsh, T. (2002). Phase transition behavior: From decision to optimization. In *Proceedings of the 5th International Symposium on the Theory and Applications of Satisfiability Testing*, SAT.

[Suárez-Vega et al., 2011] Suárez-Vega, R., Santos-Peñate, D. R., Dorta-González, P., and Rodríguez-Díaz, M. (2011). A multi-criteria GIS based procedure to solve a network competitive location problem. *Applied Geography*, 31(1):282–291.

[Tambe, 2011] Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned.* Cambridge University Press.

[Vicente and Calamai, 1994] Vicente, L. N. and Calamai, P. H. (1994). Bilevel and multilevel programming: A bibliography review. *Journal of Global Optimization*, 5(3):291–306.

[Violin, 2014] Violin, A. (2014). *Mathematical Programming approaches to pricing problems.* PhD thesis, Université Libre de Bruxelles, Università degli Studi di Trieste.

[von Stengel and Zamir, 2004] von Stengel, B. and Zamir, S. (2004). Leadership with commitment to mixed strategies. Technical report, London School of Economics-Centre for Discrete and Applicable Mathematics (LSE-CDAM).

[Wolsey, 1998] Wolsey, L. A. (1998). *Integer programming.* Wiley-Interscience, New York, NY, USA.

[Wood, 1993] Wood, R. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling*, 17(2):1 – 18.

[Yang et al., 2014] Yang, R., Ford, B., Tambe, M., and Lemieux, A. (2014). Adaptive resource allocation for wildlife protection against illegal poachers. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

[Yang et al., 2013] Yang, R., Jiang, A. X., Tambe, M., and Ordóñez, F. (2013). Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In Rossi, F., editor, *IJCAI.* IJCAI/AAAI.

[Yin et al., 2012] Yin, Z., Jiang, A. X., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., and Sullivan, J. P. (2012). Trusts: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59–72.

[Yin et al., 2010] Yin, Z., Korzhyk, D., Kiekintveld, C., Conitzer, V., and Tambe, M. (2010). Stackelberg vs. Nash in security games: interchangeability, equivalence, and uniqueness. In van der Hoek, W., Kaminka, G. A., Lesprance, Y., Luck, M., and Sen, S., editors, *AAMAS*, pages 1139–1146. IFAAMAS.

[Yin and Tambe, 2012] Yin, Z. and Tambe, M. (2012). A unified method for handling discrete and continuous uncertainty in bayesian Stackelberg games. In Conitzer, W. and van der Hoek, editors, *AAMAS.*

[Zhang and Zhang, 2009] Zhang, J. and Zhang, Q. (2009). Stackelberg game for utility-based cooperative cognitiveradio networks. In *Proceedings of the Tenth ACM Interna-*

tional Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '09, pages 23–32, New York, NY, USA. ACM.

Appendix A

Additional material

A.1 (QUAD$_{x,q,s}$) is equivalent to (MIP-p-G$_{q,z}$)

We show that the quadratic single level formulation (QUAD$_{x,q,s}$) and the MILP formulation (MIP-p-G$_{q,z}$) are equivalent, in the sense that a 1–1 correspondence relation can be established between the sets of feasible points such that the objective value is preserved. The formulation (QUAD$_{x,q,s}$) is given by:

$$
\begin{array}{llll}
(\text{QUAD}_{x,q,s}) & \text{Max}_{x,q,s} & \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} x_i q^k_j & & \text{(A.1.1)}\\
& \text{s.t.} & \sum_{i\in I} x_i = 1, & & \text{(A.1.2)}\\
& & x_i \geq 0 & \forall i\in I, & \text{(A.1.3)}\\
& & \sum_{j\in J} q^k_j = 1 & \forall k\in K, & \text{(A.1.4)}\\
& & q^k_j \in \{0,1\} & \forall j\in J, \forall k\in K, & \text{(A.1.5)}\\
& & 0 \leq s^k - \sum_{i\in I} C^k_{ij} x_i & & \\
& & \quad \leq (1-q^k_j)M & \forall j\in J, \forall k\in K, & \text{(A.1.6)}\\
& & s^k \in \mathbb{R} & \forall k\in K. & \text{(A.1.7)}
\end{array}
$$

The formulation (MIP-p-G$_{q,z}$) is given by:

$$
\begin{array}{llll}
(\text{MIP}p\text{-G}_{q,z}) & \text{Max}_{q,z} & \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\pi^k R^k_{ij} z^k_{ij} & & \text{(A.1.8)}\\
& \text{s.t.} & \sum_{j\in J} q^k_j = 1 & \forall k\in K, & \text{(A.1.9)}\\
& & q^k_j \in \{0,1\} & \forall j\in J, \forall k\in K, & \text{(A.1.10)}\\
& & \sum_{j\in J} z^k_{ij} = \sum_{j\in J} z^1_{ij} & \forall i\in I, \forall k\in K, & \text{(A.1.11)}
\end{array}
$$

$$\sum_{i\in I} z_{ij}^k = q_j^k \qquad \forall j \in J, \forall k \in K, \tag{A.1.12}$$

$$z \geq 0 \tag{A.1.13}$$

$$\sum_{i\in I}(C_{ij}^k - C_{i\ell}^k) z_{ij}^k \geq 0 \qquad \forall \ell, j \in J, \forall k \in K. \tag{A.1.14}$$

Before proceeding with the proof, applying Fourier-Motzkin elimination on Constraint (A.1.6) to project out the s^k variables, produces:

$$\sum_{i\in I}(C_{ij}^k - C_{i\ell}^k)x_i \geq (1-q_j^k)M \qquad \forall \ell, j \in J, \forall k \in K.$$

Multiplying throughout the previous family of constraints by the corresponding $q_j^k \in \{0,1\}$ yields the following equivalent family of constraints:

$$\sum_{i\in I}(C_{ij}^k - C_{i\ell}^k)x_i q_j^k \geq 0 \qquad \forall \ell, j \in J, \forall k \in K. \tag{A.1.15}$$

Proposition A.1.1. *(QUAD$_{x,q}$) defined as maximizing (A.1.1) subject to (A.1.2)-(A.1.5), (A.1.15) is equivalent to (MIP-p-G$_{q,z}$) defined as maximizing (A.1.8) subject to (A.1.9)-(A.1.14).*

Proof. Let (x, q) be a feasible solution for (QUAD$_{x,q}$). One wants to show that if one defines $z_{ij}^k = x_i q_j^k$ for all $i \in I, j \in J, k \in K$, then (q, z) is a feasible solution for (MIP-p-G$_{q,z}$) with the same objective value. One can easily check that Constraints (A.1.9)-(A.1.14) are readily satisfied, and (q, z) provides the same objective value.

Conversely, let (q, z) be a feasible solution for (MIP-p-G$_{q,z}$). One wants to show that if one defines $x_i = \sum_{j\in J} z_{ij}^k$ for all $i \in I$ and $k \in K$, then (x, q) is feasible for (QUAD$_{x,q}$) and provides the same objective value. Constraint (A.1.2) holds because (A.1.9) and (A.1.12) hold:

$$\sum_{i\in I} x_i = \sum_{i\in I}\left(\sum_{j\in J} z_{ij}^k\right) = \sum_{j\in J}\left(\sum_{i\in I} z_{ij}^k\right) = \sum_{j\in J} q_j^k = 1$$

Constraint (A.1.3) holds because (A.1.13) holds. Constraints (A.1.4) and (A.1.5) hold by definition. Constraint (A.1.15) holds based on the following argument. Note that for each $k \in K$, there exists a single $j \in J$, say j_k, such that $q_{j_k}^k = 1$. Then, from (A.1.12), $q_{j_k}^k = \sum_{i\in I} z_{ij_k}^k = 1$. Also, $\sum_{i\in I}\sum_{j\in J} z_{ij}^k = 1$ for all $k \in K$ from summing over $i \in I$ in (A.1.12) and using (A.1.9). Therefore,

$$z_{ij}^k = 0 \qquad \forall k \in K, \forall i \in I, \forall j \in J : j \neq j_k. \tag{A.1.16}$$

In particular, it follows that

$$x_i = \sum_{j\in J} z_{ij}^1 = z_{ij_1}^1 = z_{ij_k}^k, \qquad \forall i \in I, \forall k \in K, \tag{A.1.17}$$

where the first equality follows from our definition of x, the second equality holds because of (A.1.16) and the third equality is a consequence of (A.1.11). Finally,

$$x_i q_j^k = z_{ij_k}^k q_j^k = z_{ij}^k \qquad \forall i \in I, \forall j \in J, \forall k \in K, \tag{A.1.18}$$

where the first equality holds because of (A.1.17) and the second equality is a consequence of (A.1.16). It thus follows that (A.1.15) holds and therefore, (x, q) is feasible for $(\text{QUAD}_{x,q})$ and attains the same objective value. ■

