

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Mathematical Programming Games

GABRIELE DRAGOTTO

Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Mathématiques

Février 2022

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Cette thèse intitulée :

Mathematical Programming Games

présentée par **Gabriele DRAGOTTO**

en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Louis-Martin ROUSSEAU, président

Andrea LODI, membre et directeur de recherche

Adrian VETTA, membre

Jeffrey LINDEROTH, membre externe

DEDICATION

To my Family

*“Oh me! Oh life! of the questions of these recurring,
Of the endless trains of the faithless, of cities fill'd with the foolish,
Of myself forever reproaching myself, (for who more foolish than I, and who more faithless?)
Of eyes that vainly crave the light, of the objects mean, of the struggle ever renew'd,
Of the poor results of all, of the plodding and sordid crowds I see around me,
Of the empty and useless years of the rest, with the rest me intertwined,
The question, O me! so sad, recurring—What good amid these, O me, O life?”*

Answer.

*That you are here—that life exists and identity,
That the powerful play goes on, and you may contribute a verse.”*

Walt Whitman

ACKNOWLEDGEMENTS

“Words do not express thoughts very well. They always become a little different immediately after they are expressed, a little distorted, a little foolish.”

Hermann Hesse

This first acknowledgments page is a dual page: it is, simultaneously, the first fully-written page in this dissertation and the last one I wrote. The few lines below are just an underrepresentation of these three intense years filled with friendship, love, curiosity, excitement, doubts, challenges, and resilience. It is somehow simplistic to say that these years dramatically changed my scientific and personal perspectives on several matters.

“Here, on the edge of what we know, in contact with the ocean of the unknown, shines the mystery and beauty of the world. And it’s breathtaking.”

Carlo Rovelli

First of all, thanks to my advisor Andrea Lodi for making all of this possible: your curiosity, resolution, and intuition inspired me through this exciting journey. Thank you for providing support, advice, freedom, and above all, trust. Thank you for welcoming me to the incredible group you created in Montréal.

“It is our responsibility as scientists, knowing the great progress which comes from a satisfactory philosophy of ignorance, the great progress which is the fruit of freedom of thought, to proclaim the value of this freedom; to teach how doubt is not to be feared but welcomed and discussed; and to demand this freedom as our duty to all coming generations.”

Richard P. Feynman

Thank you, Margarida Carvalho and Sriram Sankaranarayanan, for the exciting intellectual stimulation and your friendship. Margarida, I sincerely thank you for showing me that great work and exemplary scientific attitude always require a sincere human approach. I am happy to have you as co-author and a great friend. Sriram, thank you for the great talks and discussions we had during these years, and for the many hours spent together.

“And the reason why I lost interest in it was because physics stopped explaining the world around me. You get into quantum mechanics and stuff, or the forces inside the atom. [...]”

You can't see an atom, all right? And you can't— it doesn't do anything that you can see and that changes the world around you, and that was why I got into physics. And so I thought, well, this is not good. And I wanted to switch to economics, because it seemed that economics was busy understanding the world around. But it was too late, I couldn't major in economics.

And so I decided to major in mathematics [...]

Ralph E. Gomory

Federico Della Croce, Fabio Salassa, and Rosario Scatamacchia, you got me in trouble way before the doctorate! Thank you for encouraging me to broaden my interests, for your support, and, essentially, for being invaluable friends. Federico, thank you for your pieces of advice: your curious and positive attitude towards research is inspirational. Fabio and Rosario, thank you for initiating me to the “asceticism” of research, for being wise colleagues, and mostly, for being great friends. Thank you for the countless discussions we had about research, academia, work, and life (and about pesto, focaccia, ciders, cyclic groups, knapsacks): I have ZERO Regrets about our time together.

“We are travelers on a cosmic journey, stardust, swirling and dancing in the eddies and whirlpools of infinity. Life is eternal. We have stopped for a moment to encounter each other, to meet, to love, to share. This is a precious moment. It is a little parenthesis in eternity.”

Paulo Coelho

To my journey companions, Antoine, Claudio, Federico: thank you for such beautiful years; thank you for being present and supportive, for our dinners, trips, laughs, delusions, and even casual misfortunes. Do you remember our trip to D.C. or at IKEA, Antoine? Federico, have you ever tried to count the dinners you organized, the talks on the office couch, or the times I fell off my skis? Claudio, did we eat more grissini or had more talks about the meaning of life? Khalid, a warm thank you for the affection and trust and our bagel breakfasts. Thank you, Didier, Giulia, Jiaqi, Leandro, Mathieu, Matteo, Mehdi, Pietro, Umberto, and the fantastic people of DS4DM.

“I live in my dreams — that's what you sense. Other people live in dreams, but not in their own. That's the difference.”

Herman Hesse

Thank you to my friends around the world: Andrea, Anna, Fugi, Giuseppe, Isa, Ylenia, Marco, Mariasole, Monika, Seba, Rick, Riccà, Umberto, the “finest Paawolians”, and “I Ragni del Ghiaccio”. Finally, the most significant tribute goes to my family – Mom and Dad, Rebecca,

Rachele, Vincenzo, Zia Lia – and Alessandra. Your love and unconditional trust provides invaluable confidence and strength. Mostly, it pushes me to be a better human being.

RÉSUMÉ

Dans de nombreux contextes de prise de décision, un agent égoïste cherche à optimiser son bénéfice compte tenu de certaines contraintes situationnelles. Mathématiquement, la tâche du décideur est souvent formulée comme un problème d’optimisation dont la solution fournit une recommandation prescriptive sur la meilleure décision. Cependant, la prise de décision est rarement une tâche individuelle : chaque décideur égoïste interagit souvent avec d’autres décideurs ayant des intérêts similaires. Cette thèse discute et propose une nouvelle perspective pour capturer la dynamique de la prise de décision stratégique impliquant plusieurs agents résolvant des problèmes d’optimisation. Nous explorons les opportunités offertes par l’interaction entre l’optimisation - en nous concentrant sur la programmation en nombres entiers mixtes (*MIP*) - et la théorie algorithmique des jeux (*AGT*) en les analysant à travers le prisme d’un cadre unifié, capable d’intégrer des éléments des deux disciplines. Nous introduisons une taxonomie pour les jeux de programmation mathématique (*MPGs*), des jeux non coopératifs simultanés où le problème de décision de chaque agent est un problème d’optimisation exprimant un ensemble hétérogène et éventuellement complexe de contraintes. Nous développons nos contributions en considérant l’équilibre de Nash comme le principal concept de solution et fondons notre recherche sur le principe suivant : dans les *MPGs*, la plausibilité des équilibres de Nash découle de la disponibilité d’outils efficaces pour les calculer et les sélectionner. En conséquence, nous fournissons des algorithmes originaux et des cadres théoriques pour caractériser, calculer et sélectionner les équilibres de Nash dans les *MPGs*.

Tout d’abord, nous abordons le problème du calcul et de la sélection des équilibres dans les jeux de programmation en nombres entiers (*IPGs*), à savoir les *MPGs* où chaque joueur résout un programme paramétré en nombres entiers. En introduisant des concepts tels que l’*inégalité d’équilibre*, l’*oracle de séparation d’équilibre*, et la *fermeture d’équilibre*, nous permettons à des outils archétypiques de la programmation en nombres entiers d’acquérir un rôle dans la théorie des jeux. Nous concevons *ZERO Regrets*, un algorithme de plans coupants pour calculer et sélectionner les équilibres dans les *IPGs*. Nous testons l’algorithme sur un jeu d’*AGT* et sur une extension multi-agents du problème du sac à dos, et nous fournissons de nouveaux résultats théoriques et informatiques sur l’efficacité de leurs équilibres.

Ensuite, nous présentons *Cut-and-Play*, un algorithme permettant de calculer les équilibres des jeux réciproquement bilinéaires (*RBGs*), une classe de *MPGs* où l’objectif de chaque joueur est linéaire par rapport à ses variables et contient des termes bilinéaires entre ses variables et celles de ses adversaires. L’algorithme calcule les équilibres en exploitant une

série d’approximations du problème d’optimisation de chaque joueur et en s’appuyant sur des méthodes de branchement et de plans coupants. Notre approche algorithmique est générale, extensible, et elle s’intègre aux solveurs de programmation mathématique existants. En pratique, elle surpasse les meilleurs algorithmes en termes de temps de calcul et d’efficacité des équilibres.

Troisièmement, nous analysons une classe de *MPGs* parmi les leaders des jeux de Stackelberg (c’est-à-dire des jeux séquentiels leader-followers) et leur application aux marchés de l’énergie. Nous prouvons qu’il est Σ_2^P difficile de décider si le jeu admet un équilibre, et nous introduisons un algorithme pour calculer et sélectionner ces équilibres. De plus, nous fournissons une étude pratique sur le marché de l’énergie chilien-argentin et offrons des perspectives de gestion basées sur les informations fournies par les équilibres.

Enfin, nous présentons *ZERO*, une bibliothèque C++ modulaire et extensible pour expérimenter avec des *RBGs*. *ZERO* fournit une boîte à outils complète d’interfaces de modélisation pour concevoir des *RBGs*, et des algorithmes pour trouver leurs équilibres de Nash. Notre engagement envers le code source ouvert vise à favoriser le développement futur, méthodologique et pratique, dans le domaine des *RBGs*.

ABSTRACT

In many decision-making settings, a selfish agent seeks to optimize its benefit given some situational constraints. Mathematically, the decision-maker’s task is often formulated as an optimization problem whose solution provides a prescriptive recommendation on the best decision. However, decision-making is rarely an individual task: each selfish decision-maker often interacts with other similarly self-interested decision-makers. This thesis discusses and proposes a novel perspective to capture the dynamics of multi-agent strategic decision-making involving multiple agents solving optimization problems. We explore the opportunities offered by the interplay of Mathematical Optimization – specifically Mixed-Integer Programming (*MIP*) – and Algorithmic Game Theory (*AGT*) by analyzing them through the lenses of a unified framework capable of integrating elements of the two disciplines. We introduce the taxonomy of Mathematical Programming Games (*MPGs*), simultaneous non-cooperative games where each agent decision problem is an optimization problem expressing a heterogeneous and possibly complex set of constraints. We develop our contributions considering the Nash equilibrium as the primary solution concept and ground our research in the following principle: in *MPGs*, the plausibility of Nash equilibria stems from the availability of efficient tools to compute and select them. Accordingly, we provide original algorithms and theoretical frameworks to characterize, compute and select Nash equilibria in *MPGs*.

First, we tackle the problem of computing and selecting equilibria in Integer Programming Games (*IPGs*), namely *MPGs* where each player solves a parametrized integer program. By devising concepts such as *equilibrium inequality*, *equilibrium separation oracle*, and *equilibrium closure*, we let archetypical tools of integer programming acquire a game-theoretic role. We design *ZERO Regrets*, a cutting plane algorithm for computing and selecting equilibria in *IPGs*. We test the algorithm on a game from *AGT* and a multi-agent extension of the knapsack problem and further provide novel theoretical and computational results on the efficiency of equilibria.

Second, we introduce *Cut-and-Play*, an algorithm to compute equilibria for Reciprocally-Bilinear Games (*RBGs*), a class of *MPGs* where each player’s objective is linear in its variables and contains bilinear terms among the player’s variables and its opponents’ ones. The algorithm computes equilibria by exploiting a series of approximations of each player’s optimization problem and leveraging branching and cutting plane methods. Our algorithmic approach is general and extensible, and it integrates with existing mathematical programming solvers; in practice, it outperforms the state-of-the-art algorithms in both computing times

and equilibria efficiency.

Third, we analyze a class of *MPGs* among the leaders of Stackelberg Games (i.e., sequential leader-followers games) and their application in energy markets. We prove it is Σ_2^P -hard to decide if the game admits an equilibrium and introduce an algorithm for computing and selecting equilibria. Further, we provide a real-world study on the Chilean-Argentinian energy market and deliver managerial insights based on the information equilibria provide.

Finally, we present *ZERO*, a modular and extensible C++ library for experimenting with *RBGs*. *ZERO* provides a comprehensive toolkit of modeling interfaces to design *RBGs*, and several algorithms to compute their Nash equilibria. Our commitment to open-source aims at fostering methodological and practical advancements in the area of *MPGs*.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
RÉSUMÉ	vii
ABSTRACT	ix
TABLE OF CONTENTS	xi
LIST OF SYMBOLS AND ACRONYMS	xvi
LIST OF APPENDICES	xvii
CHAPTER 1 – INTRODUCTION	1
1.1 Overview and Objectives	2
1.2 Contributions	3
1.3 Thesis Outline and Organization	5
CHAPTER 2 – BACKGROUND	6
2.1 Polyhedra, Optimization, and Oracles	6
2.2 Games	9
2.2.1 Integer Programming Games	12
2.2.2 Reciprocally-Bilinear Games	13
2.2.3 Stackelberg Games and Bilevel Programming	14
2.3 Equilibria, Stability, and Prices	16
CHAPTER 3 – LITERATURE REVIEW	18
CHAPTER 4 – THE ZERO REGRETS ALGORITHM: OPTIMIZING OVER PURE NASH EQUILIBRIA VIA INTEGER PROGRAM- MING	21
4.1 Introduction	21
4.2 Definitions	24
4.3 Equilibrium Inequalities	25
4.3.1 A Lifted Space	26

4.4	The Cutting Plane Algorithm and its Oracle	28
4.5	Knapsack and Network Formation Games	32
4.5.1	Knapsack Game	32
4.5.2	Network Formation Game	34
4.6	Computational Tests	35
CHAPTER 5 – THE CUT-AND-PLAY ALGORITHM: COMPUTING NASH		
EQUILIBRIA VIA OUTER APPROXIMATIONS		38
5.1	Introduction	38
5.2	Related literature and background	41
5.2.1	Background and definitions	43
5.2.2	Integer Programming Games	46
5.2.3	<i>NASPs</i>	47
5.3	Algorithmic Scheme	48
5.3.1	The Cut-and-Play Algorithm	50
5.4	Implementing the <i>ESO</i>	54
5.4.1	The $\mathcal{V}$ -Polyhedral <i>ESO</i>	55
5.5	Tailoring the <i>CnP</i>	60
5.5.1	<i>CnP</i> for <i>NASPs</i>	60
5.5.2	<i>CnP</i> for <i>IPGs</i>	61
5.6	Computational experiments	62
5.6.1	<i>IPGs</i> tests	62
5.7	Conclusions	68
CHAPTER 6 – ARTICLE 1: WHEN NASH MEETS STACKELBERG		71
6.1	Games, Definitions, ad Overview	71
6.2	Preliminaries	76
6.2.1	Definitions	77
6.2.2	Existing Results	80
6.3	Hardness of Finding a Nash equilibrium	80
6.4	An Enumeration Algorithm to find <i>MNEs</i> for <i>NASPs</i>	89
6.5	Enhancing the Algorithm	94
6.5.1	Inner Approximation Algorithm	94
6.5.2	Enhancements for <i>PNEs</i>	97
6.6	Computational Tests	97
6.6.1	Strategic Insights	99
6.6.2	Case Study – Chile-Argentina Energy Markets	101

6.6.3	Speed Analysis	102
6.7	Concluding Remarks	104
CHAPTER 7 – ARTICLE 2: ZERO: PLAYING MATHEMATICAL PRO-		
GRAMMING GAMES		105
7.1	Why Games and Equilibria?	105
7.1.1	Background.	106
7.2	Our Contributions	108
7.3	Overview	109
7.4	Modeling the Knapsack Game	111
7.5	Conclusions and Future Directions	113
CHAPTER 8 – GENERAL DISCUSSION		114
CHAPTER 9 – CONCLUSIONS AND RECOMMENDATIONS		118
9.1	Summary of Works	118
9.2	Extensions and Future Research	119
REFERENCES		121
APPENDICES		136

LIST OF TABLES

Table 4.1	Results for <i>KPG</i>	36
Table 4.2	Results for <i>NFG</i>	37
Table 5.1	<i>NASPs</i> summary results.	63
Table 5.2	<i>IPGs</i> summary results.	65
Table 5.3	<i>IPGs</i> results in absolute values.	66
Table 5.4	<i>IPGs</i> results (in percentage) with respect to the <i>m-SGM</i> . For <i>GeoT(s)</i> , and <i>#It</i> , the lower the better. As of <i>SW*</i> , the higher the better.	67
Table 6.1	Results summary of different algorithmic configurations for <i>InstanceSetA</i>	103
Table 6.2	Results summary of different algorithmic configurations for <i>InstanceSetB</i> .	103
Table B.1	Description of the parameters for <i>EPEC</i> instances.	143
Table B.2	<i>MNE</i> and <i>PNE</i> results for <i>InstanceSetA</i> . Columns: # - Instance Number. L - Number of leaders in the instance. F - Number of followers each leader has. <i>FE</i> - Time taken for full enumeration algorithm. <i>seq1</i> to <i>rand5</i> - Time taken for inner approximation with different extension strategies. <i>MNE</i> - existence (or time limit reached TL). FE-P - Time for full enumeration to find a <i>PNE</i> . <i>PNE</i> - existence (or time limit reached TL).....	144
Table B.3	<i>MNE</i> and <i>PNE</i> results for <i>InstanceSetB</i> . Same notation as table B.2. .	146
Table B.4	Instances' solutions for <i>InstanceSetInsights</i> . The columns are, in order of appearance: the instance's number, the boolean tax switch (T^a) and the trade switch (T^r). Then, the set of results associated with each of the two countries (<i>Country One</i> , and <i>Country Two</i>). In particular: the unit-energy production level <i>Prod</i> , the domestic price per unit-energy $\$(E)$, the import <i>Imp</i> and export <i>Exp</i> unit-energies, the export price $\$(E)$, and the tax per unit-emission <i>Tax</i> . Furthermore, for each of the the 3 followers of each country, we have the type <i>Ty</i> (<i>C</i> for coal, <i>G</i> for gas, or <i>S</i> for solar), the associated emission cost per unit-energy <i>E</i> , and its production <i>Prod</i>	148

LIST OF FIGURES

Figure 2.1	Optimizing $c^\top x$ over $\mathcal{G}^R$ results in the maximizer $\bar{x}$. In (a.), branching on the variable x_t results in a feasible left sub-problem, and an infeasible right sub-problem. In (b.), cutting off $\bar{x}$ from $\mathcal{G}^R$ through the cut $\pi^\top x \leq \pi_0$ (equivalent to $x_j \leq \lfloor \tilde{x}_j^t \rfloor$).	9
Figure 2.2	The class of <i>MPGs</i> and some sub-classes: <i>IPGs</i> , finite games, and <i>RBGs</i>	12
Figure 5.1	A graphical representation of the <i>CnP</i> algorithm.	52
Figure 5.2	A 2-dimensional example of algorithm 4 trying to separate $\bar{x}$ from $\text{cl conv}(\mathcal{X})$. Here, $\mathcal{X} = \{\text{conv}(v^2, \nu)\} \cup \{\text{conv}(v^1, v^3) + \text{cone}(r^1)\}$. In light blue, $\text{cl conv}(\mathcal{X})$, while in dark blue its inner approximation $\mathcal{W} = \text{conv}(v^1, v^2, v^3)$ at a given iteration of the <i>ESO</i>	58
Figure 6.1	A schematic representation of a <i>NASP</i> . The vertical arrows are Stackelberg interactions (i.e, sequential decisions), while the horizontal ones are Nash interactions (i.e, simultaneous decisions).	72
Figure 6.2	A pictorial reprsentation of algorithm 5.	92
Figure 7.1	A schematic view of ZERO's modules, 10000 lines of code, 50 files, 40 classes, and 450 functions. The <i>namespaces</i> are in gray, and the relative content is grouped below. The primitive classes are in purple, and the associated inheritor classes are in blue. Nested namespaces are in green.	110
Figure 7.2	An Example of a C++ instantiation of a 2-player Knapsack Game in ZERO	112

LIST OF SYMBOLS AND ACRONYMS

AGT	Algorithmic Game Theory
CNP	Cut-And-Play
EPEC	Equilibrium Problem with Equilibrium Constraints
ESO	Equilibrium Separation Oracle
GT	Game Theory
IEDS	Iterated Elimination of Dominated Strategies
IPG	Integer Programming Game
LCP	Linear Complementarity Problem
MIP	Mixed Integer Programming
MNE	Mixed Nash Equilibrium
MPG	Mathematical Programming Game
NASP	Nash Game Among Stackelberg Players
OSW	Optimal Social Welfare
PNE	Pure Nash Equilibrium
PAG	Polyhedrally Aproximated Game
POA	Price of Anarchy
POS	Price of Stability
PRLP	Point-Ray Linear Program
KPG	Knapsack Game
KKT	Karush-Kuhn-Tucker (conditions)
RBG	Reciprocally-Bilinear Game
SSI	Subset Sum Interval

LIST OF APPENDICES

Appendix A	THE CUT AND PLAY ALGORITHM	136
Appendix B	WHEN NASH MEETS STACKELBERG	138

CHAPTER 1 INTRODUCTION

In many decision-making settings, a selfish agent seeks to optimize its benefit given some situational constraints. Mathematically, the decision-maker’s task is often formulated as a Mixed Integer Programming (*MIP*) problem, a mathematical program where some discrete variables represent indivisible quantities or choices. *MIP* offers compelling modeling capabilities for real-world problems, and its application domains include, to name a few, airline scheduling, kidney exchange programs, energy dispatching, and production planning. However, decision-making is rarely an individual task: each selfish decision-maker often interacts with other analogously self-interested decision-makers.