A.2 Tight M values

A.2.1 General Stackelberg games

To deduce the smallest possible big-M values in the multiple follower type MILP formulations, for the sake of notation, we initially restrict ourselves to a single follower type scenario. The extension to multiple follower types is direct. Consider the following sets:

$$\begin{aligned}
\text{F}(M^1) \;&:=\; \Bigg\{(x,q,f) : \sum_{i\in I} x_i = 1,\ x \geq 0,\ \sum_{j\in J} q_j = 1,\ q \in \{0,1\}^{|J|},\\
&\qquad f - \sum_{i\in I} R_{ij} x_i \leq (1-q_j) M_j^1 \ \forall j \in J\Bigg\},\\
\text{G}(M^2) \;&:=\; \Bigg\{(x,q,s) : \sum_{i\in I} x_i = 1,\ x \geq 0,\ \sum_{j\in J} q_j = 1,\ q \in \{0,1\}^{|J|},\\
&\qquad 0 \leq s - \sum_{i\in I} C_{ij} x_i \leq (1-q_j) M_j^2 \ \forall j \in J\Bigg\},\\
\text{H}(N) \;&:=\; \Bigg\{(x,q) : \sum_{i\in I} x_i = 1,\ x \geq 0,\ \sum_{j\in J} q_j = 1,\ q \in \{0,1\}^{|J|},\\
&\qquad \sum_{i\in I} (C_{ij} - C_{i\ell}) x_i \leq (1-q_\ell) N_{\ell j} \ \forall \ell, j \in J\Bigg\},
\end{aligned}$$

and let $(\overline{\text{F}(M^1)})$, $(\overline{\text{G}(M^2)})$ and $(\overline{\text{H}(N)})$ be the corresponding linear relaxations of these sets obtained by replacing the binary requirements on the q variables by non-negativity requirements. Further, consider the following non-negative constants M^{1^*}, M^{2^*} and N^*:

$$(M^{1^*})_{j\in J} \text{ where } M_j^{1^*} = \max_{i\in I} \left\{ \max_{\ell\in J} R_{i\ell} - R_{ij} \right\} \forall j \in J, \tag{A.2.1}$$

$$(M^{2^*})_{j\in J} \text{ where } M_j^{2^*} = \max_{i\in I} \left\{ \max_{\ell\in J} C_{i\ell} - C_{ij} \right\} \forall j \in J, \tag{A.2.2}$$

$$(N^*)_{\ell,j\in J} \text{ where } N_{\ell j}^* = \max_{i\in I} \{C_{ij} - C_{i\ell}\} \ \forall \ell, j \in J. \tag{A.2.3}$$

Lemma A.2.1. *The following set relations hold:*

$$F(M^1) = F(M^{1^*}),\ \forall M^1 \geq M^{1^*}, \tag{A.2.4}$$

$$G(M^2) = G(M^{2^*}),\ \forall M^2 \geq M^{2^*}, \tag{A.2.5}$$

$$H(N) = H(N^*),\ \forall N \geq N^*. \tag{A.2.6}$$

Proof. To show (A.2.4) holds, $\mathrm{F}(M^{1^*}) \subseteq \mathrm{F}(M^1)$ for $M^1 \geq M^{1^*}$ is trivially satisfied. To see the other inclusion, consider $(x, q, f) \in \mathrm{F}(M^1)$. From $\sum_{j \in J} q_j = 1$ and $q \in \{0, 1\}$, it follows that $\exists j^* \in J$ such that $q_{j^*} = 1$. Thus, $f \leq \max_{\ell \in J} \left\{\sum_{i \in I} R_{i\ell} x_i\right\}$. It follows that

$$f - \sum_{i \in I} R_{ij} x_i \leq \max_{\ell \in J} \left\{ \sum_{i \in I} R_{i\ell} x_i \right\} - \sum_{i \in I} R_{ij} x_i\ \forall j \in J. \tag{A.2.7}$$

The RHS of Equation (A.2.7) is a convex function, since the last term is linear and the first term is a maximum of convex functions, which is, again, a convex function. A convex function attains its maximum at an extreme point of its domain. Hence,

$$f - \sum_{i \in I} R_{ij} x_i \leq \max_{i \in I} \left\{ \max_{\ell \in J} R_{i\ell} - R_{ij} \right\} = M1^*_j\ \forall j \in J.$$

So, $(x, q, f) \in \mathrm{F}(M^{1^*})$ and (A.2.4) is satisfied. An analogous argument shows that (A.2.5) similarly holds.

Finally, to show that (A.2.6) holds, $\mathrm{H}(N^*) \subseteq \mathrm{H}(N)$ for $N \geq N^*$ is trivially satisfied. To show the remaining inclusion, consider $(x, q) \in \mathrm{H}(N)$. Since $\sum_{i \in I} x_i = 1$ and $x \geq 0$, it follows that:

$$\sum_{i \in I} (C_{ij} - C_{i\ell}) x_i \leq \max_{i \in I} \{C_{ij} - C_{i\ell}\} = N_{\ell j}\ \forall \ell, j \in J. \tag{A.2.8}$$

It follows that $(x, q) \in \mathrm{H}(N^*)$ and thus (A.2.6) holds. ■

The following lemma shows that considering values of M^1, M^2 or N smaller than M^{1^*}, M^{2^*} or N^*, leads to cutting off solutions.

Lemma A.2.2. *The following set relations hold:*

$$F(M^1) \subset F(M^{1^*}),\ \forall M^1 : \exists j \in J : M^1_j < M^{1^*}_j, \tag{A.2.9}$$

$$G(M^2) \subset G(M^{2^*}),\ \forall M^2 : \exists j \in J : M^2_j < M^{2^*}_j, \tag{A.2.10}$$

$$H(N) \subset H(N^*),\ \forall N : \exists, \ell, j \in J : N_{\ell j} < N^*_{\ell j}. \tag{A.2.11}$$

Proof. We first show that (A.2.9) holds. Let $i^* = \arg\max_{i \in I} \{\max_{h \in J} R_{ih} - R_{ij}\}$ for all $j \in J : q_j \neq 1$ and let $j^* = \arg\max_{j \in J} R_{i^*j}$. Consider (x, q, f) such that $x_{i^*} = 1$, $q_{j^*} = 1$ and $f = R_{i^*j^*}$. If one plugs in this point into constraint $f - \sum_{i \in I} R_{ij} x_i \leq (1 - q_j) M^{1^*}$ for all $j \neq j^*$, the constraint is satisfied with equality, so $(x, q, f) \in \mathrm{F}(M^{1^*})$. Therefore, for M^1

for which $\exists j \in J : M^1_j < M^{1*}_j$, $(x, q, f) \notin \mathrm{F}(M^1)$ and the result follows.

An equivalent argument shows that (A.2.10) holds. Further, to show (A.2.11), consider (x, q) such that $x_{i^*} = 1$ and $q_{j^*} = 1$. Therefore, for any $j \in J : j \neq j^*$, $\sum_{i \in I}(C_{ij} - C_{i\ell}) \leq (1 - q_j)N^*$ is satisfied with equality so $(x, q) \in \mathrm{H}(N^*)$. Then, for any N for which $\exists, \ell, j \in J : N_{\ell j} < N^*_{\ell j}$, $(x, q) \notin \mathrm{H}(N)$ and the result follows. ■

Further, $(\overline{\mathrm{F}(M^{1^*})})$, $(\overline{\mathrm{G}(M^{2^*})})$ and $(\overline{\mathrm{H}(N^*)})$ are the smallest polyhedral regions to contain $\mathrm{F}(M^1)$, $\mathrm{G}(M^2)$ and $\mathrm{H}(N)$, respectively.

Lemma A.2.3. *The following set relations hold:*

$$\begin{aligned}
\overline{F(M^{1^*})} &\subseteq \overline{F(M^1)}\ \forall M^1 \geq M^{1^*},\\
\overline{G(M^{2^*})} &\subseteq \overline{G(M^2)}\ \forall M^2 \geq M^{2^*},\\
\overline{H(N^*)} &\subseteq \overline{H(N)}\ \forall N \geq N^*.
\end{aligned}$$

Proof. The result follows directly from Lemmas A.2.1 and A.2.2 and applying linear relaxations to the sets. ■

Lemma A.2.4. *The non-negative constants M^{1^*}, M^{2^*} and N^* shown in (A.2.1)-(A.2.3) are best possible for $F(M^1)$, $G(M^2)$ and $H(N)$, respectively.*

Proof. The result follows directly from Lemma A.2.3. ■

The tight M values for multiple follower type settings are given by the following expressions:

$$\begin{aligned}
{M^1}^{k^*}_j &= \max_{i \in I}\left\{\max_{\ell \in J} R^k_{i\ell} - R^k_{ij}\right\} \forall j \in J, \forall k \in K,\\
{M^2}^{k^*}_j &= \max_{i \in I}\left\{\max_{\ell \in J} C^k_{i\ell} - C^k_{ij}\right\} \forall j \in J, \forall k \in K,\\
N^{k^*}_{\ell j} &= \max_{i \in I}\left\{C^k_{ij} - C^k_{i\ell}\right\} \forall \ell, j \in, \forall k \in K..
\end{aligned}$$

A.2.2 Stackelberg security games

As in the general setting, we initially restrict our attention to the single attacker type scenario. The extension to multiple attacker types is immediate. Consider the following sets:

$$\begin{aligned}
\mathrm{F_S}(M^1) &:= \left\{(x, c) : \sum_{i \in I : j \in i} x_i = c_j\ \forall j \in J\right\} \cap \mathrm{F}(M^1),\\
\mathrm{G_S}(M^2) &:= \left\{(x, c) : \sum_{i \in I : j \in i} x_i = c_j\ \forall j \in J\right\} \cap \mathrm{G}(M^2),
\end{aligned}$$

$$\mathrm{H_S}(N) \;:=\; \left\{(x,c) : \sum_{i\in I: j\in i} x_i = c_j \ \forall j \in J\right\} \cap \mathrm{H}(N).$$

Applying the projection result and rewriting constraints leads to:

$$\begin{aligned}
\mathrm{F_{Sec}}(M^1) \;&=\; \mathit{Proj}_{c,q,f}(\mathrm{F_S}(M^1)) := \{(c,q,f) : \sum_{j\in J} c_j \le m, c \in [0,1]^{|J|},\\
&\sum_{j\in J} q_j = 1, q \in \{0,1\}^{|J|},\\
&f - D(j|c)c_j - D(j|u)(1-c_j) \le (1-q_j)M^1_j \ \forall j \in J\}, \qquad \text{(A.2.12)}\\
\mathrm{G_{Sec}}(M^2) \;&=\; \mathit{Proj}_{c,q,s}(\mathrm{G_S}(M^2)) := \{(c,q,s) : \sum_{j\in J} c_j \le m, c \in [0,1]^{|J|},\\
&\sum_{j\in J} q_j = 1, q \in \{0,1\}^{|J|},\\
&0 \le s - A(j|c)c_j - A(j|u)(1-c_j) \le (1-q_j)M^2_j \ \forall j \in J\}, \qquad \text{(A.2.13)}\\
\mathrm{H_{Sec}}(N) \;&=\; \mathit{Proj}_{c,q}(\mathrm{H_S}(N)) := \{(c,q) : \sum_{j\in J} c_j \le m, c \in [0,1]^{|J|}, \sum_{j\in J} q_j = 1,\\
&q \in \{0,1\}^{|J|}, A(j|c)c_j + A(j|u)(1-c_j) -\\
&(A(\ell|c)c_\ell + A(\ell|u)(1-c_\ell)) \le (1-q_\ell)N_{\ell j} \ \forall \ell, j \in J\} \qquad \text{(A.2.14)}
\end{aligned}$$

and let $(\overline{\mathrm{F_{Sec}}(M^1)})$,$(\overline{\mathrm{G_{Sec}}(M^2)})$ and $(\overline{\mathrm{H_{Sec}}(N)})$ be the corresponding linear relaxations obtained by replacing the binary requirements on the q variables by a non-negativity requirement. As in the general case, consider the following non-negative constants M^{1^*}, M^{2^*} and N^*:

$$(M^{1^*})_{j\in J} \text{ where } M^{1^*}_j = \max_{\ell\in J:\ell\neq j}\{D(\ell|c), D(\ell|u)\} - \min\{D(j|c), D(j|u)\} \ \forall j \in J, \quad \text{(A.2.15)}$$

$$(M^{2^*})_{j\in J} \text{ where } M^{2^*}_j = \max_{\ell\in J:\ell\neq j}\{A(\ell|c), A(\ell|u)\} - \min\{A(j|c), A(j|u)\} \ \forall j \in J, \quad \text{(A.2.16)}$$

$$(N^*)_{\ell,j\in J} \text{ where } N^*_{\ell j} = \max\{A(j|c), A(j|u)\} - \min\{A(\ell|c), A(\ell|u)\} \ \forall \ell, j \in J. \quad \text{(A.2.17)}$$

The following lemmas, Lemmas A.2.5, A.2.6 and A.2.7 can be easily shown to hold by adapting the proofs of Lemmas A.2.1, A.2.2 and A.2.3 in the general setting, exploiting the relationship between the payoff structures in general and security Stackelberg games.

Lemma A.2.5. *The following set relations hold:*

$$\begin{aligned}
F_{Sec}(M^1) \;&=\; F_{Sec}(M^{1^*}),\ \forall M^1 \ge M^{1^*},\\
G_{Sec}(M^2) \;&=\; G_{Sec}(M^{2^*}),\ \forall M^2 \ge M^{2^*},\\
H_{Sec}(N) \;&=\; H_{Sec}(N^*),\ \forall N \ge N^*.
\end{aligned}$$

Lemma A.2.6. *The following set relations hold:*

$$\begin{aligned}
F_{Sec}(M^1) \;&\subset\; F_{Sec}(M^{1^*}),\ \forall M^1 : \exists j \in J : M^1_j < M^{1^*}_j,\\
G_{Sec}(M^2) \;&\subset\; G_{Sec}(M^{2^*}),\ \forall M^2 : \exists j \in J : M^2_j < M^{2^*}_j,\\
H_{Sec}(N) \;&\subset\; H_{Sec}(N^*),\ \forall N : \exists, \ell, j \in J : N_{\ell j} < N^*_{\ell j}.
\end{aligned}$$

Lemma A.2.7. *The following set relations hold:*

$$\begin{aligned}
\overline{F_{Sec}(M^{1^*})} &\subseteq \overline{F_{Sec}(M^1)}\ \forall M^1 \geq M^{1^*},\\
\overline{G_{Sec}(M^{2^*})} &\subseteq \overline{G_{Sec}(M^2)}\ \forall M^2 \geq M^{2^*},\\
\overline{H_{Sec}(N^*)} &\subseteq \overline{H_{Sec}(N)}\ \forall N \geq N^*.
\end{aligned}$$

From Lemmas A.2.5-A.2.7, the following result follows:

Lemma A.2.8. *The non-negative constants M^{1^*}, M^{2^*} and N^* shown in (A.2.15)-(A.2.17) are best possible for $F_{Sec}(M^1)$, $G_{Sec}(M^2)$, and $H_{Sec}(N)$, respectively.*

The tight M values for multiple attacker types settings are given by the following expressions:

$$\begin{aligned}
{M^1}_j^{k^*} &= \max_{\ell \in J: \ell \neq j}\{D^k(\ell|c), D^k(\ell|u)\} - \min\{D^k(j|c), D^k(j|u)\}\ \forall j \in J, \forall k \in K,\\
{M^2}_j^{k^*} &= \max_{\ell \in J: \ell \neq j}\{A^k(\ell|c), A^k(\ell|u)\} - \min\{A^k(j|c), A^k(j|u)\}\ \forall j \in J, \forall k \in K,\\
N_{\ell j}^{k^*} &= \max\{A^k(j|c), A^k(j|u)\} - \min\{A^k(\ell|c), A^k(\ell|u)\}\ \forall \ell, j \in J, \forall k \in K.
\end{aligned}$$

A.3 (MIP-p-G) is convex hull defining for $p = 1$

The following is an equivalent rewriting of (MIP-1-G$_{q,z}$), the single follower type restriction of (MIP-p-G$_{q,z}$), expressed only in terms of the z variables:

$$\begin{aligned}
(\text{MIP-1-G}_z) \quad \text{Max}_z \quad & \sum_{i \in I}\sum_{j \in J} R_{ij} z_{ij} & & \\
\text{s.t.} \quad & \sum_{i \in I}\sum_{j \in J} z_{ij} = 1, & & \text{(A.3.1)}\\
& \sum_{i \in I}(C_{ij} - C_{i\ell}) z_{ij} \geq 0 & \forall \ell, j \in J, \quad & \text{(A.3.2)}\\
& z_{ij} \geq 0 & \forall i \in I, \forall j \in J, \quad & \text{(A.3.3)}\\
& \sum_{i \in I} z_{ij} \in \{0, 1\} & \forall j \in J. \quad & \text{(A.3.4)}
\end{aligned}$$

The result we present next is due to [Conitzer and Sandholm, 2006] and the simple proof that follows, appears in [Moerenhout, 2012].