In the past two decades, Algorithmic Game Theory (*AGT*) emerged to deepen the understanding of these complex interactions among self-interested agents, especially in the context of the Internet [118]. Generally, *AGT* represents such interactions through games among self-interested agents and analyzes their outcomes’ efficiency (i.e., quality) and computability properties. In many real-life applications, the agents – individuals, organizations, or algorithms – have limited resources in terms of time, money, and computing power, and thus optimization and computability matters play crucial roles.

This thesis stands at the interface of *AGT* and *MIP* and studies the strategic interaction of selfish agents – or players – optimizing their benefits through mathematical programs. As we showcase, we believe singular opportunities stem from a better interplay between *AGT* and *MIP*. First of all, the two disciplines share two crucial elements: they both provide powerful modeling capabilities for real-world problems and devote significant effort to computability matters. Further, they also share a common origin, even from a historical perspective.

Historical perspective. In the aftermath of World War II, two of the most important paradigms of both Game Theory (*GT*) and Mathematical Programming originated in Princeton and shared John Von Neumann as a common ancestor.¹ Von Neumann discovered what is now known as linear programming duality, an elegant and essential theory at the core of Mathematical Programming theory and practice. As George Dantzig later evoked, Von Neumann discovered duality through the lenses of *GT*: “I have just recently completed a book with Oscar Morgenstern on the Theory of Games. What I am doing is conjecturing that the two problems [primal and dual linear program] are equivalent.” [48]. The pioneering book of Morgenstern and Von Neumann [115] – along with the seminal contributions of

¹A brief review of the historical events is in [138].

Nash [116, 117] – transformed the scientific perspective on strategic behavior. Despite Von Neumann’s obsession with computability, the physical tools available at the time were not sufficiently mature to propel significant advances in computational *GT*. Decades later, with the advent of the Internet and the availability of powerful computational platforms, game theorists became increasingly interested in *GT* frameworks and their connection to algorithms. Indeed, the Internet promoted the development of *AGT*, a discipline focusing on the interaction between *GT* and algorithms, with a distinct spotlight on computability. As Scott Shenker said, “*the Internet is an equilibrium, we just have to identify the game*” [118].

This thesis provides new paradigms to better grasp the dynamics of multi-agent decision-making in competitive settings, where agents are solving optimization problems. In particular, we study what we define as Mathematical Programming Games (*MPGs*), i.e., simultaneous games where the agents (players) solve optimization problems. In *MPGs*, the strategic interaction takes place in the objective of each player, namely, each player’s objective includes variables from its opponents’ optimization problems. The family of *MPGs* covers many known games: for instance, some classes of bilevel programs, Integer Programming Games (*IPGs*), and even paradigmatic games from *AGT* such as Network Formation Games [5]. For the purpose of this dissertation, we employ the Nash equilibrium [116, 117] as the leading solution concept and center our contributions around computing and selecting (Nash) equilibria in several classes of *MPGs*.

1.1 Overview and Objectives

This thesis proposes new perspectives to analyze the strategic interaction of players solving mathematical programs and specifically focus on the interplay of *AGT* and *MIP*. We devise the concept of *MPG* to provide a unifying framework for this interplay, and we frame our contributions on both methodological and practical perspectives. Our research employs the Nash equilibrium as the leading solution concept, and it centers around the following principle: in *MPGs*, the plausibility of Nash equilibria stems from the availability of efficient tools to compute and select them. Broadly, the research we propose tackles the following fundamental questions:

- (i.) Can *MPGs* accurately model real-world problems?
- (ii.) How different Nash equilibria compare in terms of efficiency (i.e., quality)?
- (iii.) How do we build efficient algorithms to compute and select such equilibria?
- (iv.) What are the theoretical properties of these algorithms?

- (v.) What are the prescriptive insights that equilibria provide in real-world applications?
- (vi.) Can *MPGs*' equilibria promote socially beneficial outcomes?

From a methodological perspective, we aim to understand the theoretical and algorithmic aspects of computing and selecting Nash equilibria in *MPGs*. This includes, for instance, the design of efficient algorithms and theoretical frameworks to characterize and compute equilibria and the computational complexity associated with determining their existence. From a practical perspective, we aim to provide a framework for extending the broad family of problems involving typical *MIP* applications – to name a few, logistics, scheduling, tactical decision-making – to a multi-agent setting. In this context, we aim to identify some applications domains where *MPGs* provide better decision-making insights and protect the collectivity of decision-makers from socially harmful selfish behaviors. We articulate this thesis in four self-contained chapters, each developing a research theme intersecting the abovementioned goals.

1.2 Contributions

The ZERO Regrets Algorithm: Optimizing over Pure Nash Equilibria via Integer Programming (Chapter 4, [57]) This first work addresses the issue of computing and selecting equilibria in a class of *IPGs* where objectives are linearizable through standard *MIP* techniques. We introduce *ZERO Regrets*, a cutting plane algorithm to compute, enumerate, and select Pure Nash equilibria (*PNEs*), i.e., equilibria involving only one pure action per player. In practice, the algorithm solves a difficult (i.e., $\mathcal{NP}$ -hard) problem while simultaneously having access to an oracle solving a likewise difficult problem. We introduce the concept of *equilibrium inequality*, namely an inequality that is possibly invalid for non-equilibria strategies but always valid for any strategy appearing in at least a game's *PNE*. We provide a class of equilibrium inequalities and prove – by devising the concept of *equilibrium closure* – they are sufficient to describe the convex set containing all the *PNEs*. *ZERO Regrets* manages to compute efficient (i.e., with a low price of stability) *PNEs* for Network Formation Games, a well-studied family of games in *AGT*, and for Knapsack Games (i.e., *IPGs* where each player solves a binary knapsack problem). Further, we characterize the computational complexity of determining if a Knapsack Game admits a *PNE* and provide theoretical bounds on its equilibria's efficiency (i.e., the price of stability and anarchy). Notably, we formulate a paradigmatic problem of *AGT* – the Network Formation Game – through an *IPG* and select its equilibria through *ZERO Regrets*. Up to our knowledge, this contribution is the first one providing a computationally efficient algorithm to select and enumerate *PNEs* in *IPGs*.

The Cut and Play Algorithm: Computing Nash Equilibria via Outer Approximations (Chapter 5, [33]) We study the problem of computing an *MNE* in Reciprocally-Bilinear Games (*RBGs*), namely *MPGs* where each player’s objective is linear in its variables and contains bilinear terms in its variables and the ones of its opponents. We prove the set containing all the (mixed) strategies for any player i is the polyhedron given by the convex hull of the feasible region of its optimization problem. Starting from this result, we introduce *Cut-and-Play*, an algorithm to compute Mixed Nash Equilibria (*MNEs*) – a generalization of *PNEs*– in *RBGs*. This cutting plane algorithm computes equilibria by exploiting a series of approximations of each player’s optimization problem and leveraging branching and cutting plane methods. The Cut-and-Play algorithm is general and extensible, and it integrates with existing mathematical programming solvers. In practice, it outperforms the state-of-the-art algorithms in both computing times and equilibria efficiency.

When Nash Meets Stackelberg (Chapter 6, [31]) We explore a class of simultaneous non-cooperative games among the leaders of continuous Stackelberg games (*NASP*), i.e., leader-followers games. In *NASPs*, each leader solves a linear bilevel program, while followers solve convex quadratic problems. From a complexity perspective, we characterize the computational complexity of determining whether an instance of this game has an equilibrium or not into a Σ_2^P -hard decision problem. Furthermore, we devise a series of algorithms based on enumerative procedures to compute and select equilibria. We provide a contextualization of *NASPs* in the realm of energy markets and provide extensive computational tests on a range of *NASPs* instances. Finally, we propose a real-world study on a simplified version of the Chilean-Argentinian energy market and derive some managerial insights from the game’s Nash equilibria.

ZERO: Playing Mathematical Programming Games (Chapter 7, [58]) In this last contribution, we provide a software library for equilibria computation in *MPGs*. We present *ZERO*, a modular and extensible C++ library interfacing Mathematical Programming and *GT*. *ZERO* provides a comprehensive toolkit of modeling interfaces for designing *MPGs*, helper tools, and algorithms to find Nash equilibria. Specifically, the software supports *RBGs*, *IPGs*, and extended support for integer non-convexities, linear bilevel problems, and linear equilibrium problems with equilibrium constraints. The library is modular and provides all the elementary ingredients for devising algorithms and implementations for *RBGs*. By releasing *ZERO*, we aim to encourage methodological advancements in *MPGs* and lower the associated entry barriers.

1.3 Thesis Outline and Organization

We organize this dissertation as follows. In Chapter 2, we formalize the background notions and definitions, while, in Chapter 3, we briefly provide a literature overview by providing the main pointers to the previous works. We complement this literature review by adding additional relevant elements in each chapter. In Chapters 4 to 7, we present and develop the previously outlined contributions as self-standing works. In Chapter 8, we discuss the contributions from a unified perspective, highlighting the transversal elements and the interactions among works. Finally, in Chapter 9, we summarize our work and elaborate on its strengths, limitations, and future research directions.

CHAPTER 2 BACKGROUND

This section lays out the background definitions and the common language we use throughout the dissertation. The content in the sequel should serve as a self-contained primer on the diagonal concepts we deal with in the following chapters.

2.1 Polyhedra, Optimization, and Oracles

We review some basic concepts of polyhedral theory. We employ some standard definitions from convex [20] and polyhedral [42] theories. Given a convex set $K \in \mathbb{R}^k$, we denote as $\text{cl}(K)$, $\text{int}(K)$, $\text{bd}(K)$, $\text{dim}(K)$, the *closure*, *interior*, *boundary* and *dimension* of K . Given a set A , let $\text{conv}(A)$ and $\text{cone}(A)$ be the *convex hull* and the *conic hull* of elements in A , respectively. Given $(\pi, \pi_0) \in \mathbb{R}^k \times \mathbb{R}$, we define as *valid inequality* for K a linear inequality in the form $\pi^\top x \leq \pi_0$ holding for any $x \in K$. A *cut* is a valid inequality $\pi^\top x \leq \pi_0$ for K violated by $\tilde{x} \in \mathbb{R}^k$, i.e., $\pi^\top \tilde{x} > \pi_0$ and $\pi^\top x \leq \pi_0$ for any $x \in K$. We define the *separation oracle* as the blackbox solving the separation problem in Definition 1.

Definition 1 (Separation Problem). *Given a closed convex set $\bar{K}$, and a point $\bar{x}$, either: (i.) determine that $\bar{x} \in \bar{K}$ and output **yes**, or (ii.) determine that $\bar{x} \notin \bar{K}$, and output **no** and a cut $(\pi, \pi_0) \in \mathbb{R}^k \times \mathbb{R}$ for $\bar{x}$.*

Polyhedra. A *polyhedron* $P \subseteq \mathbb{R}^p$ is a convex set generated by the intersection of finitely many halfspaces (i.e., a set of linear inequalities). Given a valid inequality $\pi^\top x \leq \pi_0$ for P , the set $F = P \cap \{x \in \mathbb{R}^p : \pi^\top x = \pi_0\}$ is a *face* for P . A point $x \in P$ is an *extreme point* if it is not a convex combination of two points in P , i.e., with $0 < \alpha < 1$. A recession direction for P is a vector $r \in \mathbb{R}^p$ so that, for any $x \in P$, $x + \alpha r \in P$ for any $\alpha \geq 0$. An extreme ray for P (or an extreme recession direction) is a recession direction for P that is not a convex combination of two or more other recession directions of P . We define $\text{rec}(P)$ and $\text{ext}(P)$ as the set of recession directions and extreme points of P , respectively. Hence, P is a combination of its extreme points $V = \text{ext}(P)$ and recession directions $R = \text{rec}(P)$, i.e., $P = \text{conv}(V) + \text{cone}(R)$. The *face* induced by an inequality $\pi^\top x \leq \pi_0$ on P is the polyhedron $F(\pi, \pi_0) = \{x \in P : \pi^\top x = \pi_0\}$. If $\text{dim}(F(\pi, \pi_0)) = 0$, the face is called a *vertex* (an extreme point) of P , while if $\text{dim}(F(\pi, \pi_0)) = \text{dim}(P) - 1$, we have a *facet* of P . The *minimal description* of P is the set of all the *facets* of P .

MIP problem. We define as *MIP* problem an optimization problem where some variables are required to be integer, as in (2.1). Specifically, we consider *MIP* problems where the objective function is linear, and finitely many linear constraints and integer requirements represent the feasible region. Thus, the optimization problem reads as

$$\max_x \{c^\top x : x \in \mathcal{G}\}, \quad \mathcal{G} = \{Ax \geq b, x \geq 0, x_j \in \mathbb{Z} \ \forall j \in \mathcal{I}\}. \quad (2.1)$$

In this formulation, $\mathcal{I}$ encapsulates the indexes of integer-constrained variables, the matrix A and the vector b have integer entries (i.e., they describe a rational polyhedron). Further, we assume the system $Ax \geq b$ has no particular structure. We define $\mathcal{G}^R$ in (2.2) as the *linear relaxation* of $\mathcal{G}$, namely the polyhedron given by the linear inequalities in $\mathcal{G}$ (and without the integrality requirements), so that

$$\mathcal{G}^R = \{Ax \geq b, x \geq 0, L_j \leq x_j \leq U_j \ \forall j \in \mathcal{I}\}. \quad (2.2)$$

In (2.2), L_j and U_j are the lower and upper bounds of the j -th integer-constrained variable. Most of the procedures we present can generalize to other classes of objective functions (i.e., convex quadratic objective functions), we restrict our exposition to the case of linear objective functions. In the *MIP* problem of (2.1), $\mathcal{G}^R$ is a polyhedron and so is $\text{conv}(\mathcal{G})$: in particular, the latter is the so-called *perfect formulation* of the problem in (2.1). In other words, the solution to the linear program $\max_x \{c^\top x : x \in \text{conv}(\mathcal{G})\}$ is the optimal solution of (2.1) fulfilling all the integrality requirements and linear constraints. Meyer's theorem [114] guarantees there always exists a perfect formulation for the problem in (2.1) (if the coefficients in (2.1) are all rational). Nevertheless, the perfect formulation may be of exponential size with respect to the input data. In practice, one only needs an intermediate polyhedron between $\mathcal{G}^R$ and $\text{conv}(\mathcal{G})$ so that optimizing $c^\top x$ over it yields an integer-feasible solution. Two classic methods for solving *MIPs* are the *Branch-and-Bound* algorithm [101] and the *cutting plane* algorithm [79].

Branch-and-Bound. The Branch-and-Bound algorithm splits the problem (2.1) in a series of smaller sub-problems in order to implicitly explore the solution space. A tree graph maps the relationship among the different sub-problems: each node in the tree corresponds to a sub-problem, and each edge points to a father-child relation. Each node is a *MIP* problem – possibly with some extra (invalid) inequalities enforcing additional bounds on the problem's discrete variables – whose relaxation gives a bound on (2.1). Let t be the index associated with each sub-problem, let $\bar{x}^t$ be the solution of node t , and let $\mathcal{B}_t = \{i \in \mathcal{I} : \bar{x}_i^t \notin \mathbb{Z}\}$ be the set of fractional variables in $\bar{x}^t$ that also belong to $\mathcal{I}$. The algorithm starts from a root

node $t = 0$, which is associated with the linear relaxation of (2.1), and splits the problem exploiting the information in $\mathcal{B}_t$. Specifically, the problem of node t generates two child problems t_{DOWN} and t_{UP} induced by picking one fractional variable $j \in \mathcal{B}_t$, and enforcing that $x_j \leq \lfloor \bar{x}_j^t \rfloor$ in t_{DOWN} , and $x_j \geq \lceil \bar{x}_j^t \rceil$ in t_{UP} . Clearly, these linear inequalities are invalid for the problem in (2.1), yet the two sub-problems t_{DOWN} and t_{UP} together necessarily contain all the solutions to the original problem. The splitting mechanism generating t_{DOWN} and t_{UP} is the so-called *branching* procedure, which also excludes $\bar{x}^t$ from the solutions space. Further, the relaxations in t_{DOWN} and t_{UP} inherit an upper bound from their parent node t , namely, their optimal value cannot improve the one of t . By exploiting this bound hierarchy, the algorithm implicitly explores the search space. For instance, assume that $\tilde{x}$ is a feasible solution for (2.1); then, any node with a bound not improving the one given by $\tilde{x}$ cannot improve the bound of the latter. Therefore, the algorithm *fathoms* any node (i.e., close any node) that does not have the chance to improve the best incumbent solution so far. In practice, this implies that the Branch-and-Bound implicitly explores the solution space by only considering “promising” nodes, i.e., nodes potentially containing improving solutions. Consequently, finding good quality solution early in the tree means generating smaller trees and possibly proving optimality quickly. Further, the *node selection* strategy – the strategy employed for selecting the next node to consider – and the *variable selection* strategy – the strategy by which one picks a fractional variable in $\mathcal{B}_t$ at node t – often play essential roles in terms of speeding up the algorithm’s convergence. Finally, in *MIPs* as the one in (2.1), the Branch-and-Bound terminates in a finite number of steps and outputs an optimal solution or proves the problem is infeasible or unbounded. Figure 2.1a provides a visualization of an iteration of the branching procedure.