Theorem A.3.1. *Every vertex z of the polyhedron (P) defined by Constraints (A.3.1)-(A.3.3) verifies Constraint (A.3.4).*

Proof. Consider a point $z \in P$ and suppose that there exists $\hat{j} \in J$ such that $q_{\hat{j}} = \sum_{i \in I} z_{i\hat{j}} \notin \{0, 1\}$, *i.e.*, q is a vector with a non zero fractional component.

Let S be the maximal subset of J that contains all indices $s \in J : q_s \neq 0$ and let us define for each $s \in S$, a point $z^{(s)}$ as follows:

$$z_{ij}^{(s)} = \begin{cases} \frac{z_{ij}}{q_j} & \text{if } j = s, \\ 0 & \text{otherwise.} \end{cases}$$

Note that since (A.3.1) holds and q has a non zero fractional component, it must be the case that $|S| \geq 2$. Let us now show that for the indices of $s \in S$, for which $q_s \neq 0$, the corresponding $z^{(s)}$ is feasible in (P).

To show that for each $s \in S$, $z^{(s)}$ satisfies (A.3.1), note that for each $s \in S$

$$\sum_{i\in I}\sum_{j\in J} z_{ij}^{(s)} = \sum_{i\in I} \frac{z_{is}}{q_s} = \frac{1}{q_s}\sum_{i\in I} z_{is} = \frac{1}{q_s} q_s = 1.$$

To show that (A.3.2) is satisfied, note that if $s = \ell$, both sides of the constraint are 0 and so the constraint trivially holds. If $s \neq \ell$,

$$\sum_{i\in I}(C_{ij} - C_{i\ell})z_{ij}^{(s)} = \sum_{i\in I}(C_{is} - C_{i\ell})\frac{z_{is}}{q_s} = \frac{1}{q_s}\sum_{i\in I}(C_{is} - C_{i\ell})z_{is} \geq 0 \qquad \forall \ell, j \in J.$$

Finally, Constraint (A.3.3) is trivially satisfied and thus, for all $s \in S$, $z^s \in P$. Further, z can be written as a convex combination of $z^{(s)}$ for all $s \in S$, where the non-negative weights are given by $q_s \geq 0$ such that $\sum_{s\in S} q_s = 1$:

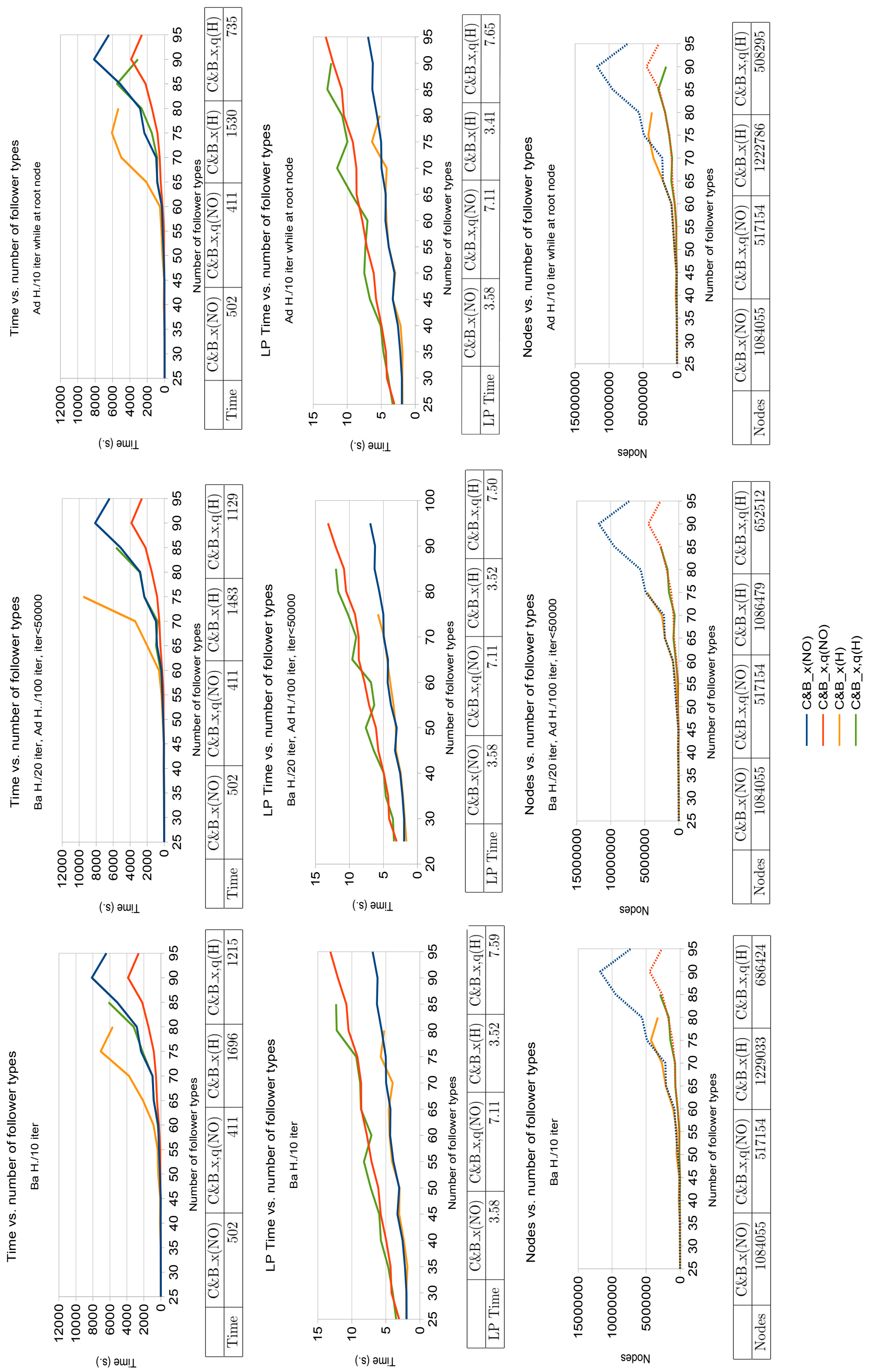
$$z_{ij} = \sum_{s\in S} q_s z_{ij}^{(s)} \qquad \forall i \in I, \forall j \in J. \tag{A.3.5}$$

To see that (A.3.5) holds, let $\tilde{J} = \{j \in J \,|\, j \neq s,\ \forall s \in S\}$, then $z_{ij}^{(s)} = 0$ for all $i \in I$ and $j \in \tilde{J}$, by definition, and, similarly, for all $i \in I$ and $j \in \tilde{J}$, $z_{ij} = 0$ since for all $j \in \tilde{J}$, $0 = q_j = \sum_{i\in I} z_{ij}$ and (A.3.3) holds. If $j \in J \setminus \tilde{J}$, then $j = \hat{s}$ for some $\hat{s} \in S$. The LHS in (A.3.5) yields $z_{i\hat{s}}$, as does the RHS:

$$\sum_{s\in S} q_s z_{i\hat{s}}^{(s)} = q_{\hat{s}} z_{i\hat{s}}^{(\hat{s})} + \sum_{s\in S: s\neq \hat{s}} q_s z_{i\hat{s}}^{(s)} = q_{\hat{s}}\frac{z_{i\hat{s}}}{q_{\hat{s}}} = z_{i\hat{s}}.$$

Thus, (A.3.5) holds. From (A.3.5) and $|S| \geq 2$, it follows that z is not a vertex of (P). ■

A.4 Effect of primal lower bound heuristics on GSG cutting plane approaches



Ba H./10 iter:

	C&B_x(NO)	C&B_x_q(NO)	C&B_x(H)	C&B_x_q(H)
Time	502	411	1696	1215
LP Time	3.58	7.11	3.52	7.59
Nodes	1084055	517154	1229033	686424

Ba H./20 iter, Ad H./100 iter, iter<50000:

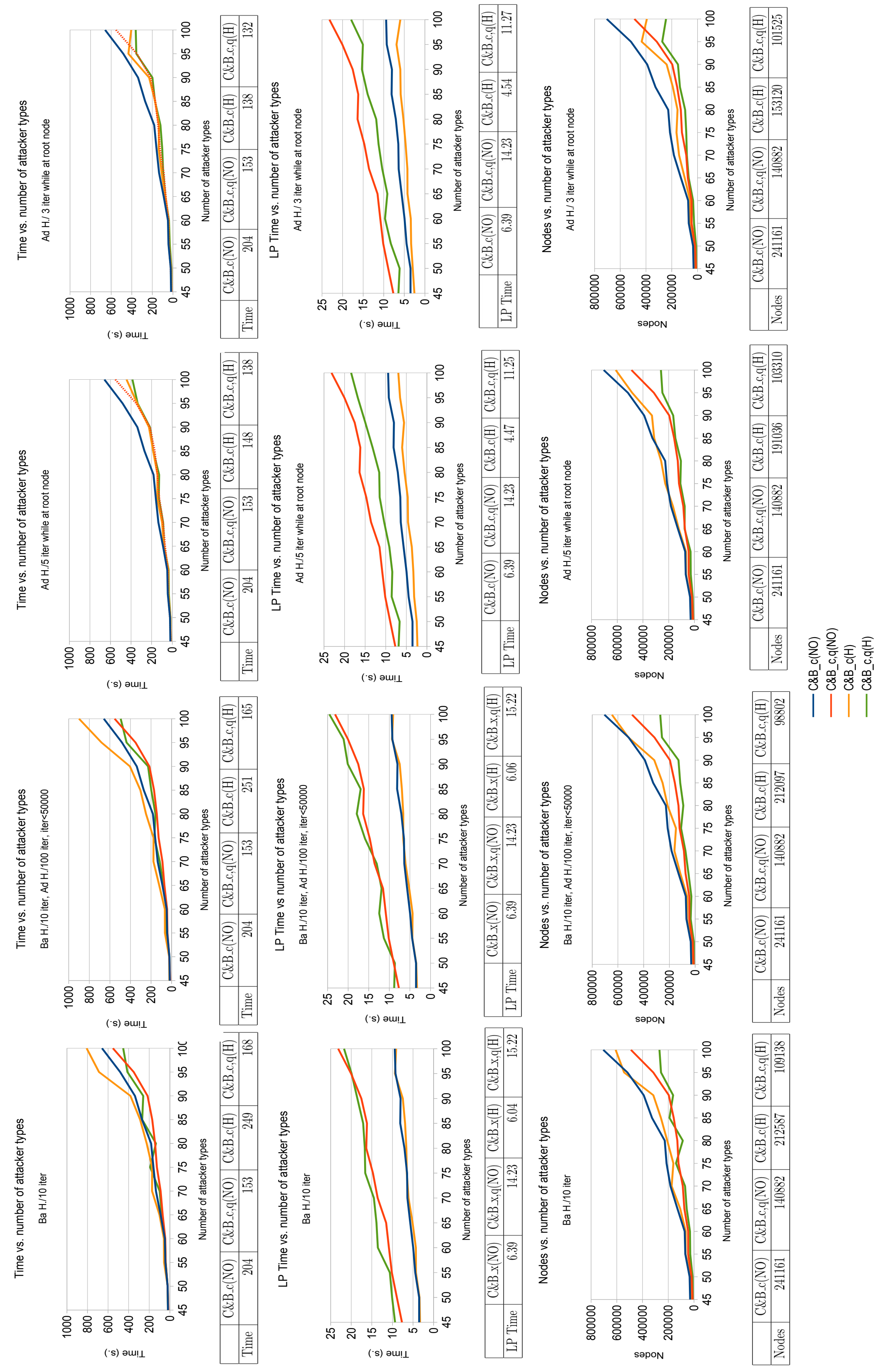
	C&B_x(NO)	C&B_x_q(NO)	C&B_x(H)	C&B_x_q(H)
Time	502	411	1483	1129
LP Time	3.58	7.11	3.52	7.50
Nodes	1084055	517154	1086479	652512

Ad H./10 iter while at root node:

	C&B_x(NO)	C&B_x_q(NO)	C&B_x(H)	C&B_x_q(H)
Time	502	411	1530	735
LP Time	3.58	7.11	3.41	7.65
Nodes	1084055	517154	1222786	508295

Figure A.4.1: Heuristic effect on general instances $I = J = \{5\}, K = \{25, \ldots, 85\}$

A.5 Effect of primal lower bound heuristics on SSG cutting plane approaches



Ba H./10 iter

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Time	204	153	249	168

	C&B.x(NO)	C&B.x.q(NO)	C&B.x(H)	C&B.x.q(H)
LP Time	6.39	14.23	6.04	15.22

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Nodes	241161	140882	212587	109138

Ba H./10 iter, Ad H./100 iter, iter<50000

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Time	204	153	251	165

	C&B.x(NO)	C&B.x.q(NO)	C&B.x(H)	C&B.x.q(H)
LP Time	6.39	14.23	6.06	15.22

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Nodes	241161	140882	212097	98802

Ad H./5 iter while at root node

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Time	204	153	148	138

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
LP Time	6.39	14.23	4.47	11.25

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Nodes	241161	140882	191036	103310

Ad H./ 3 iter while at root node

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Time	204	153	138	132

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
LP Time	6.39	14.23	4.54	11.27

	C&B.c(NO)	C&B.c.q(NO)	C&B.c(H)	C&B.c.q(H)
Nodes	241161	140882	153120	101525

Figure A.5.1: Heuristic effect on security instances $J = \{5\}, K = \{45, \ldots, 100\}, m = 2$

A.6 Recovery of an implementable defender strategy x^* from (FENCE$_{c,z,q,s,f,g}$)

Consider a small instance defined on a graph $G(V, E)$ with $V = \{v_1, \ldots, v_5\}$, $E = \{e_1, \ldots, e_6\}$ and $|J| = 6$ where $J_i = \{j_i\}$ for $i = 1, \ldots, 4$ and $J_5 = \{j_5, j_6\}$ as shown in Figure A.6.1.

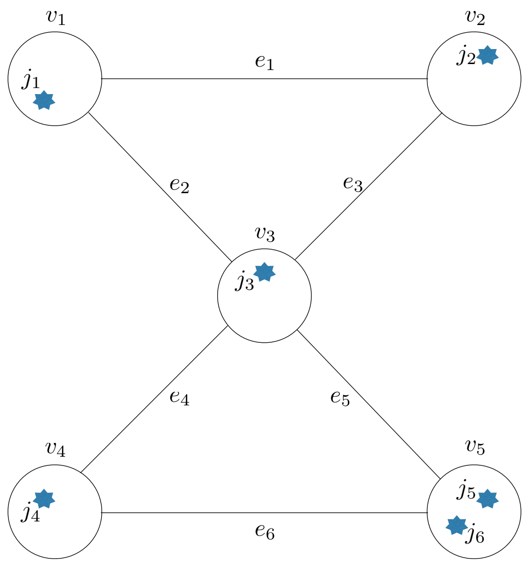


Figure A.6.1: Instance on G(V,E) with $V = \{v_1, \ldots, v_5\}$, $E = \{e_1, \ldots, e_6\}$ and $|J| = 6$

Suppose that the following values of z^*, c^* and g^* were given by (FENCE$_{c,z,q,s,f,g}$) when optimizing a problem with $m = 2$:

$$z^* = \left(1, 0, 0, \frac{1}{2}, \frac{1}{2}, 0\right)_{1\times|E|}, c^* = \left(\frac{1}{2}, \frac{1}{2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)_{1\times|J|}$$

$$g^* = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{|E|\times|J|}$$

One can easily check that (z^*, c^*, g^*) are indeed feasible for (FENCE$_{c,s,q,s,f,g}$) and could be optimal for some choice of payoff matrices D and A. To recover an implementable defender mixed strategy x, we first decompose z^*, the fractional matching of size $m = 2$ as a convex combination of integer matchings of size 2:

$$\left(1, 0, 0, \frac{1}{2}, \frac{1}{2}, 0\right) = 0.5(1, 0, 0, 1, 0, 0) + 0.5(1, 0, 0, 0, 1, 0)$$

Figure A.6.2 illustrates the decomposition. Further, Figure A.6.3 shows the construction of the box. We have six columns of height one, one for each edge in the graph G. Two