Cutting plane algorithms. The second fundamental class of algorithms is the one of cutting planes methods. We specifically describe an abstraction of the cutting plane algorithm introduced by Gomory [79]. The algorithm starts from the polyhedron $\mathcal{G}^R$, and iteratively refines it by adding valid inequalities for $\text{conv}(\mathcal{G})$. Let $\mathcal{G}^C = \mathcal{G}^R$, and let $\bar{x}$ be the maximizer of $\max_x \{c^\top x : x \in \mathcal{G}^C\}$. If $\bar{x}$ does not fulfill the integrality requirements in $\mathcal{I}$, then $\bar{x} \in \mathcal{G}^C \setminus \text{conv}(\mathcal{G})$. Equivalently, a separation oracle – given $\bar{x}$ and an implicit description of $\text{conv}(\mathcal{G})$ – outputs a cut $\pi^\top x \leq \pi_0$ valid for $\text{conv}(\mathcal{G})$ and cutting off $\bar{x}$. This latter cut becomes part of the description of $\mathcal{G}^C$, namely $\mathcal{G}^C = \mathcal{G}^C \cap \{x : \pi^\top x \leq \pi_0\}$, and the process restarts by computing $\bar{x}$ again. The algorithm terminates when the optimization problem is infeasible or unbounded, or $\bar{x}$ fulfills the integrality requirements. The cutting plane algorithm will rarely terminate with the perfect formulation $\text{conv}(\mathcal{G})$. Indeed, one expects to retrieve an intermediate polyhedron $\mathcal{G}^C$ between $\text{conv}(\mathcal{G})$ and $\mathcal{G}^R$ over which optimizing a given linear

function $c^\top x$ results in a $\bar{x}$ satisfying the integrality requirements. In other words, one expects to refine $\mathcal{G}^R$ until the vertex $\bar{x}$ in the direction given by c belongs also to $\text{conv}(\mathcal{G})$. Figure 2.1b provides a visualization of an iteration of a cutting plane algorithm.

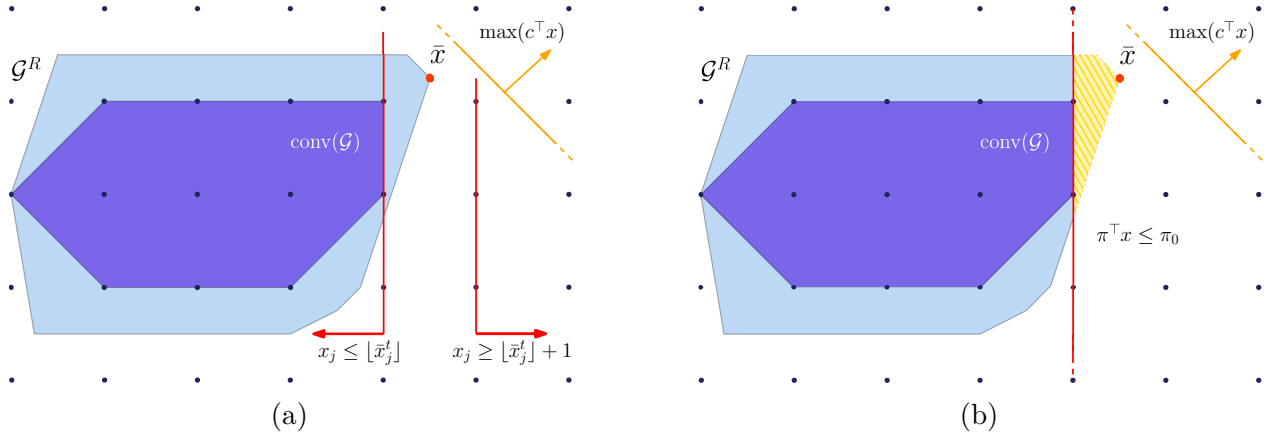


Figure 2.1 Optimizing $c^\top x$ over $\mathcal{G}^R$ results in the maximizer $\bar{x}$. In (a.), branching on the variable x_t results in a feasible left sub-problem, and an infeasible right sub-problem. In (b.), cutting off $\bar{x}$ from $\mathcal{G}^R$ through the cut $\pi^\top x \leq \pi_0$ (equivalent to $x_j \leq \lfloor \bar{x}_j^t \rfloor$).

Branch-and-Cut. The seminal work of Padberg and Rinaldi [120] combined both branching and cutting procedures in the so-called Branch-and-Cut algorithm. The authors introduced the algorithm in the context of the traveling salesman problem, although their approach is general and applies to any *MIP* in the form of (2.1). The work from Padberg and Rinaldi provides – also from a chronological point of view – the first efficient combination of a cutting plane method, a branching scheme, and a rich set of heuristics. We refer the reader to [1, 15, 106] for more detailed surveys on the *MIP* technology.

2.2 Games

GT lies at the interface of Mathematics, Economics, and Social Sciences. This discipline attempts to shed new light on the dynamics with which individuals and groups behave and make decisions, often in contexts such as markets, governmental institutions, and, more recently, the Internet. The ultimate goal is twofold; first, an analytical understanding of the decision-making process and its drivers. Second, a prescriptive recommendation on how an agent should act to shield against the misbehavior and selfishness of its peers.

A common language. We formally introduce some of the game-theoretic terms we will use in the dissertation. We call a *game* a set of *players* (or agents) making a decision with a

given order of play. Each player chooses an alternative from its set of feasible alternatives, or *actions*. When all players decided, each player's outcome – or its *payoff* – is a function of its decision and the ones of the other players. A game has *perfect information* if each player, when deciding, is perfectly aware of any choices that happened before its decision. Further, a game has *complete information* if every player is aware of the payoff functions and the actions of every other player. We call a player *rational* if it would never play an action hurting (i.e., decreasing when it maximizes its benefit) its payoff, i.e., a player would never play an action yielding a payoff of k if it can get $k + \epsilon$ with $\epsilon > 0$. Players may decide *simultaneously*, as one would do in rock paper scissors, or *sequentially*, namely following a given order of priority. We define a game as *finite* if the set of actions available to each player is finite, there are finitely many players and outcomes, and the game does not continue indefinitely but terminates at some point.

This thesis focuses on simultaneous and sequential games with complete and perfect information where *all* players are rational. Most importantly, we assume there exists a preferable representation of the decision problems that players face. Specifically, we assume one can represent each player's decision-making problem as a mathematical program. We formalize these assumptions in Definition 2 with the concept of *MPG*. As a standard game theory notation, let x^i denote the vector of variables of player i , and let the operator $(\cdot)^{-i}$ be $(\cdot)$ except i . The vector $x^{-i} = (x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^n)$ represents the variables of i 's *opponents* (all players but i).

Definition 2 (Mathematical Programming Game). *A Mathematical Programming Game (MPG) among n players is a simultaneous game where each rational player $i = 1, 2, \dots, n$ solves the optimization problem*

$$\max_{x^i} f^i(x^i, x^{-i}) \quad (2.3a)$$

$$\text{s.t. } x^i \in \mathcal{X}^i \quad (2.3b)$$

where $\mathcal{X}^i$ and $f^i(x^i, x^{-i}) : \prod_{j=1}^n \mathcal{X}^j \rightarrow \mathbb{R}$ are the set of actions and the payoff function of i , respectively.

The optimization problem of player i is *parametrized* in x^{-i} , namely plugging x^{-i} as a parameter results in an optimization problem purely expressed in the variables x^i . Furthermore, each $\mathcal{X}^i$ only includes the variables from the respective player i . Depending on the structure of $\mathcal{X}^i$, *MPGs* may represent several well-studied classes of games.

We define *MPGs* intending to provide a general and unified category for games represented through mathematical programs. Indeed, the taxonomy we propose follows from the following

observations:

- (i.) *Actions representation.* We implicitly represent each player's action space with a set $\mathcal{X}^i$. The action space may be unbounded and contain infinitely or finitely many elements. This representation enables the understanding of the game even from a geometrical perspective. For instance, when for any i the set $\mathcal{X}^i$ is a polyhedron and $f(x^i, x^{-i})$ is a linear function in x^i (given the parameters x^{-i}), each player is solving a (parametrized) linear program. As the vertices of $\mathcal{X}^i$ are geometrically relevant for the simplex method, they are also relevant for the game: no rational player i would adopt a strategy in the interior of its polyhedron $\mathcal{X}^i$ as no optimal solution to the i -th linear program lies in the interior of $\mathcal{X}^i$. The definition of *MPG* is similar to the one of the so-called Nash Equilibrium Problems [63]. However, most of the methods developed in this latter context (i.e., equilibrium programming methods) often assume $\mathcal{X}^i$ has a specific structure, i.e., the players' feasible regions are continuous [64, 130]. In the broad definition of *MPGs*, we do not assume that computing equilibria should require any specific method or structure on any $\mathcal{X}^i$.
- (ii.) *Intersecting objectives with AGT.* We aim to build a language intersecting both elements of game theory and mathematical programming. In this sense, our *MPG* definition is as broad as possible. We aim to extend the theoretical analysis on the efficiency of equilibria performed on standard *AGT* games (i.e., the price of stability and anarchy) to the realm of *MPGs*. Concurrently, we aim at providing theoretical and algorithmic frameworks to select equilibria in *MPGs*. For instance, as we will show later, we reformulate a class of Network Formation Games (i.e., a set of players build a network on a graph by sharing the cost of the edges) as an *MPG*, and subsequently select the game's equilibria with optimization tools.
- (iii.) *Modeling requirements.* *AGT* and mechanism design [118] often change the game's structure or incentives in order to ensure one can efficiently compute equilibria with guaranteed efficiency properties (i.e., prices). However, this may not be the case in applications requiring an explicit optimization problem that captures the application's complexity. Representing the decision-makers' complex set of operational requirements through constraints of *MPGs* increases modelization fidelity: it may be the case that the application's requirements do not allow to simplify the model. Furthermore, the complexity associated with the model itself often translates into equivalently rich insights stemming from the game's equilibria. This is, for instance, the case with what we present in Chapter 6 in the context of an energy-market application, where multi-level hierarchical interactions render a complex model.

Strategies and payoffs. We call each point $x^i \in \mathcal{X}^i$ a *pure strategy* for i . Let $\mathcal{X} = \prod_{i=1}^n \mathcal{X}^i$ be the space of all players' variables: then, we call $x \in \mathcal{X} = (x^1, \dots, x^n)$ a *pure strategy profile* for the game. Let Δ^i be the space of Borel probability measures over $\mathcal{X}^i$, i.e., a space where each player i randomizes over its pure strategies. Then, any σ^i is a *mixed strategy* if $\sigma^i \in \Delta^i$. We define $\text{supp}(\sigma^i) = \{x^i \in \mathcal{X}^i : \sigma^i(x^i) > 0\}$ as the *support* of the mixed strategy σ^i , with $\sigma^i(x^i)$ being the probability of playing x^i in σ^i . Any mixed strategy σ^i with a singleton support $|\text{supp}(\sigma^i)| = 1$ is then a pure strategy for i . Similarly to pure strategies, $\Delta = \prod_{i=1}^n \Delta^i$, and $\sigma \in \Delta$ is a *mixed strategy profile* for the game. The *payoff* of i under the mixed strategy profile σ is

$$f^i(\sigma^i, \sigma^{-i}) = \int_{\mathcal{X}^i} f^i(x^i, x^{-i}) d\sigma. \quad (2.4)$$

In the sequel, we review some of the families of *MPGs* we consider in this thesis. Figure 2.2 provides a visualization of the relationship among such families.

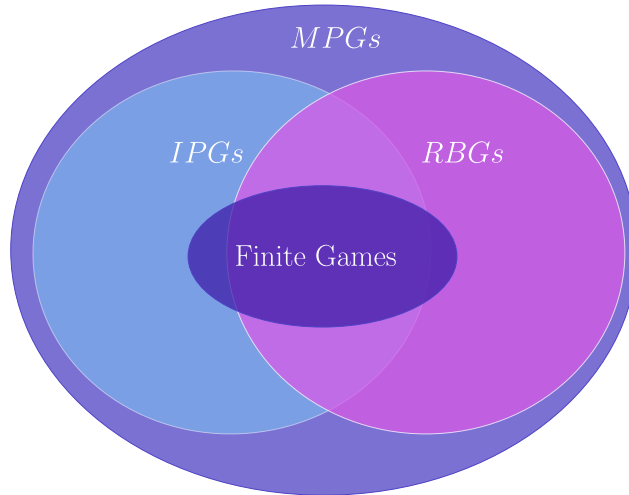


Figure 2.2 The class of *MPGs* and some sub-classes: *IPGs*, finite games, and *RBGs*.

2.2.1 Integer Programming Games

An Integer Programming Game (*IPG*) is an *MPG* where, for each player i , the set $\mathcal{X}^i$ is a *MIP* set defined as

$$\mathcal{X}^i = \{A^i x^i \geq b^i, x_j^i \in \mathbb{Z} \text{ for any } j \in \mathcal{I}^i\}. \quad (2.5)$$

Without loss of generality, we assume the entries of the system of inequalities $A^i x^i \geq b^i$ are integer numbers (i.e., the system of inequalities describes a rational polyhedron) and $\mathcal{I}^i$

contains the indices of the integer-constrained variables. Köppe et al. [95] introduced¹ the *IPGs*, while Carvalho et al. [29] proved the class of *IPGs* include any finite game.

Example 1. Consider the Rock-Paper-Scissors game, where $n = 2$ players simultaneously decide whether to play rock, paper, or scissors. We can reformulate the game as an *IPG* as follows. Each player i has 3 binary variables: $x_R^i = 1$ if i plays rock, $x_P^i = 1$ if it plays paper, and $x_S^i = 1$ if it plays scissors. The payoff is 0 if both players play analogous strategies (i.e., $x_R^1 = x_R^2$), 1 if i wins, and -1 if i loses. Thus, each of the two players solves the parametrized integer program

$$\max_{x^i} \quad -1x_R^i x_P^{-i} + 1x_R^i x_S^{-i} + 0x_R^i x_R^{-i} + 1x_P^i x_R^{-i} - 1x_P^i x_S^{-i} \quad (2.6a)$$

$$+ 0x_P^i x_P^{-i} - 1x_S^i x_R^{-i} + 0x_S^i x_S^{-i} + 1x_S^i x_P^{-i} \quad (2.6b)$$

$$\text{s.t.} \quad x_R^i + x_P^i + x_S^i = 1, x^i \in \{0, 1\}^3. \quad (2.6c)$$

IPGs inherit the modeling capabilities of *MIP* and are expressive modeling tools extending typical *MIP* and Operations Research tasks – such as resource allocation, scheduling, or routing – to a multi-agent setting.

2.2.2 Reciprocally-Bilinear Games

A Reciprocally-Bilinear Game (*RBG*) is an *MPG* where, for each player i , the objective $f^i(x^i, x^{-i})$ takes the form of

$$f^i(x^i, x^{-i}) = (c^i)^\top x^i + (x^{-i})^\top C^i x^i. \quad (2.7)$$

In $f^i(x^i, x^{-i})$, the entries of the vector c^i and the matrix C^i are integer numbers. While *IPGs* constrain the players' feasible sets (their actions), *RBGs* enforce a particular structure on their objectives (their payoffs). In an *IPG*, when each player has an objective conforming to (2.7), the *IPG* is also an *RBG*. Vice versa, when each player in an *RBG* has a feasible set in the form of *MIP* set (as in (2.5)), then the *RBG* is also an *IPG*. *RBGs* force the interaction among players to respect a bilinear function in x^i and x^{-i} . In this thesis, we study *polyhedrally-representable* *RBGs*, namely *RBGs* where, for any player i : (i.) an oracle can optimize a linear function over $\mathcal{X}^i$, and (ii.) $\text{cl conv}(\mathcal{X}^i)$ is a polyhedron. Many classes of well-studied optimization problems are polyhedrally-representable: for instance, linear

¹Some authors deal with Nash Equilibrium Problems with discrete variables, which correspond to *IPGs*. For instance, Sagratella [130]. In the literature of *IPGs*, Nash equilibrium problems with discrete variables are rarely cited.

complementarity problems, some classes of *MIP* problems, linear bilevel programs and reverse convex programs [85, 86].

2.2.3 Stackelberg Games and Bilevel Programming

A Stackelberg competition [53, 139] is a game where two players play sequentially, namely, there are two rounds of decisions. First, the so-called leader (the first player) decides, and afterward, the second player (the follower) decides. Bilevel Programming generalizes the Stackelberg competition so that one or more followers decide after the leader [40].

Mathematical programming formulation. In what follows, we use an apex i to later extend the formulation for multiple leaders. Let w^i be the leaders' variables, and m_i the number of its followers. Each follower $j = 1, \dots, m_i$ controls the variables $y^{i,j}$, and the set of all leader and followers variables is given by $x^i = (w^i, y^i)$, with $y^i = (y^{i,1}, \dots, y^{i,m_i})$. We present the general formulation for the bilevel problem in (2.8), where each leader i solves the optimization problem

$$\max_{w^i} l(x^i) \tag{2.8a}$$

$$\text{s.t. } x^i \in \mathcal{W}^i \tag{2.8b}$$

$$y^{i,j} \in \arg \max_{y^{i,j}} \{f^{i,j}(x^i) : y^{i,j} \in \mathcal{Y}^{i,j}(w^i)\} \quad \forall j = 1, \dots, m_i \tag{2.8c}$$

$$x^i = (w^i, y^{i,1}, \dots, y^{i,m_i}). \tag{2.8d}$$

The leader optimizes a function $l(x^i)$ (2.8a), subject to some constraints $\mathcal{W}^i$ (2.8b); both the leader's objective function and feasible region may contain its followers' variables. Each follower j solves a mathematical program as in (2.8c), where its objective $f^{i,j}(x^i)$ can include its variables as well as the leader's and other followers' ones. Indeed, the followers are playing a simultaneous game among themselves whenever their objective function is parametrized in the variables of other followers, i.e., the j -th objective function includes terms in y^{-j} . Further, each follower's feasible region $\mathcal{Y}^{i,j}(w^i)$ is parametrized through the leader's variables w^i .

For a given leader solution $\bar{w}^i$, there may be multiple optimal solutions for its followers. Whenever the followers pick the optimal solution benefitting the leader the most, the solution is *optimistic*. Otherwise, if the followers damage the leader's payoff as much as they can, the solution is *pessimistic*. In general, bilevel programming has many connections to mathematical programs with equilibrium constraints², i.e., mathematical programs with a system of

²We refer the reader to Chapter 6 for more details on Mathematical Programs with Equilibrium Constraints, bilevel programming, and *MPGs*.

complementarity constraints or variational inequalities [110]. From a modeling perspective, several applications require the bilevel framework to model sequential interactions: to name a few, taxation schemes for green-energy transition [11, 12, 67, 111], price setting problems [21, 99, 100], interdiction problems [25, 26], electricity markets problems [89].