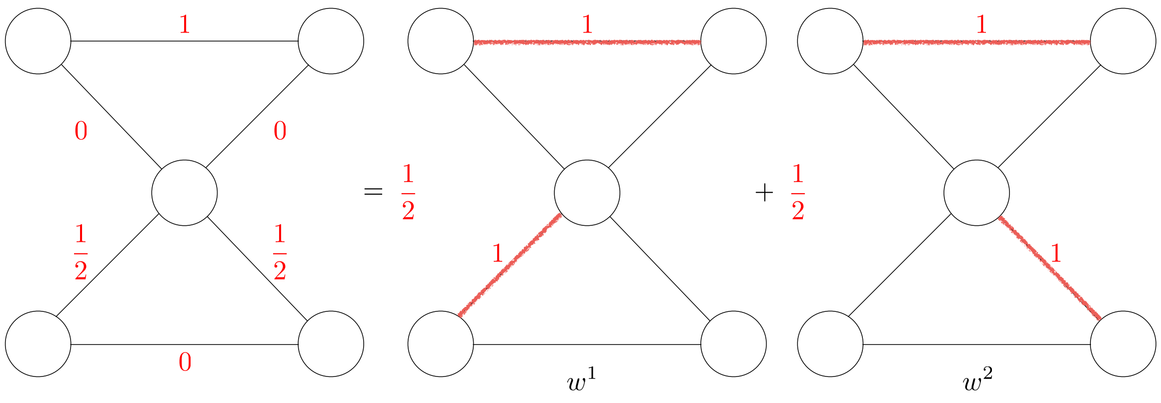


Figure A.6.2: Decomposing z^* as a convex combination of integer m-matchings w^1 and w^2

horizontal segments are drawn. One for pure m-matching w^1 of width $\lambda_{w^1} = \frac{1}{2}$ and one for pure m-matching w^2 of width $\lambda_{w^2} = \frac{1}{2}$. For each edge, segments corresponding to each of the pure matchings are blocked out if the corresponding edge does not appear in the corresponding matching (marked 'NO' in Figure A.6.3). For each edge $e \in E$, cover the segments which have not been blocked out with the values of the variables $g^*_{e,j}$ for all $j \in J_e$. Then, define x by identifying the minimum constant width horizontal segments formed in the diagram after drawing horizontal lines across all the columns along each segment's subdivisions. From figure A.6.3, one can easily identify the four pure strategies

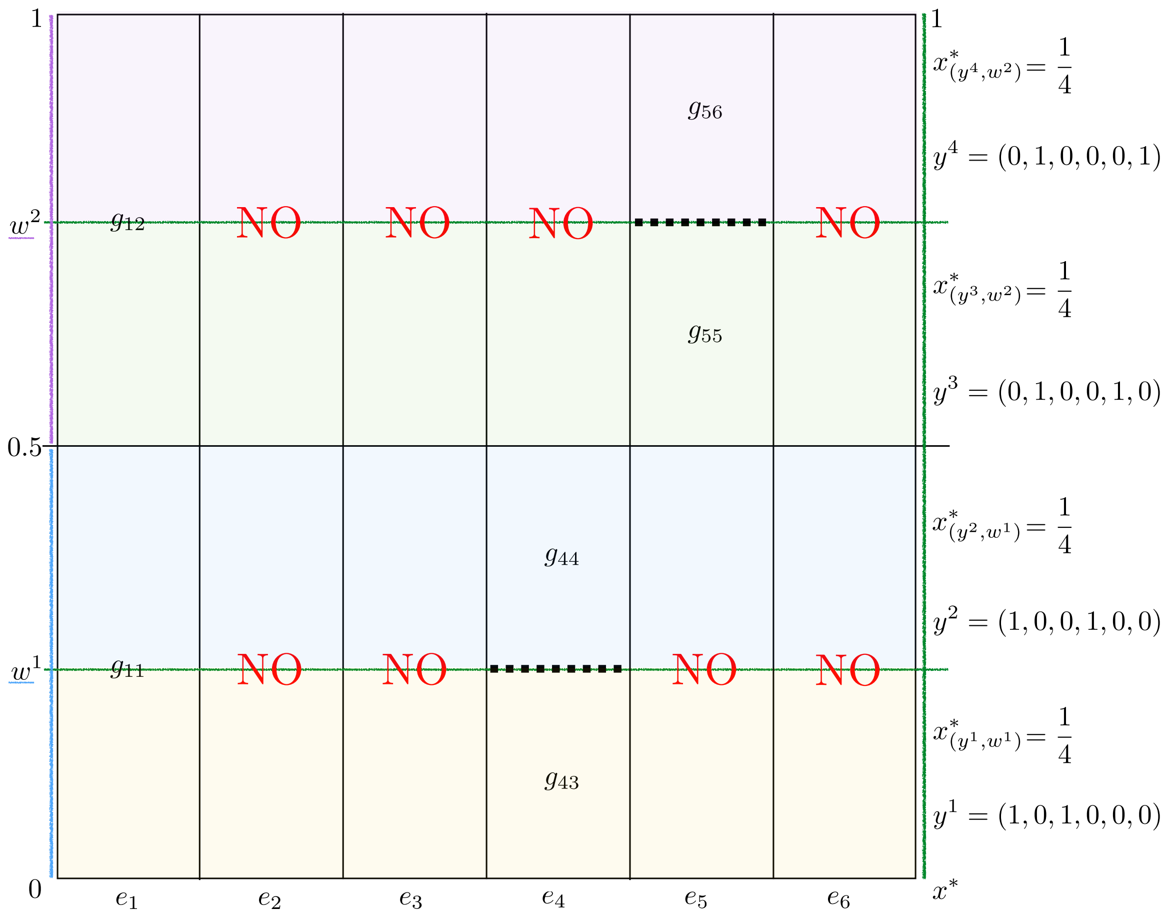


Figure A.6.3: The box method allows to recover implementable defender strategy x^*

which compose the optimal mixed strategy x^*. The mixed strategy recovered is shown in Table A.6.1. Finally, note that pure strategies recovered are indeed pure strategies in

Defender pure strategy	Edges + Targets protected	Weight in mixed strategy
1	$\{(e_1, e_4), (j_1, j_3)\}$	$\frac{1}{4}$
2	$\{(e_1, e_4), (j_1, j_4)\}$	$\frac{1}{4}$
3	$\{(e_1, e_5), (j_2, j_5)\}$	$\frac{1}{4}$
4	$\{(e_1, e_5), (j_2, j_6)\}$	$\frac{1}{4}$

Table A.6.1: Defender mixed strategy

our game, *i.e.*, each of the (y, w) recovered satisfy $(y, w) \in I$. Further, $x^* = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ is indeed a probability distribution over the four pure strategies recovered in Table A.6.1 since $x^*_{(y,w)} \geq 0$ for all $(y, w) \in I$ and $\sum_{(y,w)\in I} x^*_{(y,w)} = 1$. Note that the recovered defender mixed strategy x^* complies with the optimal solution we started with in that it satisfies:

$$\begin{aligned}
z^*_e &= \sum_{(y,w)\in I: w_e = 1} x^*_{(y,w)} \qquad \forall e \in E, \\
g^*_{ej} &= \sum_{(y,w)\in I: w_e = 1, y_j = 1} x^*_{(y,w)} \qquad \forall e \in E, \forall j \in J_e, \\
c^*_j &= \sum_{(y,w)\in I: y_j = 1} x^*_{(y,w)} \qquad \forall j \in J.
\end{aligned}$$

A.7 Determining action areas along the border using QGIS

In the case study problem discussed in Chapter 7, Carabineros needs to develop night-shift schedules to deploy joint details from paired precincts to twenty vigilance locations located along the borders of the precincts. These vigilance locations are chosen by Carabineros because they satisfy certain requirements that make them apt to conduct surveillance of criminals attempting to cross the border.

The definition of action areas around each location is a first step towards quantifying the payoffs associated to each location. To generate payoffs for the locations one can rely on geographical factors such as road density or village density in the vicinity of the location, whether or not there are any unregulated border crossing points close to the location but also other factors such as satellite images of footsteps that indicate whether or not a certain crossing has been used by migrants recently or past history of criminal arrests in the area, among many others. It is important, therefore, to define the concept of closeness to a given location $j \in J$. We model this through the definition of action areas.

We first define visibility polygons, based on which, we define the action areas. We then assume that any criminal crossing through a manned action area is automatically detected

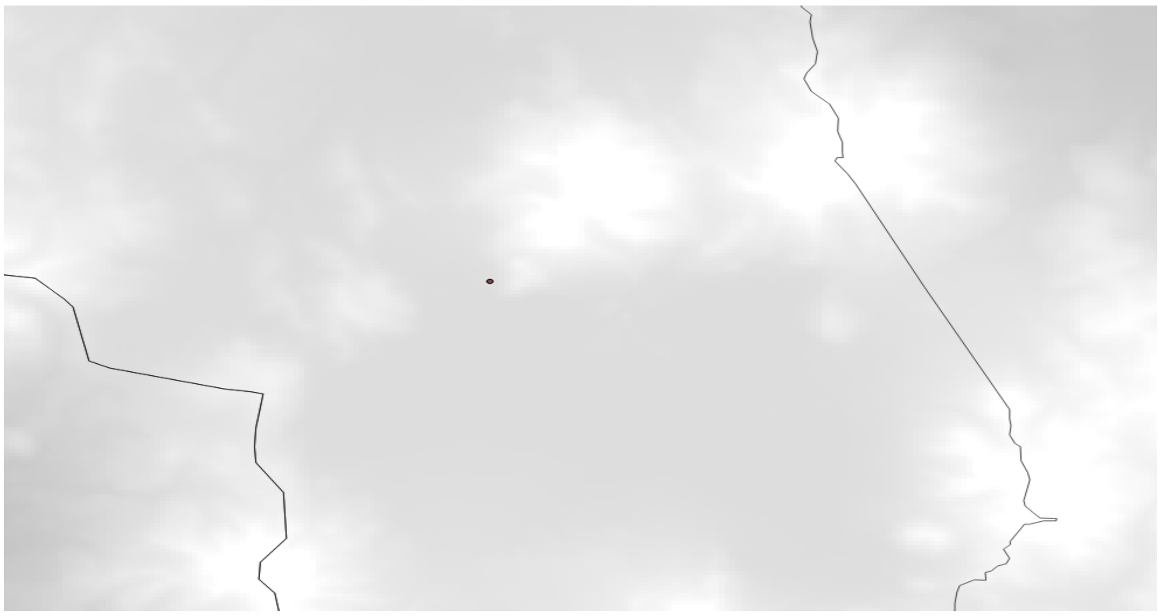
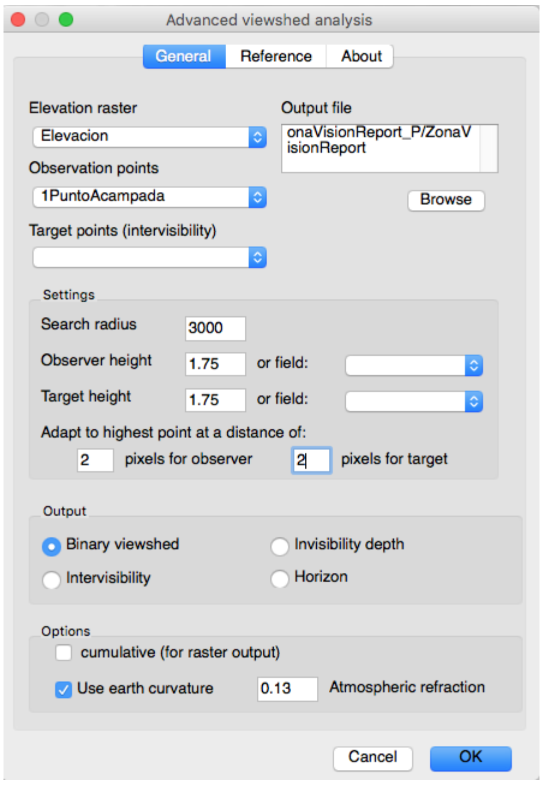


Figure A.7.1: Raster layer of DEM and shape file containing location

by Carabineros. A visibility polygon around a location is determined by how far around him a Carabinero can see from a vigilance location and this is determined by the equipment he has (binoculars, infrared goggles, . . .) as well as by the orography of the terrain. Standard night shift border patrol equipment includes binoculars that allow to see a human-sized target between three and five other away under normal weather conditions. To automatize the process of determining the visibility polygons while also taking the orography of the terrain into account, we use the Viewshed Analysis included in V2.12 of the QGIS software.

First, we obtain a *raster* layer of a Digital Elevation Model (DEM) of the XV region of Chile from a trustworthy source such as [Albers, 2015]. It is important to make sure that the different layers that we use all use the same metric. The metric used in this project is *EPSG 5361, SIRGAS Chile/UTM Zona 19S*. We then load a vector *shape* file that contains one of the vigilance outposts. In Figure A.7.1 we see the DEM, where the areas in white represent elevated areas, and the vigilance outpost is depicted by a black dot.

The Viewshed Analysis plugin can then be invoked. Figure A.7.2 shows the menu for this plugin. One loads the elevation raster and the vigilance outpost for which one wishes to compute the viewshed. One then tunes the settings such as the search radius, which in the example is set to three kilometers, the heights of the observer and the target, both taken as an average height of 1,75m, and whether or not one chooses to account for the curvature of the Earth. To fix any inaccuracies that may originate from imprecisions in the geo-referenced locations of the vigilance outposts, we allow the viewshed tool to move the observer to the highest location within two pixels of the given location. This minor concession does not significantly alter the visibility polygons but we assume that if a Carabinero is next to a elevated mass of land, he will climb on to it to improve his range of vision.


Advanced viewshed analysis

General | Reference | About

Elevation raster: Elevacion

Output file: onaVisionReport_P/ZonaVisionReport

Observation points: 1PuntoAcampada

Target points (intervisibility):

Settings

Search radius: 3000

Observer height: 1.75 or field:

Target height: 1.75 or field:

Adapt to highest point at a distance of: 2 pixels for observer, 2 pixels for target

Output

[● Binary viewshed] ○ Invisibility depth

○ Intervisibility ○ Horizon

Options

☐ cumulative (for raster output)

☑ Use earth curvature 0.13 Atmospheric refraction

Figure A.7.2: Viewshed tool

We further select the option binary viewshed, which provides a raster layer where pixels in white denote an observable area from the current location and pixels in black denote the area which cannot be observed from the current location. The raster layer returned by the binary viewshed is shown in Figure A.7.3.

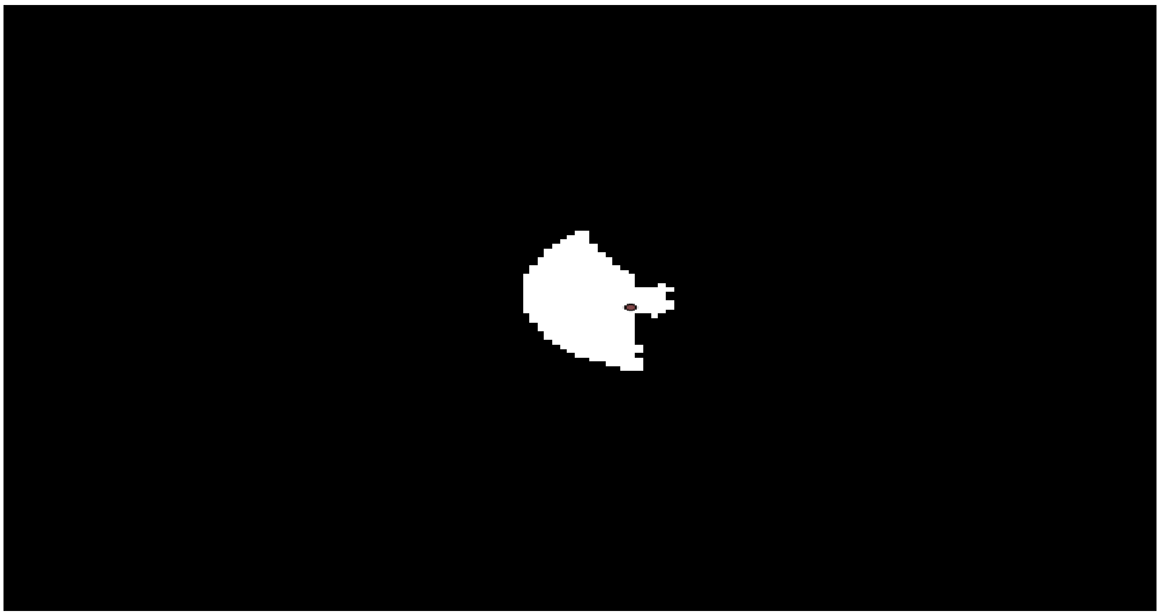


Figure A.7.3: Binary viewshed raster layer

The binary viewshed raster layer is then transformed into a shape layer and the style of the layer can then be modified so that only the visibility polygon is colored, leaving the unobservable area without any color. The resulting vector layer is shown in Figure A.7.4.