A single level. The definition we provide in (2.8) is rather general and encompasses a wide variety of problems. In some cases, one can reformulate the constraints in (2.8c) so that the bilevel program collapses to a single-level optimization problem. For instance, if the followers are solving convex quadratic programs (i.e., linear programs), then the Karush-Kuhn-Tucker (*KKT*) conditions on the followers' problems (in the form of complementarity constraints) are sufficient to reformulate the bilevel problem into a single level program. Conversely, if both the followers' and the leader's problem contains integer variables, one may not be able to reformulate the problem as a single-level mathematical program of polynomial size [25]. From a complexity perspective, deciding if (2.8) admits a solution is generally Σ_2^P -hard in the polynomial hierarchy of complexity, as proved by [90]. In other words, one could ideally have a Branch-and-Bound framework to solve (2.8), yet, it may need a separation oracle that can simultaneously solve $\mathcal{NP}$ -hard problems [54, 108]. This thesis mainly considers bilevel problems whose lower levels are finitely many convex-quadratic problems, thus bilevel problems admitting a single-level reformulation.

Complex interactions. Besides followers playing a simultaneous game, it may also happen that multiple leaders are playing a simultaneous game among themselves. In this case, each leader's objective function would include the variables associated with other leaders, with i being the index of each leader. This complex system of hierarchical interactions is particularly effective for modeling a series of applications, among which international energy markets or insurance systems. In this thesis, specifically in Chapter 6, we study this type of multi-leader bilevel problem through the concept of Nash games among Stackelberg Players (*NASP*). In this game, several bilevel leaders play a simultaneous game, their followers play a simultaneous game, and bilevel leaders play a sequential game with their respective followers. By reformulating each leader's problem into a single-level, we will reformulate this game as an *RBG*.

2.3 Equilibria, Stability, and Prices

Solving an optimization problem accounts for finding a feasible solution that maximizes (or minimizes) a given objective function, subject to some constraints. However, when dealing with games, we demand solution concepts that take into account the self-interested behavior of each player. A player may be playing its optimal x^i given its opponents' strategies x^{-i} , yet this may not be the case for its opponents. Therefore, an opponent may deviate and pick a strategy whose payoff is more attractive from its perspective. The central question is whether there exist (or not) some *stable* solutions so that no player has an incentive to selfishly defect from them. Arguably, the most famous concept of stable solution is the Nash equilibrium [116, 117]. Nash equilibria are stable solutions precisely because no single player has an incentive to profitably defect from them.

Definition 3. Consider an MPG as in Definition 2. A pure strategy profile $\bar{x} = (\bar{x}^1, \dots, \bar{x}^n)$ is a *Pure Nash Equilibrium* (PNE) if, for any player i and any other strategy $\hat{x}^i \in \mathcal{X}^i$, $f^i(\bar{x}^i, \bar{x}^{-i}) \geq f^i(\hat{x}^i, \bar{x}^{-i})$. Similarly, a mixed strategy profile $\bar{\sigma}$ is a *Mixed Nash Equilibrium* (MNE) if, for any player i and any other mixed strategy $\hat{\sigma}^i \in \Delta^i$, $f^i(\bar{\sigma}^i, \bar{\sigma}^{-i}) \geq f^i(\hat{\sigma}^i, \bar{\sigma}^{-i})$ holds.

In Definition 3, the inequality's RHS represents a possible *deviation* $\hat{x}^i$ or $\hat{\sigma}^i$ from the equilibrium strategy, while the inequality's direction enforces that any deviation cannot improve the payoff of the equilibrium. We remark that whenever the MNE $\bar{\sigma}$ has a singleton support for each player – namely when $|\text{supp}(\bar{\sigma}^i)| = 1$ for any i – the MNE is also a PNE.

Efficiency and the benevolent authority. In numerous contexts, a central and benevolent authority may suggest a strategy to each player in order to maximize a measure of collective welfare. This welfare measure may be, for instance, the *social welfare*, namely the sum of all players' payoffs. Often, the solution that maximizes this welfare measure – the so-called *social optimum* – is not an equilibrium. So, what kind of solution should the authority suggest instead? First, if the authority proposes a collective solution, it must ensure there are little to no incentives for the selfish agents to refuse it. Thus, it should propose an equilibrium so that agents are not incentivized to defer from it. Second, the authority should propose a solution sufficiently close – in terms of “efficiency” or objective value – to the social optimum in order to provide an appealing collective welfare to the players (i.e., all players as a society). The best trade-off between these two objectives is the *best* Nash equilibrium, i.e., the equilibrium that maximizes the welfare measure among all the equilibria (if any).

Existence, prices, and paradigmatic interrogatives. Not all Nash equilibria are created equally, assuming at least one exists. First, establishing if an equilibrium exists may turn out to be a difficult task [49]. Nash proved that there is always an *MNE* equilibrium in finite games. However, this does not apply to any game: for instance, in *IPGs*, deciding if a *PNE* exists is generally Σ_2^p -complete in the polynomial hierarchy [29]. In general, an equilibrium may exist when there are finitely many strategies or when the game satisfies some particular requirements; for instance, if all players solve strictly convex programs, there is a unique equilibrium. Second, in order to provide a measure for equilibria’s efficiency, Koutsoupias and Papadimitriou [96] introduced the concept of *Price of Anarchy (PoA)*, the ratio between the welfare value of the social optimum and the worst-possible (in terms of the welfare’s measure) Nash equilibrium. Symmetrically, Correa et al. [45] and Anshelevich et al. [4] introduced the *Price of Stability (PoS)*, the ratio between the social optimum’s welfare and the one of the best-possible Nash equilibrium. Often, the *AGT* literature focuses on providing theoretical bounds on these prices by exploiting some of the game’s structural properties [4, 5, 35, 118, 129]. However, in practice, determining whether Nash equilibria with certain properties exist, i.e., with a welfare of at least a given threshold, is generally hard [43, 76].

All considered, we could identify four paradigmatic questions regarding Nash equilibria and their properties that often arise in *AGT*:

- (i.) Does at least one Nash equilibrium exist?
- (ii.) How do we compute an equilibrium, if any?
- (iii.) How does this equilibrium compare to the social optimum?
- (iv.) How do we select an equilibrium if more than one exists?

Usually, the answers to such interrogatives are far from being trivial and may depend on the game’s properties. In this thesis, we will address these questions from the perspective of *MPGs*.

CHAPTER 3 LITERATURE REVIEW

We provide a brief literature review on the connecting themes of this dissertation. Each of the four following chapters will complement this review with more specific pointers to the relevant literature.

Nash Equilibrium and Existence. Nash [116, 117] introduced his equilibrium concept and proved the existence of at least an *MNE* in finite games. Glicksberg [78] established the existence of *MNEs* for continuous games, i.e., games where the players' sets of strategies are nonempty compact metric spaces and their payoff functions are continuous. More recently, Stein et al. [140] proved that any *MNE* in a separable game – namely where players' payoffs take a sum-of-products form – has a corresponding payoff-equivalent *MNE* with finite support. In the context of *IPGs*, Carvalho et al. [29, 32] proved there is always an *MNE* whenever all the players' strategy sets are nonempty and bounded, i.e., they are described by a polytope plus some integrality requirements.

Computing Equilibria. A significant stake of the algorithmic approaches related to computing Nash equilibria deals with finite games. Although the literature is vast, we provide a synopsis by categorizing the contributions into two prominent families.

The first family is the one of complementarity methods. Historically, the first contribution in this area is the Lemke-Howson algorithm, a path following method that works for any 2-player finite game [102]. The algorithm has a strong geometrical interpretation: it represents the game by associating a polytope to each player and pivoting among its vertices to find an equilibrium. In the worst case, the algorithm may require a number of exponential pivots in the number of each player's pure strategies. The generalization of Eaves [59] can also solve linear complementarity problems with an exponential worst-case bound [46]. Wilson [146] and Rosenmüller [127] proposed an extension to the Lemke-Howson algorithm to solve n -player games. However, their method requires a series of non-linear equation systems. Von Stengel [144] provides a review on the Lemke-Howson algorithm and other linear methods for finding Nash equilibria in extensive form and bimatrix games with 2 players. Avis et al. [7] proposed two algorithms that exploit geometrical arguments similar to the one of the Lemke-Howson in order to enumerate the Nash equilibria in 2-player games. Audet et al. [6] proposed an algorithm to enumerate Nash equilibria in bimatrix games (2 or 3-players) by solving *MIP* problems associated with the complementarity slackness conditions of each player's (parametrized) linear program. Specifically, such conditions stem from the primal-dual

optimality conditions of the (parametrized) linear programs representing each player’s decision problem.

The second family is the one of support enumeration algorithms. Given a finite game, the main idea is to determine whether an equilibrium with a specific support exists or not. A linear program or an optimization problem can often decide whether the answer to such a question is positive or negative. Sandholm et al. [133] proposed a series of *MIP* formulations to find equilibria with given supports (specifically, in 2-player normal form games) and provided an extension to select equilibria. Porter et al. [124] proposed a search algorithm that proves to be highly efficient in practice. The algorithm prioritizes the search towards equilibria with small and balanced supports.

When each player’s feasible set is large and possibly uncountable, the *MPG* representation becomes crucial as the previous methodologies may turn out to be impractical. Therefore, describing the game through each player’s optimization problem arguably gives the most compact representation. If each player’s problem is convex in its variables, a broadly studied family of algorithms is the one of equilibrium programming methods [63], namely methods where a non-linear complementarity problem can compute an equilibrium. However, such methodologies generally do not work with non-convex problems, for instance, *IPGs*. The only exception is Sagratella [130], that introduced a branching method to enumerate all pure equilibria when payoffs are convex and convex constraints along with integrality requirements on variables model the strategy sets. The approach is general and exploits a branching routine to handle integral non-convexities.

Recently, Carvalho et al. [32] introduced the sample generation method for *IPGs* to compute their equilibria. The algorithm determines if an *MNE* with a given support exists by sampling strategies from each player’s strategy space and attempting to compute an equilibrium with a restricted support. Cronert and Minner [47] provided a version of this algorithm to better select candidate equilibria if more than one exists; however, it only works when players solve pure integer programs.

Complexity. Determining the hardness of computing an equilibrium is crucial from a theoretical perspective, and deeply influences algorithmic design. Daskalakis et al. [49] showed that the task of computing an *MNE* for finite games – specifically, 3-player games – is PPAD-complete. Conitzer and Sandholm [43, 44] proved it is $\mathcal{NP}$ -hard to determine if an equilibrium with certain natural properties – i.e., with a given social welfare – exists and the associated problems are often inapproximable. Carvalho et al. [29, 32] showed that deciding on the existence of an *MNE*, as well as of a *PNE*, in *IPGs* is Σ_2^P -complete in the general

case. For a more detailed exposition and review on the complexity associated with computing equilibria, we refer to Nisan [118, Ch.2].

Efficiency. When multiple equilibria exist, discriminating among them is especially relevant. From a theoretical perspective, Harsanyi and Selten [84] provided an elegant and complete theory of equilibria selection. From a practical perspective, however, equilibria selection often poses significant algorithmic and theoretical challenges by indirectly exposing the task of equilibria enumeration. Even for some 2-player [126] and Avis et al. [7] and bimatrix [6] games, there are considerable computational challenges in designing efficient algorithms for selecting equilibria. This thesis primarily focuses on measuring the efficiency of equilibria by determining their *prices*. Koutsoupias and Papadimitriou [96] introduced the *PoA*, and symmetrically, Anshelevich et al. [4], Correa et al. [45] introduced the *PoS*. Several works in the *AGT* literature focus on providing theoretical bounds for *PoS* and *PoA* by exploiting each game’s structural properties and tend to provide algorithms to compute (approximate) equilibria with given efficiency guarantees [4, 5, 35, 118, 129].

CHAPTER 4 THE ZERO REGRETS ALGORITHM: OPTIMIZING OVER PURE NASH EQUILIBRIA VIA INTEGER PROGRAMMING

Authors: Gabriele Dragotto and Rosario Scatamacchia¹.

Abstract In Algorithmic Game Theory (*AGT*), designing efficient algorithms to compute Nash equilibria poses considerable challenges. We make progress in the field and shed new light on the intersection between Algorithmic Game Theory and Integer Programming. We introduce *ZERO Regrets*, a general cutting plane algorithm to compute, enumerate, and select Pure Nash Equilibria (*PNEs*) in Integer Programming Games, a class of simultaneous and non-cooperative games. We present a theoretical foundation for our algorithmic reasoning and provide a polyhedral characterization of the convex hull of the Pure Nash Equilibria. We introduce the concept of *equilibrium inequality*, and devise an *equilibrium separation oracle* to separate non-equilibrium strategies from *PNEs*. We test *ZERO Regrets* on two paradigmatic classes of games: the Knapsack Game and the Network Formation Game, a well-studied game in *AGT*. Our algorithm successfully solves relevant instances of both games and shows promising applications for equilibria selection.

4.1 Introduction

The concept of Nash equilibrium [116, 117] revolutionized the understanding of the strategic behavior. In many decision-making settings, a selfish agent seeks to optimize its objective function (subject to some constraints) and often interacts with other selfish agents influencing its decisions. The Nash equilibria are *stable* solutions, meaning that no single agent has an incentive to profitably defect from them. However, the quality of equilibria – in terms of a given welfare measure (typically the sum of all agents’ payoffs) – often does not match the quality of the *social optimum*, i.e., the best possible solution for the collectivity. In general, the social optimum is not a stable solution and therefore does not emerge naturally from the agents’ interactions. Nevertheless, in numerous contexts, a central authority may suggest solutions to the agents. On the one hand, if the authority proposes a collective solution to the agents, it should ensure there are little to no incentives to refuse it. On the other hand, the authority should propose a solution sufficiently close – in terms of quality – to the social optimum. The best trade-off between these two objectives is the *best* Nash equilibrium, i.e., a solution that optimizes a welfare measure among the equilibria. Often, the main

¹A pre-print is available in [57].

focus is on selecting a Pure Nash Equilibrium (*PNE*), a stable solution where each agent selects one alternative with unit probability (in contrast to a Mixed Strategy Equilibrium, where agents randomize over the set of their alternatives). Algorithmic Game Theory (*AGT*) studies the intertwining between game theory and algorithms, with emphasis on equilibria's efficiency (quality) [118]. *AGT* attracted significant attention from the computer science and optimization community in the last two decades. Several recent works [32, 82, 95] considered Integer Programming Games (*IPGs*), namely games where the agents solve (parametrized) integer programs. In this work, we study a class of *simultaneous* and *non-cooperative IPGs* among n players (agents) as in Definition 4, where every player has m integer variables.

Definition 4 (*IPG*). *Each player $i = 1, 2, \dots, n$ solves (4.1), where $u^i(x^i, x^{-i})$ – given x^{-i} – is a linear function in x^i with integer coefficients, $A^i \in \mathbb{Z}^{r \times m}$, $b^i \in \mathbb{Z}^r$.*

$$\max_{x^i} \{u^i(x^i, x^{-i}) : x^i \in \mathcal{X}^i\}, \quad \mathcal{X}^i := \{A^i x^i \leq b^i, x^i \in \mathbb{Z}^m\}. \quad (4.1)$$

As standard game theory notation, let x^i denote the vector of variables of player i , and let the operator $(\cdot)^{-i}$ be $(\cdot)$ except i . The vector $x^{-i} = (x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^n)$ represents the variables of i 's *opponents* (all players but i), and the set of linear constraints $A^i x^i \leq b^i$ defines the feasible region $\mathcal{X}^i$ of i . We assume all integer variables are bounded by appropriate linear constraints, and thus that $\mathcal{X}^i$ is finite. In *IPGs*, the strategic interaction takes place in the players' objective functions, and not within their feasible regions. Specifically, players choose their strategy simultaneously, and each player i 's utility $u^i(x^i, x^{-i})$ is a linear function in x^i and parametrized in i 's opponents variables x^{-i} . We assume the entries of A^i and b^i and the coefficients of $u^i(x^i, x^{-i})$ are integers. Further, considering the space of all players' variables $(x^1, \dots, x^n)$, we assume one can always linearize the non-linear terms in each u^i with a finite number of linear inequalities and auxiliary variables (e.g., [137, 143]). Besides, we assume (i.) players have *complete information* about the structure of the game, i.e., each player knows the other players' optimization problems via their feasible regions and objectives, (ii.) each player is *rational*, namely it always selects the best possible strategy given the information available on its opponents, and (iii.) *common knowledge of rationality*, namely each player knows its opponents are rational, and there is complete information. Clearly, if $n = 1$, the game collapses to a single linear integer program. However, for $n > 1$, the problem is an *IPG* [32, 95]. In this work, we focus on optimizing over the set of Pure Nash Equilibria (*PNEs*) for the *IPG* defined above, and on characterizing the polyhedral structure of the set containing the *PNEs*.

Literature. Not all Nash equilibria are created equal. Three paradigmatic questions in *AGT* are often: (i.) Does at least one *PNE* exist? (ii.) How good (or bad) is a *PNE* compared to the social optimum? (iii.) If more than one equilibrium exists, can one select the best *PNE* according to a given measure of quality? Even from an algorithmic perspective, the answers to such questions often require a cumbersome effort. Establishing that a *PNE* does not exist may turn out to be a difficult task [49]. Nash proved that there is always an equilibrium in finite games (i.e., with a finite number of strategies), yet it may be a Mixed Strategy Equilibrium. In *IPGs* where the set of players’ strategies is large, deciding if a *PNE* exists is generally Σ_2^P -hard in the polynomial hierarchy [29]. To measure the efficiency of equilibria, [96] introduced the concept of *Price of Anarchy* (*PoA*), the ratio between the welfare value of the worst-possible equilibrium and the welfare value of the social optimum. Similarly, [4, 45] introduced the *Price of Stability* (*PoS*), the ratio between the welfare value of the best-possible equilibrium and the social optimum’s one. Such definitions hold when agents minimize a cost, e.g., the costs of routing packets in a network. Otherwise, when agents maximize their benefits, we exchange numerator and denominator in the *PoA* (and the *PoS*). In the *AGT* literature, many works focus on providing theoretical bounds for the *PoS* and the *PoA*, often by exploiting the game’s structural properties [4, 5, 35, 118, 129]. Furthermore, equilibria selection indirectly exposes the issue of equilibria enumeration, and from a practical perspective, little is known about enumeration and selection of *PNEs*. Even in some 2-player games (i.e., normal-form [7, 126] and bimatrix [6]) there are considerable computational challenges in designing efficient algorithms for these tasks. In the context of *IPGs*, [32, 33] propose algorithms to compute an equilibrium, not necessarily a *PNE*, without focusing on the selection aspect. Recently, [47] introduced an enumerative procedure (based on [32]), yet, [47] provide results for very small instances (i.e., $n = 2$ with $m = 4$ in the Knapsack game).