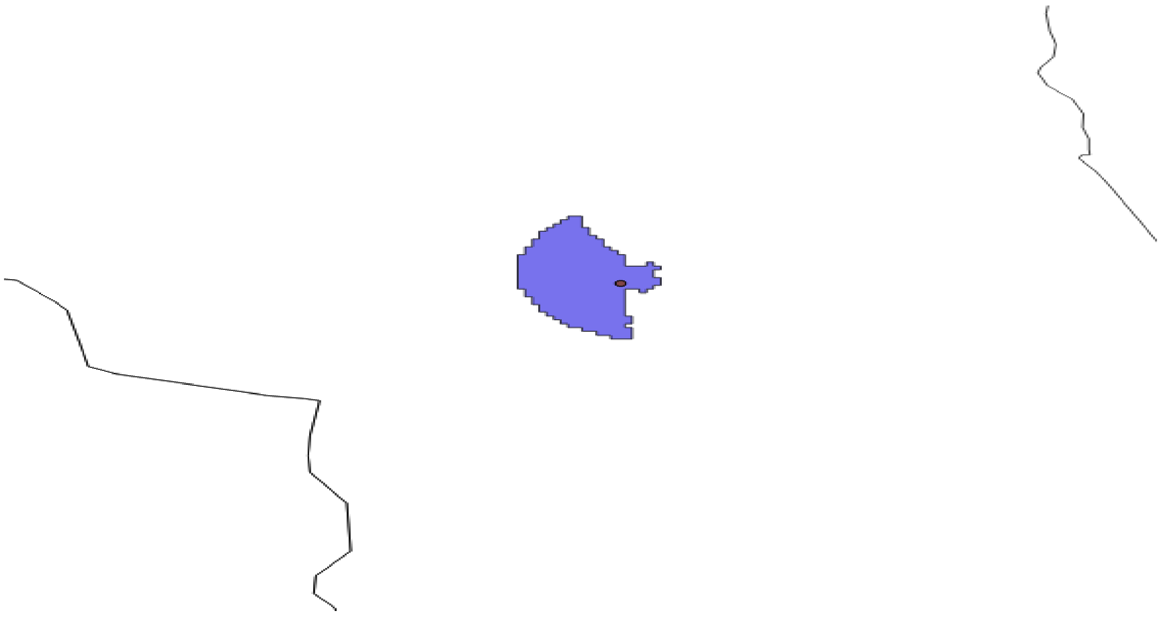


Figure A.7.4: Shape layer with the visibility polygon and vigilance location

In order to build the action areas based on the visibility polygons, we assume that the Carabineros manning a vigilance outpost can move around the location and to some extent avoid some obstacles which prevent them from having a clear line of sight. We therefore consider the convex hull of the visibility polygon as the action area. The action area can be considered as the area around the designated vigilance location where Carabineros detects the presence of a criminal crossing through it. Once an action area is constructed around a vigilance location, we may dispense with the point that marks the location and consider the action area instead. The corresponding action area in our running example is shown in Figure A.7.5.

Now, we are in a position to study factors that one may take into consideration when estimating the payoffs for the players at this location. One may want to consider road density inside the action area, previous arrests in the location and the types of these arrests or how close the vigilance area is to neighbouring settlements. In Figure A.7.6, we add this information. The vigilance outpost is marked by a yellow star and note that two roads cross through the action area, that there is a settlement, marked by a blue square, on the outskirts of the area and that there is previous history of criminal arrests in the area, signified by a red point. In this particular case, the arrest dates to July 2012, when stolen vehicles were impounded and a shipping of 2880 pairs of children's socks was intercepted. All of this information can then be taken into account when determining the payoffs for

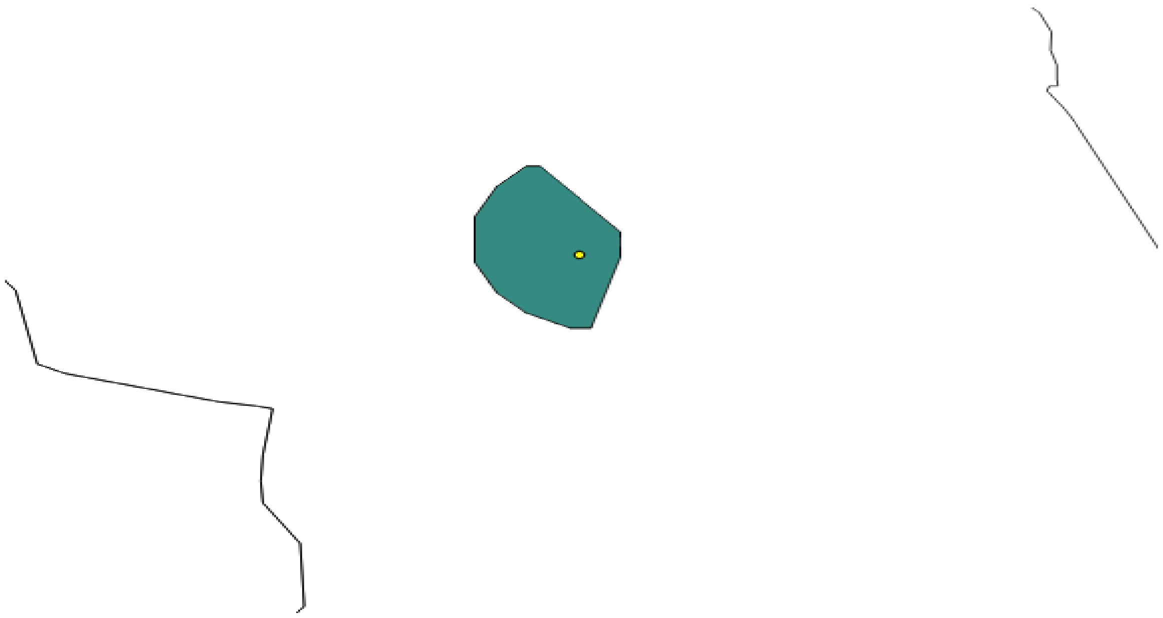


Figure A.7.5: Shape layer with the action area

Carabineros and the different types of criminals at this particular location.

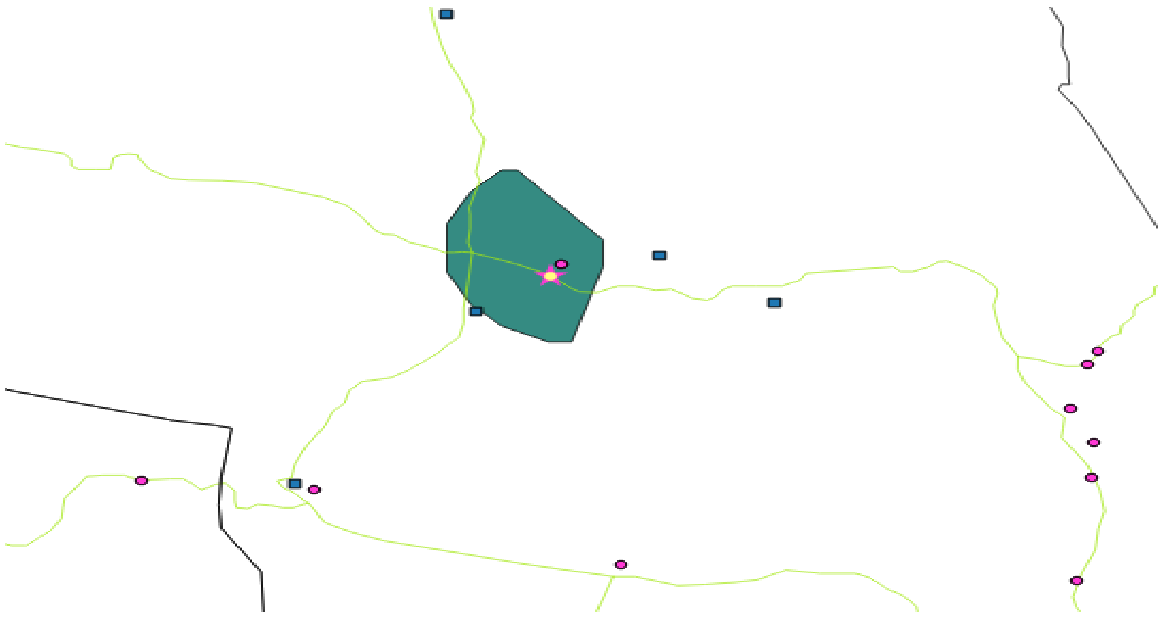


Figure A.7.6: Map of action area, roads, settlements and past criminal arrests

Further, note that the concept of an action area seems appropriate for the purpose of generating the targets in a Stackelberg game because of its simplicity to implement and adaptability. Action areas can be easily modified to fit different contexts: the visibility range, the size of the target and the height of the observer can be tuned to better describe different situations. Then, relevant information about the action areas can be exploited to generate payoffs for the different players in the game.

List of Tables

List of Figures

Vita

Carlos Casorrán Amilburu was born in Alicante, Spain on April 20th, 1987. He finished a Licenciatura (5-year undergraduate degree) in Mathematics at the Universidad de Alicante in 2011. He then went on to finish a Masters degree in Matemática Avanzada y Profesional (Advanced and Professional Mathematics) the following year at the Universidad de Murcia also in Spain. Carlos wrote his Masters thesis *Aplicaciones de la Optimización Combinatoria a la Biología Computacional* (Combinatorial optimization applications to computational biology) under the tutelage of Prof. Alfredo Marín. In 2013 he moved to Brussels, Belgium and started his Ph.D. studies on Stackelberg games and their applications to the field of security with Prof. Martine Labbé at the Université Libre de Bruxelles. In 2015 and 2016 Carlos travelled to Santiago, Chile, where he worked closely with his Ph.D. co-supervisor Prof. Fernando Ordóñez at the Universidad de Chile.

Carlos *circa* 1993