Contributions. In this work, we shed new light on the intersection between *AGT* and integer programming. We propose a new theoretical and algorithmic framework to efficiently compute, enumerate, and select *PNEs* for the *IPGs* in Definition 4. We summarize our contributions as follows:

- (i.) From a theoretical perspective, we provide a polyhedral characterization of the convex hull of the *PNEs*. We adapt the concept of valid inequality, closure, and separation oracle to the domain of Nash equilibria. Specifically, we introduce the concept of *equilibrium inequality* to guide the exploration of the set of *PNEs*. With this respect, we provide a general class of equilibrium inequalities and prove – through the concept of *equilibrium closure* – they are sufficient to define the convex hull of the *PNEs*.

- (ii.) From a practical perspective, we design a cutting plane algorithm – *ZERO Regrets* – that computes the best *PNE* for a given welfare measure. This algorithm is flexible and can potentially enumerate *all* the *PNEs* and also compute approximate *PNEs*. The algorithm exploits an *equilibrium separation oracle*, namely a procedure separating non-equilibrium strategies from *PNEs* through the class of inequalities we introduce.
- (iii.) We test our algorithmic framework on two paradigmatic classes of games from the realm of *IPGs* and *AGT*. First, the Knapsack Game, an *IPG* where each player solves a knapsack problem. For this problem, we also provide theoretical results on the hardness of establishing the existence of *PNEs*. Second, the class of Network Formation Games, a well-known and intensely investigated problem in *AGT*, where players build a network over a graph via a cost-sharing mechanism. *ZERO Regrets* proves to be highly efficient in practice and successfully selects *PNEs* on relevant instances of both games.

4.2 Definitions

We assume the reader is familiar with basic concepts of polyhedral theory and integer programming [42]. We introduce further notation and definitions related to an *IPG* instance G , where we omit explicit references to G when not necessary. Let $\mathcal{X}^i$ be the *set of feasible strategies* (or the feasible set) of player i , and let any strategy $\bar{x}^i \in \mathcal{X}^i$ be a (*pure*) *strategy* for i . Any $\bar{x} = (\bar{x}^1, \dots, \bar{x}^n)$ – with $\bar{x}^i \in \mathcal{X}^i$ for any i – is a *strategy profile*. Let vector $x^{-i} = (x^1, \dots, x^{i-1}, x^{i+1}, \dots, x^n)$ denote the vector of *the opponents'* (*pure*) *strategies* of player i . The *payoff* for i under the profile $\bar{x}$ is $u^i(\bar{x}^i, \bar{x}^{-i})$. We define $S(\bar{x}) = \sum_{i=1}^n u^i(\bar{x}^i, \bar{x}^{-i})$ as the *social welfare* corresponding to a given strategy profile $\bar{x}$.

Equilibria and Prices. A strategy $\bar{x}^i$ is a best-response strategy for player i given its opponents' strategies $\bar{x}^{-i}$ if $u^i(\bar{x}^i, \bar{x}^{-i}) \geq u^i(\hat{x}^i, \bar{x}^{-i})$ for any $\hat{x}^i \in \mathcal{X}^i$: equivalently, i cannot profitably deviate to any $\hat{x}^i$ from $\bar{x}^i$. The difference $u^i(\bar{x}^i, \bar{x}^{-i}) - u^i(\hat{x}^i, \bar{x}^{-i})$ is called the *regret* of strategy $\hat{x}^i$ under $\bar{x}^{-i}$. Let $\mathcal{BR}(i, \bar{x}^{-i}) = \{x^i \in \mathcal{X}^i : u^i(x^i, \bar{x}^{-i}) \geq u^i(\hat{x}^i, \bar{x}^{-i}) \forall \hat{x}^i \in \mathcal{X}^i\}$ be the set of best-responses for i under $\bar{x}^{-i}$. A strategy profile $\bar{x}$ is a *PNE* if, for any player i and any strategy $\hat{x}^i \in \mathcal{X}^i$, then $u^i(\bar{x}^i, \bar{x}^{-i}) \geq u^i(\hat{x}^i, \bar{x}^{-i})$, i.e. any $\bar{x}^i$ is a best-response to $\bar{x}^{-i}$ (all regrets are 0). Equivalently, in a *PNE* no player i can unilaterally improve its payoff by deviating from its strategy $\bar{x}^i$. We define the *optimal social welfare* as $OSW = \max_{x^1, \dots, x^n} \{S(x) : x^i \in \mathcal{X}^i \forall i = 1, 2, \dots, n\}$. Given G , we denote as $\mathcal{N} = \{x = (x^1, \dots, x^n) : x \text{ is a } PNE \text{ for } G\}$ the set of its *PNEs*. Also, let $\mathcal{N}^i := \{x^i : (x^i, x^{-i}) \in \mathcal{N}\}$ – with $\mathcal{N}^i \subseteq \mathcal{X}^i$ – be the set of *equilibrium strategies* for i , namely the strategies of i appearing in at least a *PNE*. If $\mathcal{N}$ is not empty, let: (i.) $\dot{x} \in \mathcal{N}$ be so that $S(\dot{x}) \leq S(\bar{x})$ for any $\bar{x} \in \mathcal{N}$ (i.e., the *PNE* with the *worst*

welfare), and (ii.) $\tilde{x} \in \mathcal{N}$ be so that $S(\tilde{x}) \geq S(\bar{x})$ for any $\bar{x} \in \mathcal{N}$ (i.e., the *PNE* with the *best* welfare). Assuming w.l.o.g. $OSW > 0$ and $S(\tilde{x}) > 0$, the *PoA* of G is $\frac{OSW}{S(\tilde{x})}$, and the *PoS* is $\frac{OSW}{S(\tilde{x})}$.

Polyhedral Theory. For a set S , let $\text{conv}(S)$ be its convex hull. Let P be a polyhedron: $\text{bd}(P)$, $\text{ext}(P)$, $\text{rec}(P)$, $\text{int}(P)$, is the boundary, set of its vertices (extreme points), cone of recession directions, and interior, respectively. Let $P \subseteq \mathbb{R}^p$ and $\tilde{x} \notin P$ a point in $\mathbb{R}^p$: a *cut* is a valid inequality $\pi^\top x \leq \pi_0$ for P violated by $\tilde{x}$, i.e., $\pi^\top \tilde{x} > \pi_0$ and $\pi^\top x \leq \pi_0$ for any $x \in P$. Given a point $\hat{x} \in \mathbb{R}^p$ and P , we define the *separation problem* as the task of determining that either (i.) $\hat{x} \in P$, or (ii.) $\hat{x} \notin P$ and returning a cut $\pi^\top x \leq \pi_0$ for P and $\hat{x}$. For each player i , the set $\text{conv}(\mathcal{X}^i)$ is the *perfect formulation* of $\mathcal{X}^i$, namely an integral polyhedron whose vertices are in $\mathcal{X}^i$. Note that, since $u^i(x^i, x^{-i})$ is linear in x^i for any given x^{-i} , the set of player i 's best-responses is in $\text{bd}(\text{conv}(\mathcal{X}^i))$.

4.3 Equilibrium Inequalities

Cutting plane methods are attractive tools for integer programs, both theoretical and applied perspectives. The essential idea is to iteratively refine a relaxation of the original problem by cutting off fractional solutions via valid inequalities for the integer program's perfect formulation. Nevertheless, in an *IPG* where the solution paradigm is the Nash equilibrium, we argue there exist stronger families of cuts, yet, not necessarily *valid* for each player's perfect formulation $\text{conv}(\mathcal{X}^i)$. In fact, for any player i , some of its best-responses in $\text{ext}(\text{conv}(\mathcal{X}^i))$ may never appear in a *PNE*, since no equilibrium strategies of i 's opponents induce i to play such best-responses. Here we introduce a general class of inequalities to characterize the nature of $\text{conv}(\mathcal{N})$. Such inequalities play a pivotal role in the cutting plane algorithm of Section 4.4.

Dominance and Rationality. We ground our reasoning in the concepts of *rationality* and *dominance* [14, 121]. Given two strategies $\bar{x}^i \in \mathcal{X}^i$ and $\hat{x}^i \in \mathcal{X}^i$ for i , $\bar{x}^i$ is strictly *dominated* by $\hat{x}^i$ if, for any choice of opponents strategies x^{-i} , then $u^i(\hat{x}, x^{-i}) > u^i(\bar{x}, x^{-i})$. Then, a *rational* player will never play dominated strategies. This also implies no i would play any strategy in $\text{int}(\text{conv}(\mathcal{X}^i))$. Since dominated strategies – by definition – are never best-responses, they will never be part of any *PNE*. In Example 2, the set $\mathcal{X}^2$ is made of 3 strategies $(x_1^2, x_2^2) = (0, 0), (1, 0), (0, 1)$. Yet, $(x_1^2, x_2^2) = (0, 0)$ is dominated by $(x_1^2, x_2^2) = (0, 1)$, and the latter is dominated by $(x_1^2, x_2^2) = (1, 0)$. However, when considering player 1, we need the assumption of *common knowledge of rationality* to conclude which strategy it will play.

Player 1 needs to know that player 2 would never play $x_2^2 = 1$ to declare $(x_1^1, x_2^1) = (0, 1)$ being dominated by $(x_1^1, x_2^1) = (1, 0)$. When searching for a *PNE* in this example, it follows that $\mathcal{N}^1 = \{(x_1^1, x_2^1) = (1, 0)\}$ and $\mathcal{N}^2 = \{(x_1^2, x_2^2) = (1, 0)\}$. This inductive (and iterative) process of removal of strictly dominated strategies is known as the *iterated elimination of dominated strategies* (*IEDS*). This process produces tighter sets of strategies, and never excludes any *PNE* from the game [142, Ch.4].

Example 2. Consider the IPG where player 1 solves $\max_{x_1} \{6x_1^1 + x_2^1 - 4x_1^1x_1^2 + 3x_2^1x_2^2 : 3x_1^1 + 2x_2^1 \leq 4, x_1^1 \in \{0, 1\}^2\}$, and player 2 solves $\max_{x_2} \{4x_1^2 + 2x_2^2 - x_1^2x_1^1 - x_2^2x_2^1 : 3x_1^2 + 2x_2^2 \leq 4, x_2^2 \in \{0, 1\}^2\}$. The only *PNE* is $(\bar{x}_1^1, \bar{x}_2^1) = (1, 0)$, $(\bar{x}_1^2, \bar{x}_2^2) = (1, 0)$ with a welfare of $S(\bar{x}) = 5$, $u^1(\bar{x}^1, \bar{x}^2) = 2$, and $u^2(\bar{x}^2, \bar{x}^1) = 3$.

In the same fashion of *IEDS*, we propose a family of inequalities that cuts off – from each player’s feasible set – the strategies that never appear in a *PNE*. Thus, from an *IPG* instance G , we aim to derive an instance G' where $\mathcal{N}^i$ replaces each player’s feasible set $\mathcal{X}^i$. Note that, since all $\mathcal{X}^i$ are finite sets, all $\mathcal{N}^i$ are finite as well as the number of *PNEs*.

4.3.1 A Lifted Space

Given the social welfare $S(x)$, we aim to find the *PNE* maximizing it, namely we aim to perform equilibria selection. In this context, the first urgent question is what space should we work in. Since *PNEs* are defined by mutually optimal strategies, a natural choice is to consider a space of all players’ variables x . In our framework, we assume the existence of a higher-dimensional (lifted) space where we linearize the non-linear terms in any $u^i(\cdot)$ via auxiliary variables z and corresponding constraints (e.g., [137, 143]). Although our scheme holds for an arbitrary $f(x) : \prod_{i=1}^n \mathcal{X}^i \rightarrow \mathbb{R}$ we can linearize to $f(x, z)$, we focus on $S(x)$ and the corresponding higher-dimensional $S(x, z)$ defined in the lifted space. In (4.2) we describe this lifted space, where $\mathcal{L}$ is the set of linear constraints necessary to linearize the non-linear terms and includes integer requirements and bounds on the z variables. Any vector $x^1, \dots, x^n, z$ in (4.2) corresponds to a unique strategy profile $x = (x^1, \dots, x^n)$, since x induces z . $\mathcal{K}$ is then a set defined by linear constraints and integer requirements, and thus it is reasonable to deal with $\text{conv}(\mathcal{K})$ and some of its projections. For brevity, let $\text{proj}_x \text{conv}(\mathcal{K}) = \{x = (x^1, \dots, x^n) : \exists z \text{ s.t. } (x^1, \dots, x^n, z) \in \text{conv}(\mathcal{K})\}$, and let $u^i(x^i, x^{-i})$ include the z variables when working in the space of $\text{conv}(\mathcal{K})$.

$$\mathcal{K} = \{(x^1, \dots, x^n, z) \in \mathcal{L}, x^i \in \mathcal{X}^i \text{ for any } i = 1, \dots, n\}. \quad (4.2)$$

Equilibrium Inequalities. The integer points in $\text{proj}_x(\text{conv}(\mathcal{K}))$ encompass all the game's strategy profiles. However, we need to focus on the polyhedron $\mathcal{E} = \{(x^1, \dots, x^n, z) \in \text{conv}(\mathcal{K}) : (x^1, \dots, x^n) \in \text{conv}(\mathcal{N})\}$, since projecting out z yields the convex hull of *PNE* profiles $\text{conv}(\mathcal{N})$. Note that, by definition, $\mathcal{E}$ is a polyhedron (since the definition of $\mathcal{L}$), and $\text{proj}_{x^i}(\mathcal{E}) = \text{conv}(\mathcal{N}^i)$. The role of $\mathcal{E}$ is similar to the one of a perfect formulation for an integer program. As optimizing a linear function over a perfect formulation results in an integer optimum, optimizing a linear function $S(x, z)$ over $\mathcal{E}$ results in a *PNE*. We define $\mathcal{E}$ as the *perfect equilibrium formulation* for G . Also, the equivalent of the integrality gap in integer programming is the *PoS*, namely the ratio between the optimal value of $f(x, z)$ over $\text{conv}(\mathcal{K})$ and $\mathcal{E}$. All considered, we establish the concept of *equilibrium inequality*, a valid inequality for $\mathcal{E}$.

Definition 5 (Equilibrium Inequality). *Consider an IPG instance G . An inequality is an equilibrium inequality for G if it is a valid inequality for $\mathcal{E}$.*

A Class of Equilibrium Inequalities. We introduce a generic class of equilibrium inequalities that are linear in the space of $\text{conv}(\mathcal{K})$. Consider any strategy $\tilde{x}^i \in \mathcal{X}^i$ for i : for any i 's opponents' strategy x^{-i} , $u^i(\tilde{x}^i, x^{-i})$ provides a lower bound on i 's payoff since $\tilde{x}^i \in \mathcal{X}^i$ (i.e., is a feasible point). Then, $u^i(\tilde{x}^i, x^{-i}) \leq u^i(x^i, x^{-i})$ holds for every player i . We introduce such inequalities in Proposition 1.

Proposition 1. *Consider an IPG instance G . For any player i and $\tilde{x}^i \in \mathcal{X}^i$, the inequality $u^i(\tilde{x}^i, x^{-i}) \leq u^i(x^i, x^{-i})$ is an equilibrium inequality.*

Proof. If a point $(\bar{x}, \bar{z}) \in \mathcal{E}$, then $\bar{x} \in \text{conv}(\mathcal{N})$. First, consider the case where $\bar{x} \in \text{ext}(\text{conv}(\mathcal{N}))$, namely $\bar{x} \in \mathcal{N}$ by definition. Assume $(\bar{x}, \bar{z})$ violates the inequality associated with at least a player i , then, $u^i(\tilde{x}^i, \bar{x}^{-i}) > u^i(\bar{x}^i, \bar{x}^{-i})$. Therefore, i can profitably deviate from $\bar{x}^i$ to $\tilde{x}^i$ under $\bar{x}^{-i}$, which contradicts $\bar{x} \in \mathcal{N}$ and $(\bar{x}, \bar{z}) \in \mathcal{E}$. Thus, no point $(\bar{x}, \bar{z}) \in \mathcal{E}$ with $\bar{x} \in \text{ext}(\text{conv}(\mathcal{N}))$ violates the inequality. Since we can represent any point $(\bar{x}, \bar{z}) \in \mathcal{E}$ as a convex combination of the extreme points of $\text{conv}(\mathcal{N})$, the proposition holds by iterating the previous reasoning for each extreme point in the support of $(\bar{x}, \bar{z})$. $\square$

A fundamental issue is whether the inequalities of Proposition 1 are sufficient to define the set $\mathcal{E}$. By modulating the concept of closure introduced by Chvátal [37], we prove this is indeed the case. We define the *equilibrium closure* as the points in $\text{conv}(\mathcal{K})$ satisfying the equilibrium inequalities of Proposition 1.

Theorem 1. Consider an IPG instance G where $|\mathcal{N}| \neq 0$. Let the equilibrium closure of $\text{conv}(\mathcal{K})$ for the set of equilibrium inequalities in Proposition 1 be

$$P^e := \left\{ (x, z) \in \text{conv}(\mathcal{K}) \left| \begin{array}{l} u^i(\tilde{x}^i, x^{-i}) \leq u^i(x^i, x^{-i}) \\ \forall \tilde{x} : \tilde{x}^i \in \mathcal{BR}(i, \tilde{x}^{-i}), i = 1, \dots, n \end{array} \right. \right\},$$

where the equilibrium inequalities consider only the best-responses $\tilde{x}^i$ for any player i . Then, (i.) P^e is a rational polyhedron, (ii.) $\text{int}(P^e)$ contains no points $(\bar{x}, \bar{z}) : \bar{x} \in \mathbb{Z}^{nm}$, (iii.) $P^e = \mathcal{E}$.

Proof. Proof of (i.) Since $\mathcal{X}^i$ is finite for any i , the number of best-responses (and correspondingly, of equilibrium inequalities) is finite. Also, $\text{conv}(\mathcal{K})$ is a rational polyhedron, and any equilibrium inequality has integer coefficients. It follows that P^e is a rational polyhedron.

Proof of (ii.) Assume there exists a $(\bar{x}, \bar{z}) \in \text{int}(P^e)$ so that $\bar{x} \in \mathbb{Z}^{nm}$. Then, $\bar{x} \in \mathcal{N}$ by definition of Nash equilibrium. However, since $\bar{x} \in \text{int}(P^e)$, then no equilibrium inequality is tight, contradicting the fact $\bar{x}$ is a PNE . This implies all PNE s lie on the boundary of P^e .

Proof of (iii.) The inequalities defining P^e are equilibrium inequalities, then any $(\bar{x}, \bar{z}) \in \mathcal{E}$ belongs to P^e , implying $\mathcal{E} \subseteq P^e$. However, since all PNE s are on the boundary of P^e , $P^e \subseteq \text{conv}(\mathcal{K})$ and $\mathcal{E} \subseteq \text{conv}(\mathcal{K})$, we necessarily must have $P^e = \mathcal{E}$. $\square$

4.4 The Cutting Plane Algorithm and its Oracle

If an oracle gives us $\mathcal{E}$ through a set of linear inequalities, then an optimal solution to $\max_{x^1, \dots, x^n, z} \{f(x, z) : (x, z) \in \mathcal{E}\}$ (i.e., a linear program) that is also an extreme point of $\mathcal{E}$ is a PNE for G for any linear function $f(x, z)$ (thus for $S(x, z)$). However, there are two major issues. First, $\mathcal{E} \subseteq \text{conv}(\mathcal{K})$, and $\text{conv}(\mathcal{K})$ is a perfect formulation described by a possibly large number of inequalities. Second, even if an oracle provides $\text{conv}(\mathcal{K})$, retrieving $\mathcal{E}$ through Theorem 1 may still require a large number of inequalities. In practice, we actually do not need $\mathcal{E}$ nor $\text{conv}(\mathcal{K})$: a more reasonable goal is to get a polyhedron containing $\text{conv}(\mathcal{K})$ over which we can optimize $f(x, z)$ efficiently (i.e., with a linear program) and obtain an integer solution that is also a PNE . The first challenge is to obtain an integer solution: yet, we could deploy known families of integer programming cutting planes (they are also equilibrium inequalities as they are valid for $\mathcal{E}$) and branching schemes. Equivalently, we exploit a Mixed-Integer Programming (*MIP*) solver to solve $\max_{x^1, \dots, x^n, z} \{f(x, z) : (x, z) \in K\}$. Once the maximizer fulfills the integrality requirements, we are done if it is also a PNE . Otherwise, the second challenge is to cut off such maximizer, since it is not a PNE , by separating at least an equilibrium inequality from Proposition 1.

Equilibrium Separation Oracle. Given an integer point $(\tilde{x}, \tilde{z}) \in \text{conv}(\mathcal{K})$ (i.e., the point returned by the *MIP* solver), the central question is to decide if $\tilde{x} \in \mathcal{N}$, and, if not, to derive an equilibrium inequality to cut off $(\tilde{x}, \tilde{z})$. If we use the equilibrium inequalities from Proposition 1, the process terminates in a finite number of iterations, since Theorem 1. In the spirit of [81, 91], we define a separation oracle for the equilibrium inequalities and $\mathcal{E}$. The *equilibrium separation oracle* solves the *equilibrium separation problem* of Definition 6.

Definition 6 (Equilibrium Separation Problem). *Consider an IPG instance G . Given a point $(\bar{x}, \bar{z})$, the equilibrium separation problem is the task of determining that either: (i.) $(\bar{x}, \bar{z}) \in \mathcal{E}$, or (ii.) $(\bar{x}, \bar{z}) \notin \mathcal{E}$ and return an equilibrium inequality violated by $(\bar{x}, \bar{z})$.*

Algorithm 1 presents our separation oracle for the inequalities of Proposition 1. Given $(\bar{x}, \bar{z})$ and an empty set of linear inequalities ϕ , the algorithm outputs either **yes** if $(\bar{x}, \bar{z}) \in \mathcal{E}$, or **no** and adds in ϕ some inequalities violated by $(\bar{x}, \bar{z})$. The algorithm separates at most one inequality for any player i . Note that $\bar{x}^i$ should be a best-response to be in a *PNE*. Then, we solve $\max_{x^i} \{u^i(x^i, \bar{x}^{-i}) : A^i x^i \leq b^i, x^i \in \mathbb{Z}^m\}$, where $\hat{x}^i$ is one of its maximizers. If $u^i(\bar{x}^i, \bar{x}^{-i}) = u^i(\hat{x}^i, \bar{x}^{-i})$, then $\bar{x}^i$ is also a best-response. However, if $u^i(\hat{x}^i, \bar{x}^{-i}) > u^i(\bar{x}^i, \bar{x}^{-i})$, the algorithm adds to ϕ an equilibrium inequality $u^i(\hat{x}^i, x^{-i}) \leq u^i(x^i, x^{-i})$ violated by $(\bar{x}, \bar{z})$. After considering all players, if $|\phi| = 0$, then $\bar{x}$ is by definition a *PNE* and the answer is **yes**. Otherwise, the algorithm returns **no** and $\phi \neq \emptyset$, i.e., at least an equilibrium inequality cutting off $(\bar{x}, \bar{z})$.

Algorithm 1: Equilibrium Separation Oracle

Data: An IPG instance G , a point $(\bar{x}, \bar{z})$, and a set of cuts $\phi = \emptyset$.

Result: Either: (i.) **yes** if $(\bar{x}, \bar{z}) \in \mathcal{E}$, or (ii.) **no** and ϕ .

```

1 for  $i \leftarrow 1$  to  $n$  do
2    $\hat{x}^i \leftarrow \max_{x^i} \{u^i(x^i, \bar{x}^{-i}) : A^i x^i \leq b^i, x^i \in \mathbb{Z}^m\}$  ;
3   if  $u^i(\hat{x}^i, \bar{x}^{-i}) > u^i(\bar{x}^i, \bar{x}^{-i})$  then
4      $\text{Add } u^i(\hat{x}^i, x^{-i}) \leq u^i(x^i, x^{-i})$  to  $\phi$ ;
5 if  $|\phi| = 0$  then return yes ;
6 else return no and } \phi ;
```

ZERO Regrets. We present our cutting plane algorithm *ZERO Regrets* in Algorithm 2. The inputs are an instance G , and (a linearizable) $f(x)$, while the output is either the *PNE* $\bar{x}$ maximizing $f(x)$, or a certificate that no *PNE* exists. Let Φ be a set of equilibrium inequalities, and $\mathcal{Q} = \max_{x^1, \dots, x^n, z} \{f(x, z) : (x, z) \in \mathcal{K}, (x, z) \text{ s.t. } \Phi\}$. We assume $\mathcal{Q}$ is feasible and bounded (otherwise, there is no point in getting a *PNE* with an arbitrarily bad welfare). At each

iteration, we compute an optimal solution $(\bar{x}, \bar{z})$ of $\mathcal{Q}$. Then, the equilibrium separation oracle (Algorithm 1) evaluates such solution: if the oracle returns **yes**, then $\bar{x} = \bar{x}$ is the *PNE* maximizing $f(x)$ in G . Otherwise, the oracle returns a set ϕ of equilibrium inequalities cutting off $(\bar{x}, \bar{z})$, and the algorithm adds ϕ to Φ . Therefore, the process restarts by solving $\mathcal{Q}$ with the additional set of constraints. If at any iteration $\mathcal{Q}$ becomes infeasible, then G has no *PNE*. Theorem 1 implies both correctness and finite termination of Algorithm 2.

Algorithm 2: ZERO Regrets

Data: An *IPG* instance G , and a function $f(x)$.

Result: Either: (i.) the *PNE* $\bar{x}$ maximizing $f(x)$, or (ii.) **no PNE**

```

1  $\Phi = \{0 \leq 1\}$ , and  $\mathcal{Q} = \max_{x^1, \dots, x^n, z} \{f(x, z) : (x, z) \in \mathcal{K}, (x, z) \text{ s.t. } \Phi\}$ ;
2 while true do
3   if  $\mathcal{Q}$  is infeasible then return no PNE ;
4    $(\bar{x}, \bar{z}) = \arg \max \mathcal{Q}$ ;  $\phi = \emptyset$  ;
5   if EquilibriumSeparationOracle( $G, (\bar{x}, \bar{z}), \phi$ ) is yes then
6     return  $\bar{x} = \bar{x}$ ;
7   else add  $\phi$  to  $\Phi$  ;
```

In Example 3, we provide a toy example to show the rationale of *ZERO Regrets*.

Example 3. Consider the Knapsack Game instance of Example 1 where player 1 solves $\max_{x^1} \{6x_1^1 + x_2^1 - 4x_1^1x_2^1 + 3x_2^1x_2^2 : 3x_1^1 + 2x_2^1 \leq 4, x^1 \in \{0, 1\}^2\}$, and player 2 solves $\max_{x^2} \{4x_1^2 + 2x_2^2 - x_1^2x_1^1 - x_2^2x_2^1 : 3x_1^2 + 2x_2^2 \leq 4, x^2 \in \{0, 1\}^2\}$. To linearize the players' utility functions, a viable option is to introduce the binary variables z_i^1, z_i^2, z_i^{12} equal to 1 if only Player 1 selects item i , only Player 2 selects item i or both players select item i , respectively. For $i = 1, 2$, the variables x_i^1 and x_i^2 are linked with the additional z variables by the constraints $x_i^1 = z_i^1 + z_i^{12}$, $x_i^2 = z_i^2 + z_i^{12}$, with $z_i^1 + z_i^2 + z_i^{12} \leq 1$. Thus, player 1's utility function is $6x_1^1 + x_2^1 - 4z_1^{12} + 3z_2^{12}$ and player 2's utility function is $4x_1^2 + 2x_2^2 - z_1^{12} - z_2^{12}$. Correspondingly, problem $\mathcal{Q}$ maximizing the social welfare is

$$\begin{aligned}
\max \quad & 6x_1^1 + x_2^1 + 4x_1^2 + 2x_2^2 - 5z_1^{12} + 2z_2^{12} \\
& 3x_1^1 + 2x_2^1 \leq 4, \quad 3x_1^2 + 2x_2^2 \leq 4 \\
& x_i^1 = z_i^1 + z_i^{12}, \quad x_i^2 = z_i^2 + z_i^{12}, \quad z_i^1 + z_i^2 + z_i^{12} \leq 1 \quad i = 1, 2. \\
& x_i^1, x_i^2, z_i^1, z_i^2, z_i^{12} \in \{0, 1\} \quad i = 1, 2.
\end{aligned}$$

An optimal solution of the problem is $(\bar{x}_1^1, \bar{x}_2^1) = (1, 0)$, $(\bar{x}_1^2, \bar{x}_2^2) = (0, 1)$ (we do not report the z values for conciseness). The social welfare is 8 and the utility values are 6 and 2, respectively. However, this solution is not a *PNE*. In fact, while the best-response to $\bar{x}^2$ for player 1 is $\bar{x}^1$,

the best-response to $\bar{x}^1$ for player 2 is $(\hat{x}_1^2, \hat{x}_2^2) = (1, 0)$ with an utility value of 3. Therefore, from player 2, we derive the equilibrium inequality

$$4 - x_1^1 \leq 4x_1^2 + 2x_2^2 - z_1^{12} - z_2^{12}$$

cutting off $\bar{x}$. The left-hand side of the inequality represents $u^2(\hat{x}^2, x^1)$, namely player 2's utility function $4x_1^2 + 2x_2^2 - x_1^2x_1^1 - x_2^2x_2^1$ with $x^2 = \hat{x}^2$. The right-hand side of the inequality represents $u^2(x^2, x^1)$, namely player 2's utility function linearized in the lifted space. By adding the equilibrium inequality to $\mathcal{Q}$, the optimal solution is then $(\bar{x}_1^1, \bar{x}_2^1) = (1, 0)$, $(\bar{x}_1^2, \bar{x}_2^2) = (1, 0)$ with utility values 2 and 3 and a welfare of 5. Since $\bar{x}$ is a PNE, the algorithm terminates.

Game-theoretical Interpretation. We provide a straightforward game-theoretical interpretation of *ZERO Regrets*. The algorithm acts as a central authority (i.e., a central planner) when optimizing $f(x, z)$ over $\mathcal{K}$, meaning that it produces a solution that optimizes the welfare. Afterward, it proposes the solution to each player, who evaluates it through the equilibrium separation oracle. The latter acts as a *rationality* blackbox, in the sense that it advises each player i whether the proposed strategy is acceptable or not. In other words, the rationality blackbox tells the player i if it should selfishly (and rationally) deviate to a better strategy, ignoring the central authority advice. On the one hand, if the rationality blackbox says the solution is acceptable for player i , then the player knows (through the oracle) it should accept the proposed strategy. On the other hand, if at least one player i refuses the proposed solution, the central authority should exclude such a solution and formulate a new proposal. Namely, it should cut off the non-equilibrium strategy and compute a new solution.

Some Remarks. We conclude with some further considerations on *ZERO Regrets*. First, it is sufficient to add just one equilibrium inequality in ϕ to cut off a given solution $(\bar{x}, \bar{z})$. However, we expect a good trade-off between $|\phi| = 1$ and $|\phi| = n$ may speed up the algorithm's convergence. Second, we can modify Algorithm 2 to enumerate all PNEs in $\mathcal{N}$ as follows. In Step 6, instead of terminating and returning $\bar{x}$, we memorize $\bar{x}$ and add an (invalid) inequality cutting off $(\bar{x}, \bar{z})$ from $\mathcal{E}$. Since all x are integers, such inequality can be, for instance, a classical hamming distance from $\bar{x}$. The algorithm will eventually cut off any PNE until $\mathcal{Q}$ becomes infeasible. Third, *ZERO Regrets* can compute approximate PNEs, i.e., when each player can deviate at most by a small ϵ with respect to a best-response [118]. Approximate PNEs may be a reasonable compromise in games where no PNE exists. W.l.o.g, if ϵ is integer, we can adapt Algorithm 1 to separate ϵ -equilibrium inequalities $u^i(\hat{x}^i, x^{-i}) - \epsilon \leq u^i(x^i, x^{-i})$, without affecting the correctness of Algorithm 2 (yet returning an ϵ -equilibrium).

4.5 Knapsack and Network Formation Games

We evaluate *ZERO Regrets* for the task of selecting equilibria with maximum welfare on two well-known classes of games. Specifically, we consider the Knapsack Game [27, 32, 33] – for which we also provide further theoretical results – and the Network Formation Game [5, 35].

4.5.1 Knapsack Game

The Knapsack Game (*KPG*) is an *IPG* among n players, where each i solves the a binary knapsack problem [92] with m items as

$$\max_{x^i} \left\{ \sum_{j=1}^m p_j^i x_j^i + \sum_{k=1, k \neq i}^n \sum_{j=1}^m C_{k,j}^i x_j^i x_j^k : \sum_{j=1}^m w_j^i x_j^i \leq b^i, \mathbf{x}^i \in \{0, 1\}^m \right\}. \quad (4.3)$$

As in the classical knapsack problem, we assume that the profits p_j^i , weights w_j^i and capacities b^i are in $\mathbb{Z}_0^+$. The selection of an item j by a player $k \neq i$ impacts either negatively or positively the item profit for player i through integer coefficients $C_{k,j}^i$. Clearly, given the strategies of the other players x^{-i} , computing a corresponding best-response for player i is $\mathcal{NP}$ -hard. [27] introduced the game with $n = 2$ and $p_j^i = 0 \ \forall j = 1, \dots, m, i = 1, 2$. [32, 33] consider a more general game variant (p_j^i and w_j^i in $\mathbb{Z}$) and provided algorithms to compute mixed-strategy Nash equilibria, yet the focus was not on *PNEs* nor on equilibria selection. We can straightforwardly apply our algorithmic framework to the *KPG* in (4.3), since we can linearize the bilinear products $x_j^i x_j^k$ (for any i, k, j) with $\mathcal{O}(mn^2)$ auxiliary variables and additional constraints. We claim the *KPG* can be extremely difficult to solve even with two players: in Theorem 2, we prove that deciding if a *KPG* instance has a *PNE* – even with $n = 2$ – is Σ_2^P -complete in the polynomial hierarchy, matching the result of [32] for general *IPGs*. Also, we show that when at least one *PNE* exists, the *PoS* and *PoA* can be arbitrarily bad.

Theorem 2. *Deciding if a KPG instance has a PNE is a Σ_2^P -complete problem.*

We perform a reduction from the *DeNegre Bilevel Knapsack Problem (BKP)* below, which is Σ_2^P -complete [25].

Definition 7 (BKP). *Given two m -dimensional non-negative integer vectors a and b and two non-negative integers A and B , the BKP asks whether there exists a binary vector x – with $\sum_{j=1}^m a_j x_j \leq A$ – satisfying $\sum_{j=1}^m b_j y_j (1 - x_j) \leq B - 1$ for any binary vector y such that $\sum_{j=1}^m b_j y_j \leq B$.*

Without loss of generality, we assume $a_j \leq A$ for any j . If this is not the case, we can always modify the original *BKP* instance as follows: (i.) we replace A with $2A + 1$, any $a_j \leq A$

with $2a_j$, and any $a_j > A$ with $(2A + 1)$, and (ii.) we add a new element $m + 1$ (i.e., new item), with $a_{m+1} = 1$ and $b_{m+1} = B$. In any solution of this modified instance, we must have $x_{m+1} = 1$, otherwise $\sum_{j=1}^{m+1} b_j y_j (1 - x_j) \leq B - 1$ would never hold since $\sum_{j=1}^{m+1} b_j y_j (1 - x_j) = B$ when $x_{m+1} = 0$ and $y_{m+1} = 1$. Setting $x_{m+1} = 1$ gives a residual capacity $2A$ for the packing constraint of x . Indeed, every subset of x variables with original $a_j \leq A$ that was satisfying $\sum_{j=1}^m a_j x_j \leq A$ now satisfies $\sum_{j=1}^m 2a_j x_j \leq 2A$. On the contrary, we cannot select any x_j variable with original $a_j > A$. Thus, a solution (if any) to the original instance corresponds to a solution to the modified instance, and vice versa.

Proof. First, note that deciding if KPG admits a PNE is in Σ_2^P , as we ask whether there is a strategy profile where every player cannot improve its payoff with any of its strategies, and we can compute the payoff of such strategies in polynomial time. Given a BKP instance, we construct a KPG instance with 2 players as follows. We consider $m + 1$ items and associate the elements of vectors x and y with the first m elements of vectors x^1 and x^2 , respectively. Player 1 solves the problem in (4.4), whereas player 2 solves the problem in (4.5).

$$\max_{x^1} \left\{ \sum_{j=1}^m b_j x_j^1 x_j^2 + x_{m+1}^1 x_{m+1}^2 : \sum_{j=1}^m a_j x_j^1 \leq A, x^1 \in \{0, 1\}^{m+1} \right\} \quad (4.4)$$

$$\begin{aligned} \max_{x^2} \{ & (B - 1)x_{m+1}^2 + \sum_{j=1}^m b_j x_j^2 - \sum_{j=1}^m b_j x_j^2 x_j^1 : \\ & \sum_{j=1}^m b_j x_j^2 + Bx_{m+1}^2 \leq B, x^2 \in \{0, 1\}^{m+1} \} \end{aligned} \quad (4.5)$$

We need to show that the KPG instance has a PNE if and only if the corresponding BKP instance has a solution.

BKP has a solution. We assume the BKP instance has a solution $\bar{x}$. We prove that $\hat{x}^1 = (\bar{x}, 1)$, $\hat{x}^2 = (\bar{0}, 1)$ (with $\bar{0}$ being an m -dimensional vector of zeros) is a PNE . First, both the strategies $\hat{x}^1$ and $\hat{x}^2$ are feasible by construction. Given $\hat{x}^2$, player 1 attains the maximum payoff of 1 by playing strategy $\hat{x}^1$. The strategy $\hat{x}^2$ yields a payoff of $B - 1$ for player 2 when player 1 plays $\hat{x}^1$. Player 2 cannot profitably deviate by setting $x_{m+1}^2 = 0$. This follows from the fact that the BKP instance has a solution $\bar{x}$ and, given that $\hat{x}_j^1 = \bar{x}_j$ for $j = 1, \dots, m$,

the following inequality must hold

$$\sum_{j=1}^m b_j x_j^2 - \sum_{j=1}^m b_j x_j^2 \hat{x}_j^1 \leq B - 1.$$

Thus, the pair of strategies $(\hat{x}^1, \hat{x}^2)$ is also a *PNE* for the *KPG* instance.

BKP has not a solution. If the *BKP* instance has not a solution, player 2 never plays $x_{m+1}^2 = 1$ in a best-response, as it can always obtain a payoff of B with variables $x_1^2, \dots, x_m^2$ for any player 1's feasible strategy. Consider any player 2's best-response $\hat{x}^2$, with $\hat{x}_{m+1}^2 = 0$, and assume the *KPG* instance has a *PNE* $(\hat{x}^1, \hat{x}^2)$. Then, in the player 1's best-response $\hat{x}^1$, there exists at least one $\hat{x}_j^1 = 1$ when $\hat{x}_j^2 = 1$ and $b_j > 0$ (since $a_j \leq A$ for any j). However, in this case, player 2 would deviate from $\hat{x}^2$, since $\hat{x}^2$ gives a payoff $< B$ under $\hat{x}^1$. Thus, no *PNE* exists in the *KPG* instance. $\square$

Proposition 2. *The PoA and the PoS in KPG can be arbitrarily bad.*

Proof. Consider the following *KPG* instance with $n = 2$: $i = 1$ solves the problem $\max_{x^1} \{Mx_1^1 + x_2^1 - (M - 2)x_1^1 x_1^2 - x_2^1 x_2^2 : 3x_1^1 + 2x_2^1 \leq 4, x^1 \in \{0, 1\}^2\}$ where M is an arbitrarily large value; $i = 2$ solves $\max_{x^2} \{4x_1^2 + x_2^2 - x_1^2 x_1^1 - x_2^2 x_2^1 : 3x_1^2 + 2x_2^2 \leq 4, x^2 \in \{0, 1\}^2\}$. The only *PNE* is $(\bar{x}_1^1, \bar{x}_2^1, \bar{x}_1^2, \bar{x}_2^2) = (1, 0, 1, 0)$, with $u^1(\bar{x}^1, \bar{x}^2) = 2$, $u^2(\bar{x}^2, \bar{x}^1) = 3$, $S(\bar{x}) = 5$. The maximum welfare $OSW = M + 1$ is given by $(\hat{x}_1^1, \hat{x}_2^1, \hat{x}_1^2, \hat{x}_2^2) = (1, 0, 0, 1)$, i.e. *OSW* is arbitrarily large and there are no bounds on both the *PoA* and the *PoS*. $\square$

4.5.2 Network Formation Game

Network design games are paradigmatic problems in Algorithmic Game Theory [5, 35, 118]. We consider a Network Formation Game (*NFG*) where n players are interested in building a network. Let $G(V, E)$ be a directed graph representing a network layout, where V , E are the sets of vertices and edges, respectively. Each edge $(h, l) \in E$ has a construction cost $c_{hl} \in \mathbb{Z}^+$, and each player i wants to connect an origin s^i with a destination t^i while minimizing its construction costs. A cost-sharing mechanism determines the cost of each edge $c_{hl}^i(x)$ for i as a function of the number of players crossing (h, l) . A commonly adopted mechanism is the *Shapley cost-sharing mechanism*, where players using (h, l) equally share its cost c_{hl} . The goal is to find a *PNE* (if any) minimizing the sum of construction costs for each player. Although the *NFG* with *Shapley cost-sharing mechanism* is a potential game (i.e., best-response dynamics always converge, and there is always a *PNE*), selecting the best *PNE* is an $\mathcal{NP}$ -hard problem [5]. We model the *NFG* as an *IPG* as follows. For any player

i and edge (h, l) , let the binary variables x_{hl}^i be 1 if i uses the edge. We use classical flow constraints modeling a path between s^i and t^i . For conciseness, we represent these constraints and binary requirements with a set $\mathcal{F}^i$ for each i . Thus, each i solves

$$\min_{x^i} \left\{ \sum_{(h,l) \in E} \frac{c_{hl} x_{hl}^i}{\sum_{k=1}^n x_{hl}^k} : x^i \in \mathcal{F}^i \right\}. \quad (4.6)$$

In this formulation, we assume that if $\sum_{k=1}^n x_{hl}^k = 0$, then the fraction is equal to 0. For any i , the cost contribution of each edge (h, l) to the objective is not linear in x , yet we can linearize it. For instance, consider a game with $n = 3$ and the objective of $i = 1$. Let the binary variable $z_{hl}^{j,\dots,k}$ be 1 if only players $j, \dots, k$ select the edge (h, l) . Then, $x_{hl}^1 = z_{hl}^1 + z_{hl}^{12} + z_{hl}^{13} + z_{hl}^{123}$, $x_{hl}^2 = z_{hl}^2 + z_{hl}^{12} + z_{hl}^{23} + z_{hl}^{123}$, $x_{hl}^3 = z_{hl}^3 + z_{hl}^{13} + z_{hl}^{23} + z_{hl}^{123}$, and the clique $z_{hl}^1 + z_{hl}^2 + z_{hl}^3 + z_{hl}^{12} + z_{hl}^{13} + z_{hl}^{23} + z_{hl}^{123} \leq 1$. The term for edge (h, l) in the objective of $i = 1$ is then $c_{hl} z_{hl}^1 + \frac{c_{hl}}{2}(z_{hl}^{12} + z_{hl}^{13}) + \frac{c_{hl}}{3} z_{hl}^{123}$. In our tests, we also model the general weighted *NFG* [35], where each i has a weight w^i , and the cost share of each selected (h, l) is $w^i c_{hl}$ divided by the weights of all players using (h, l) . Specifically, we consider the 3-player weighted *NFG*, where a *PNE* may not exist [5, 35].

4.6 Computational Tests

We performed our tests on an *Intel Xeon Gold 6142*, with 128GB of RAM and with *Gurobi* 9.2 as *MIP* solver for Algorithm 2 and to compute *OSW*. The time-limit for *ZERO Regrets* is 1800 seconds.

Knapsack Game. We generate *KPG* instances with $n = 2, 3$ and $m = 25, 50, 75, 100$. In any instance, for any i , p_j^i and w_j^i are random integers uniformly distributed in $[1, 100]$. We consider three values for the knapsack capacity b^i equal to $0.2 \sum_{j=1}^m w_j^i$, $0.5 \sum_{j=1}^m w_j^i$, $0.8 \sum_{j=1}^m w_j^i$, respectively. We consider three different distributions for the integer interaction coefficients $C_{k,j}^i$. For any i , they can be: *a*) equal and uniformly distributed in $[1, 100]$, or *b*) random and uniformly distributed in $[1, 100]$, or *c*) random and uniformly distributed in $[-100, 100]$. In Table 4.1, we present the results for the 72 instances. For any given number of players n , items m and distribution of coefficients $C_{k,j}^i$ ($((n, m, d))$), we report the performance over 3 instances with different capacity distributions, in terms of average number of equilibrium inequalities added ($\#EI$), average computational time (*Time*), average *PoS* (*PoS*) when we find the best *PNE*, number of time-limit hits (*Tl*). The averages $\#EI$ and *Time* consider also the instances where we hit the time-limit. *ZERO Regrets* solves almost all instances with $n = 2$, especially with distribution *a*. Both running times and number of

equilibrium inequalities are generally limited. The PoS is generally low, and increases with distribution c due to the complex interactions stemming from negative $C_{k,j}^i$. We remind that a PoS close to 1 does not mean the instance is “easy”. On the contrary, a $PoS \approx 1$ highlights the existence of a high-quality PNE , with a welfare close to OSW . Thus, this result also provides further evidence on the need of selecting such a PNE . With $n = 3$, *ZERO Regrets* performs well when $m < 75$, and solves larger instances with distribution a . Previous works consider up to $m = 40$ items with $n = 3$, and does not perform equilibria selection with $m > 4$.

Network Formation Game. We consider the NFG with $n = 3$ on grid-based (directed) graphs $G(V, E)$, where each i has to cross the grid from left to right to reach its destinations. Compared to a standard grid graph, we randomly add some edges between adjacent layers to increase the number of paths. The instances are so that $|V| \in [50, 500]$, and the costs c_{hl} for each edge (h, l) are random integers uniformly distributed in $[20, 100]$. We consider three distributions of player’s weights: (i.) the Shapely-mechanism, with $w^1 = w^2 = w^3 = 1$, or (ii.) $w^1 = 0.6$, $w^2 = 0.2$, and $w^3 = 0.2$, or (iii.) $w^1 = 0.45$, $w^2 = 0.45$, and $w^3 = 0.1$. Table 4.2 reports the results, where we average over the distributions of the players’ weights. For each graph, the table reports the graph size $(|V|, |E|)$, whereas the other columns have the same meaning of the ones of Table 4.1. Similarly to the KPG , we effectively solve all the instances but 3. Generally, the literature does not consider this problem from a practical perspective but provides theoretical bounds on the PoS and PoA . Nevertheless, we can compute high-quality PNE s even in large-size graphs (i.e., $PoS \approx 1$), with a limited number of equilibrium inequalities and modest running times.

Table 4.1 Results for KPG .

(n, m, d)	#EI	Time	PoS	Tl	(n, m, d)	#EI	Time	PoS	Tl
(2, 25, a)	10.67	0.08	1.04	0	(3, 25, a)	17.33	0.79	1.01	0
(2, 25, b)	15.67	0.17	1.02	0	(3, 25, b)	29.67	1.36	1.02	0
(2, 25, c)	40.00	1.52	1.06	0	(3, 25, c)	157.33	640.02	1.26	1
(2, 50, a)	15.00	0.22	1.02	0	(3, 50, a)	67.00	115.06	1.02	0
(2, 50, b)	41.67	1.27	1.01	0	(3, 50, b)	182.00	627.30	1.01	1
(2, 50, c)	112.00	30.75	1.08	0	(3, 50, c)	193.67	1800.00	-	3
(2, 75, a)	45.33	2.55	1.00	0	(3, 75, a)	156.33	1267.78	1.01	2
(2, 75, b)	146.33	94.03	1.02	0	(3, 75, b)	297.33	1800.00	-	3
(2, 75, c)	242.67	636.72	1.07	1	(3, 75, c)	179.00	1800.00	-	3
(2, 100, a)	37.00	2.24	1.01	0	(3, 100, a)	156.33	1267.78	1.01	2
(2, 100, b)	188.00	234.44	1.01	0	(3, 100, b)	297.33	1800.00	-	3
(2, 100, c)	293.00	1215.17	1.05	2	(3, 100, c)	179.00	1800.00	-	3

Table 4.2 Results for *NFG*.

(V , E)	#EI	Time	<i>PoS</i>	Tl	(V , E)	#EI	Time	<i>PoS</i>	Tl
(50, 99)	5.00	0.07	1.12	0	(300, 626)	20.33	6.52	1.00	0
(100, 206)	9.00	0.13	1.00	0	(350, 730)	20.67	6.70	1.00	0
(150, 308)	9.67	0.47	1.01	0	(400, 822)	302.00	654.73	1.01	1
(200, 416)	18.67	1.85	1.00	0	(450, 934)	492.00	1200.43	1.01	2
(250, 517)	68.67	51.55	1.02	0	(500, 1060)	40.33	104.80	1.00	0

With our results, we highlight there may exist high-quality *PNEs* (i.e, small *PoS*), and how our theoretical and computational framework shows promising applications for selecting such equilibria.

Acknowledgements

We would like to thank Ulrich Pferschy for valuable discussions on our work.

CHAPTER 5 THE CUT-AND-PLAY ALGORITHM: COMPUTING NASH EQUILIBRIA VIA OUTER APPROXIMATIONS

Authors: Margarida Carvalho, Gabriele Dragotto, Andrea Lodi and Sriram Sankaranarayanan¹.

Abstract The concept of Nash equilibrium enlightens the structure of rational behavior in multi-agent settings. However, the concept is as helpful as one may compute it efficiently. We introduce the *Cut-and-Play*, an algorithm to compute Nash equilibria for a class of non-cooperative simultaneous games where each player’s objective is linear in their variables and bilinear in the other players’ variables. Using the rich theory of integer programming, we alternate between constructing (i.) increasingly tighter outer approximations of the convex hull of each player’s feasible set – by using branching and cutting plane methods – and (ii.) increasingly better inner approximations of these hulls – by finding extreme points and rays of the convex hulls. In particular, when these convex hulls are polyhedra, we prove the correctness of our algorithm and leverage the mixed integer programming technology to compute equilibria for a large class of games. Further, we integrate existing cutting plane families inside the algorithm, significantly speeding up equilibria computation. We showcase a set of extensive computational results for Integer Programming Games and simultaneous games among bilevel leaders. In both cases, our framework outperforms the state-of-the-art in computing time and solution quality.

5.1 Introduction

Game Theory and Mixed Integer Programming (*MIP*) – and some of their founding ideas – fortuitously share a common root. As noted in [138], two critical contributions in the two fields originated in Princeton and shared John Von Neumann as a common ancestor. On the one hand, linear programming duality provides an elegant and essential component of *MIP* theory and computations. On the other hand, the early development of Game Theory by Von Neumann and Morgenstern served as an opener for a rigorous mathematical methodology to model complex interactions among agents. Von Neumann brilliantly hinted at a beautiful connection between the two fields – as George Dantzig explained in [48] – by intertwining linear programming duality and zero-sum games. In a sense, the (ante litteram) game-theoretical interpretation of duality for zero-sum games initiated a symbiosis between Mathematical Programming and Game Theory. A few years later, Nash [116, 117] introduced – in his two

¹A pre-print is available in [33].

seminal doctoral papers – his solution approach for strategic behavior, namely the concept of *Nash equilibrium*. Nash’s solution concept provides a *stable* solution, in the sense that no “rational” and selfish agent would defer it and get a benefit in doing so. However, in practical applications, the plausibility of the Nash equilibrium concept can only stem from the availability of efficient tools to compute it.

In this paper, we further contribute to bridging the gap between *MIP* and Game Theory by extending some of the algorithmic rationales of *MIP* to a family of games. We strongly believe that improving the state-of-the-art techniques for equilibria computation may help extend the broad family of problems involving typical Operations Research tasks – i.e., logistics, scheduling, tactical decision-making – to a multi-agent setting. Recently, the *MIP* community has been notably active in incorporating game dynamics into optimization frameworks. Games can broaden the modeling capabilities of *MIP*, and extend classical combinatorial and decision-making problems to multi-agent settings that can account for interactions among multiple decision-makers. For instance, bilevel programming [13, 26, 72, 89, 94, 99]) and Integer Programming Games (*IPGs*) [29, 32, 47, 57, 82, 95]. This recent research direction suggests that the joint endeavor between game theory and *MIP* can widen their theoretical understanding and practical impact.

RBGs. As a standard game-theoretic notation, let the operator $(\cdot)^{-i}$ define $(\cdot)$ except i ; e.g., if $x = (x^1, \dots, x^n)$, then $x^{-2} = (x^1, x^3, \dots, x^n)$. In this paper, we study the problem of computing Nash equilibria – arguably the most notorious concept of *stable* solution for strategic behavior – for Reciprocally-Bilinear Games (*RBGs*), a class of *non-cooperative simultaneous* games among n players as in Definition 8.

Definition 8 (*RBG*). A Reciprocally-Bilinear Game (*RBG*) is a game among n players with each player $i = 1, 2, \dots, n$ solving the optimization problem

$$\min_{x^i} \quad (c^i)^\top x^i + (x^{-i})^\top C^i x^i \quad (5.1a)$$

$$\text{s.t.} \quad x^i \in \mathcal{X}^i \quad (5.1b)$$

where $\mathcal{X}^i$ is a set (not necessarily closed), C and c are a matrix and a vector with integer entries, respectively. An *RBG* is polyhedrally-representable if $\text{cl conv}(\mathcal{X}^i)$ is a polyhedron for each i , and one can optimize an arbitrary linear function on $\mathcal{X}^i$.

From the definition, the following properties hold for each player i : (i.) its objective function is *reciprocally-bilinear*, namely, it is linear in its variables x^i , and contains bilinear terms in x^i

and the other players' variables x^{-i} (ii.) its constraint set $\mathcal{X}^i$ contains only the x^i variables, i.e., the interaction takes place at the objective level (i.e., the game is not a generalized Nash equilibrium problem). Besides, we also assume that (i.) all players are rational, namely they want to minimize their objective function, and (ii.) there is *common knowledge of rationality* with complete information, namely each player i knows its optimization problem – as well as the ones of the other players – and knows that every player is rational. We refer to $\mathcal{X}^i$ as the set of (*pure*) *strategies* of i , namely the set of actions i can adapt in the game. As we will motivate later, the actual strategy set for each player is $\text{cl conv}(\mathcal{X}^i)$, as it represents the set of all (*mixed*) *strategies* the player can adopt. Finally, we assume $\text{cl conv}(\mathcal{X}^i)$ is a polyhedron in order to guarantee finite termination of our algorithms.

Contributions. In this work, we employ the rich theory of integer programming – for instance, concepts such as relaxation, valid inequalities, disjunctive programming, branching – and extend them to compute Nash equilibria for *RBGs*. Our approach is *general*, as it does not exploit any game-specific structure besides the polyhedral representability of the (mixed) strategy sets. Further, it stands on the shoulders of many *giants*: the theoretical and practical apparatus of integer programming. We show the integration of such mathematical programming tools – and specifically integer programming ones – has promising implications in equilibria computation. Specifically, we start from the concept of outer approximation. In *MIP*, one often exploits a series of increasingly tighter outer approximations – or (linear) relaxations [101, 120]. However, a game's approximation (e.g., when at least a player's optimization problem is outer approximated) does not possess the concept of bound when the solution paradigm is the Nash equilibrium. In this paper, we establish an algorithmic rationale to exploit the approximations of *RBGs*. We briefly summarize our contributions as follows.

- (i.) We introduce *Cut-and-play* (*CnP*), a general algorithm to find mixed-strategy Nash equilibria (*MNEs*) for *RBGs*. The algorithm finds an equilibrium or provides a proof of its non-existence by constructing increasingly tighter outer approximations of $\text{cl conv}(\mathcal{X}^i)$ for any player i . At every iteration, the *CnP* attempts to compute an equilibrium for an “easier” approximated game, and eventually refines the approximation by using cutting planes and branching.
- (ii.) As a subroutine to the *CnP*, we introduce an *Enhanced Separation Oracle* (*ESO*) for polyhedrally-representable sets. Given a point $\bar{x} \in \mathbb{R}^n$ and a set $\mathcal{X} \subseteq \mathbb{R}^n$, the *ESO* returns either (i.) **no** and $(\bar{\pi}, \bar{\pi}_0) \in \mathbb{R}^n \times \mathbb{R}$ such that $\bar{\pi}^\top \bar{x} > \bar{\pi}_0$ with $\bar{\pi}^\top x \leq \bar{\pi}_0$ for all $x \in \mathcal{X}$, or (ii.) **yes**, and $v^1, \dots, v^u \in \mathcal{X}$, extreme rays $r^1, \dots, r^p$ of $\text{cl conv}(\mathcal{X})$, $\alpha_1, \dots, \alpha_u \in [0, 1]$

and $\beta_1, \dots, \beta_p \in \mathbb{R}_+$ such that $\sum_{k=1}^u \alpha_k = 1$ and $\bar{x} = \sum_{k=1}^u \alpha_k v^k + \sum_{j=1}^p \beta_j r^j$. The idea grounds in the well-studied concept of separation oracle [19, 38, 71, 81, 91, 120], and the implementation we provide overcomes the computational issues associated with the ellipsoid algorithm. From a game-theoretic perspective, given a point $\tilde{\sigma}^i \in \text{cl conv}(\mathcal{X}^i)$, the *ESO* returns the pure strategies that player i plays in $\tilde{\sigma}^i$ with their associated probabilities.

- (iii.) We provide extensive computational results on *IPGs* and Nash games Among Stackelberg Players (*NASPs*) [31]. In both cases, our work improves the state-of-the-art algorithms in computing time and solution quality (given a quality's measure). Moreover, in *IPGs*, we show how *MIP* cutting planes reduce the number of *CnP* iterations and significantly improve convergence.

Outline. Section 5.2 provides a literature overview and the necessary background definitions from Game Theory, Optimization, and *MIP*. Section 5.3 presents our *CnP* algorithm. As a sine qua non component, Section 5.4 introduces the *ESO*, which separates a given point from the (unknown) set of mixed strategies. Section 5.4.1 introduces an implementation of the *ESO*, and weighs up some practical aspects concerning the algorithmic implementation. Section 5.5 presents the two main applications of the algorithm, namely *IPGs* and *NASPs*, and tailors some components of the algorithmic approach to the two families of games. Finally, Section 5.6 showcases a comprehensive set of computational results.

5.2 Related literature and background

Nash Equilibrium. We employ the standard concept of Nash Equilibrium [116, 117] as solution concept. For any given *RBG* instance, we aim to find an equilibrium or show that none exists. When there is only one player – namely $n = 1$ – the game becomes trivial, and its equilibrium collapses to the solution of a single optimization problem. However, when multiple agents are simultaneously deciding, the notion of Nash equilibrium becomes essential. At any equilibrium point, no player can *unilaterally* deviate from the equilibrium point and improve its payoff. In general, the equilibrium may map a probability distribution over every player's set of pure strategies. The pure strategies contributing to the equilibrium with a positive probability build the so-called *support* of the equilibrium. In general, we will refer to an equilibrium as a mixed-strategy Nash Equilibrium (*MNE*). However, when the support is a singleton, we refer to it as a pure-strategy Nash Equilibrium (*PNE*).

Nash [117] proves the existence of *MNE* for *finite games*, while the renowned contribution of Daskalakis et al. [49] show the task of computing an *MNE* for such games is PPAD-complete. For what concerns *continuous games* – namely when the set of strategies is infinite – Glicksberg [78] establishes the existence of *MNEs* whenever the strategy spaces are compact and utilities continuous. Stein et al. [140] prove that for all the *MNEs* in separable games – namely where players’ payoffs take a sum-of-products form – there is a payoff equivalent *MNE* of finite support. Carvalho et al. [29, 32] focus on *IPGs*, where each player strategy set is defined by inequalities and integrality requirements. They point out that the problem of deciding if an *IPG* has an *MNE*– or even a *PNE*– is Σ_2^P -complete. However, whenever the feasible region of each player is nonempty and bounded, the *IPG* always has an *MNE*. Similarly, Carvalho et al. [31] proves that deciding if a *NASP* has a *PNE*– or analogously an *MNE*– is Σ_2^P -hard. Further, they prove there is always an *MNE* if the players’ feasible regions are nonempty and bounded. In both *IPGs* and *NASPs*, an *MNE* may not exist if one or more players have an unbounded action space.

Computing equilibria. A significant segment of the algorithmic approaches for Nash equilibria computation focuses on finite games, namely games represented in normal-form (i.e., the input are matrices of payoffs for all possible game’s outcomes). The first family of algorithms is the one of complementarity-based methods. Historically, the first contribution to this family – and in general to the task of equilibria computation – is the Lemke-Howson algorithm, a path following algorithm which works for any 2-player finite (i.e., finitely many strategies, players and outcomes) game [102]. The algorithm has a strong geometrical interpretation since it represents the game with a polytope for each player and pivots among its vertices until it reaches an equilibrium vertex. Rosenmüller [127], Wilson [146] extended the Lemke-Howson algorithm for n -player games. However, such methods require the solution of a series of non-linear equation systems. The second family is the one of homotopy-based algorithms. Scarf [135] proposed a simplicial subdivision algorithm, where the equilibrium is then the fixed point over the product of unit simplices describing the strategy space of each player. Finally, a third family of approaches is the one of support enumeration algorithms. Given a 2-player game, this class of algorithms builds a linear system of equations and inequalities to determine if a given support (a subset of pure strategies) contains an equilibrium for the given game. Following this idea, Sandholm et al. [133] find equilibria by formulating mixed-integer programs, while Porter et al. [124] conceived a simple search algorithm that shows to be very efficient in practice. The latter prioritizes the search towards equilibria of small and balanced supports. A further refinement of their algorithm includes backtracking steps and an effective pruning of strictly dominated pure strategies (i.e., strategies that will never be played from a

rational agent).

However, in *RBGs*, $\mathcal{X}^i$ is large and may be even uncountable, thus the use of the described methodologies for finite games is rather impractical. Indeed, representing a game by enumerating its possible outcomes may be challenging. This applies to any game represented by each player optimization problem, for instance, games with a representation similar to the one of *RBGs*. When each player optimization problem is convex in their decision variables, a broadly studied family of algorithms is the one of equilibrium programming methods [63]. The equilibrium problem often reformulates as a non-linear complementarity problem or an equivalent variational inequality. Compared to homotopy methods, the latter family of methods does not guarantee global convergence [112]. Furthermore, several methodologies address the non-convex cases. Carvalho et al. [32] introduce the Sampled Generation Method (*SGM*) for *IPGs*. At each iteration, the algorithm computes an equilibrium in a restricted game (i.e., a game containing a subset of each player's strategies). Then, if at least one player can rationally improve its objective function by adopting a strategy outside the restricted game, the algorithm discards the previously computed equilibrium and adds such strategy to the next iteration's restricted game. Each restricted game is represented in normal-form, enabling the use of many of the methodologies described before. Cronert and Minner [47] extend the *SGM* algorithm to better select candidate equilibria if more than one exist. More recently, Dragotto and Scatamacchia [57] provide an efficient algorithm to compute and enumerate *PNEs* in *IPGs* through integer programming formulations and valid inequalities. Sagratella [130] introduces a branching method to enumerate all pure equilibria when payoffs are convex and convex constraints along with integrality requirements on variables model the strategy sets. Their approach is general and exploits a branching routine to handle integral non-convexities.

5.2.1 Background and definitions

We employ some standard definitions from convex [20], polyhedral, [42] and complementarity [46] theories. Given any set $K \in \mathbb{R}^k$, we denote as $\text{cl}(K)$, $\text{int}(K)$, $\text{bd}(K)$ the closure, interior, and boundary of K . A face F for K is a non-empty closed convex set so that if $x, y \in K$, and $(\alpha x + (1 - \alpha)y) \in F$ for any $\alpha \in [0, 1]$, then $x, y \in F$. $x \in K$ is an extreme point if it cannot be expressed as a strict convex combination of two points in K , i.e., with $0 < \alpha < 1$. A recession direction for K is a vector $r \in \mathbb{R}^k$ so that, for any $x \in K$, $x + \alpha r \in K$ for any $\alpha \geq 0$. An extreme ray for K , or extreme recession direction, is a recession direction for K that cannot be expressed as a convex combination of two or more other recession directions for K . Given a closed convex set $\bar{K} \in \mathbb{R}^k$, we denote as $\text{rec}(\bar{K})$ and $\text{ext}(\bar{K})$ the set of recession directions and extreme points of $\bar{K}$, respectively. Given two closed convex sets $\bar{K}^1 \in \mathbb{R}^k$, and

$\bar{K}^2 \in \mathbb{R}^k$, we define as their Minkowski sums $\bar{K}^1 + \bar{K}^2$ as $\{x : x = k^1 + k^2, k^1 \in \bar{K}^1, k^2 \in \bar{K}^2\}$. Depending on its structure, $\bar{K}$ can be: (i.) a polyhedron if finitely many half-spaces generate it, (ii.) a (closed) cone if for any $k \in \bar{K}$ and positive $\alpha \in \mathbb{R}$, then $\alpha k \in \bar{K}$. Furthermore, $\bar{K}$ is convex if any convex combinations of points in $\bar{K}$ still belongs to $\bar{K}$. A polyhedral cone is a special convex cone given by the conic hull of a finite set of recession directions, i.e., applying the cone operator on a finite set. If $\bar{K}$ is also bounded, Krein and Milman proved that $\bar{K} = \text{conv}(\text{ext}(\bar{K}))$. Should $\bar{K}$ be a polyhedron (therefore possibly unbounded), then the theorem is known as the Weyl theorem, and decomposes $\bar{K}$ into the (Minkowsky) sum of its extreme points $V = \text{ext}(\bar{K})$ and the conic combination of its recession directions $R = \text{rec}(\bar{K})$, namely $\bar{K} = \text{conv}(V) + \text{cone}(R)$. This vertex-ray representation of a polyhedron is also known as the $\mathcal{V}$ -polyhedral representation. A valid inequality for $\bar{K}$ is an inequality of the form $\pi^\top x \geq \pi_0$ holding for any $x \in \bar{K}$, and it is supporting if there exists an $x_0 \in \bar{K}$ so that $\pi^\top x_0 = \pi_0$, i.e., x_0 is a boundary point for $\bar{K}$, and the inequality is a face for $\bar{K}$. We define as separation oracle the blackbox solving the *separation problem* of Definition 9.

Definition 9 (Separation Problem). *Given a closed convex set $\bar{K}$, and a point $\bar{x}$, either: (i.) determine that $\bar{x} \in \bar{K}$ and output **yes**, or (ii.) determine that $\bar{x} \notin \bar{K}$, and output **no** and $(\bar{\pi}, \bar{\pi}_0) \in \mathbb{R}^k \times \mathbb{R}$, with $\bar{\pi}^\top x \leq \bar{\pi}_0$ being a valid inequality for any $x \in \bar{K}$ while $\bar{\pi}^\top \bar{x} > \bar{\pi}_0$ (i.e., a separating hyperplane).*

Given $\bar{K}$, we call a (polyhedral) set O as a (polyhedral) outer approximation of $\bar{K}$ if $\bar{K} \subseteq O$. Conversely, a (polyhedral) set I is a (polyhedral) inner approximation of $\bar{K}$ if $I \subseteq \bar{K}$. Let x be a set of k variables, M a $k \times k$ matrix, and q a k -dimensional vector. We define as (linear) *complementarity constraint* the expression $0 \leq x \perp z = (Mx + q) \geq 0$, where the $\perp$ operator serves the purpose of vector multiplication (e.g., $x \perp z \equiv x^\top z = 0$), and M and q are a matrix and a vector of appropriate dimensions. A linear complementarity problem (*LCP*) is the problem of finding a vector x such that $0 \leq x \perp z = (Mx + q) \geq 0$, or show that no such vector exist.

Games. We previously introduced the generic family of *RBGs* as *simultaneous non-cooperative* games among n players satisfying Definition 8. In particular, many classes of well-studied optimization problems are polyhedrally-representable: linear complementarity problems (*LCPs*), *MIP*, linear bilevel programs and reverse convex programs all satisfy this property [85, 86]. We provide some examples in Remark 1.

Remark 1. *If any $\mathcal{X}^i$ is either a polyhedron or a union of polyhedra, then $\text{clconv}(\mathcal{X}^i)$ is polyhedral, and the game is polyhedrally-representable. Also, IPGs with reciprocally-bilinear objectives are polyhedrally-representable. There are other polyhedrally-representable games as*

well: for example, let $\mathcal{X}^i = \{x^i \in \mathbb{R}^3 : ((x_1^i)^2 + (x_2^i)^2 \geq x_3^i, -x_3^i \leq x_1^i, x_2^i \leq x_1^i, x_3^i \geq 0)\}$. This set, despite being non-polyhedral, has a polyhedral convex-hull, $\text{cl conv}(\mathcal{X}^i) = \{-x_3^i \leq x_1^i, x_2^i \leq x_3^i, x_3^i \geq 0\}$.

We denote the feasible set of player i as $\mathcal{X}^i$. We call any point $x^i \in \mathcal{X}^i$ a *pure-strategy* of player i . When a player i randomizes over its pure-strategies $\mathcal{X}^i$, we obtain a so-called *mixed-strategy*. More formally, σ^i is a mixed-strategy – or simply a strategy – for i if it is a probability distribution over the pure strategies $\mathcal{X}^i$. Let Δ^i denote the set of mixed strategies for i , namely the space of probability distributions over $\mathcal{X}^i$. Furthermore, $\text{supp}(\sigma^i) = \{x^i \in \mathcal{X}^i : \sigma^i(x^i) > 0\}$ is the *support* of the mixed-strategy σ^i . We will refer to $\sigma^i(x^i)$ as the probability of playing x^i in $\sigma^i = \sigma^i(x^i)x^i$. Clearly, any mixed-strategy σ^i with a singleton support (e.g., $|\text{supp}(\sigma^i)| = 1$) is a pure strategy. Given a player i , we denote with $\sigma^{-i} \in \prod_{j=1, i \neq j}^n \Delta^j$ the *other players' strategies*, a probability distribution over the pure-strategies of i 's opponents. We will refer to the objective function of i evaluated for $\hat{x} = (\hat{x}^i, \hat{x}^{-i})$ – namely $f^i(\hat{x}^i, \hat{x}^{-i})$ – as the *payoff* of i for a given pure strategy profile. Note that we slightly abuse the definition of payoff, since all *RBG* players are in fact minimizing their objective functions. *RBGs* fall into the bigger category of *separable games* (e.g., the objective functions are sums of polynomials), since the players' objectives are bilinear expressions. Hence, their *MNEs* all have finite supports or finite support equivalents [140]. The expected payoff of player i for a given mixed-strategy profile $\sigma = (\sigma^1, \dots, \sigma^n)$ is

$$\begin{aligned} \mathbb{E}(f^i(\sigma^i, \sigma^{-i})) &= f^i(\sigma^i, \sigma^{-i}) = (c^i)^\top \sigma^i + (\sigma^{-i})^\top C^i \sigma^i = \\ &= \sum_{x^i \in \text{supp}(\sigma^i)} (c^i)^\top x^i \cdot \sigma^i(x^i) + \sum_{x \in \text{supp}(\sigma)} (x^{-i} \cdot \sigma^{-i}(x^{-i}))^\top C^i x^i \cdot \sigma^i(x^i). \end{aligned} \quad (5.2)$$

Equilibria. A strategy $\bar{\sigma}^i$ is a best-response strategy for player i given (its opponents' strategies) $\bar{\sigma}^{-i}$ if $f^i(\bar{\sigma}^i, \bar{\sigma}^{-i}) \leq f^i(\hat{\sigma}^i, \bar{\sigma}^{-i})$ for any other strategy $\hat{\sigma}^i$. A strategy profile $\bar{\sigma} = (\bar{\sigma}^1, \dots, \bar{\sigma}^n)$ is an *MNE* if, for each player i and strategy $\tilde{\sigma}^i \in \Delta^i$, then $f^i(\bar{\sigma}^i, \bar{\sigma}^{-i}) \leq f^i(\tilde{\sigma}^i, \bar{\sigma}^{-i})$. Any strategy x^i in the support $\text{supp}(\bar{\sigma}^i)$ of the *MNE* is a best-response strategy for i . In other words, given an *MNE* $\bar{\sigma}$, any i is indifferent among the pure-strategies in the support of $\bar{\sigma}^i$. This was formalized by Nash as in Theorem 3.

Theorem 3 (Equality of Payoffs [116, 117]). *Assume σ^i is a (mixed) best-response for player i given σ^{-i} . Then, $f^i(\sigma^i, \sigma^{-i}) = f^i(x^i, \sigma^{-i})$ for any $x^i \in \text{supp}(\sigma^i)$.*

In Theorem 4, we generalize a theorem from Carvalho et al. [31]. With this result, we ground the algorithmic rationale we present in this paper.

Theorem 4. *Consider two RBGs G and $\tilde{G}$ such that (i.) the objectives of player i in G and $\tilde{G}$ are the same, and (ii.) the feasible set of player i in G is $\mathcal{X}^i$ and in $\tilde{G}$ is $\text{cl conv}(\mathcal{X}^i)$. For any PNE $\tilde{\sigma}$ of $\tilde{G}$, there exists an MNE $\hat{\sigma}$ of G such that each player i gets the same payoff in G and $\tilde{G}$. Further, if $\tilde{G}$ has no PNE, then G has no MNE.*

In the proof (which is in appendix A.1), we exploit the linearity of each player’s objective (given its opponents’ choices) to compute the players’ expectation of payoffs. Thus, assuming the game is an *RBG* is crucial for the result to hold. As of Theorem 4, any i -th player’s problem in a polyhedrally-representable *RBG* translates to an equivalent convex game where the feasible set is $\text{cl conv}(\mathcal{X}^i)$ instead of $\mathcal{X}^i$. In such a game, one can directly optimize over the mixed strategies, and each player i solves a linear program, where x^i are its variables and x^{-i} are parameters. By pairing all the players’ complementarity conditions – namely the primal-dual slackness of each i -th linear program – we obtain an *LCP* where any feasible solution is a *PNE* for the convex game [46]. However, one may not necessarily have access to $\text{cl conv}(\mathcal{X}^i)$ (e.g., an integer programming perfect formulation). In this sense, reformulating the game on the $\text{cl conv}(\mathcal{X}^i)$ -sets may be practically unviable.

5.2.2 Integer Programming Games

IPGs extend the realm of integer programming to a multi-agent setting. Each player i solves the integer program

$$\min_{x^i} \quad \Pi^i(x^i, x^{-i}) \quad (5.3a)$$

$$\text{s.t.} \quad A^i x^i \leq b^i, x^i \geq 0, x_j^i \in \mathbb{N} \forall j \in \mathcal{I}^i. \quad (5.3b)$$

The matrix A^i and the vector b^i have rational entries, and we require some variables – whose indexes are in $\mathcal{I}^i$ – to be integer. To follow the standard notation from Köppe et al. [95], Π^i is a continuous payoff function. Whenever Π^i takes the form of the objective in Definition 8, the *IPG* is also an *RBG*. This latter family of *IPGs* is of particular interest from the *MIP* community since it naturally extends a broad range of tasks from the Operations Research community to a multi-agent setting. For the scope of this work, we focus on *IPGs* that are also *RBGs*. Hence, for any i , the set $\mathcal{X}^i$ is the set of feasible points for (5.3). The perfect formulation of the feasible region in (5.3) for any player i is $\text{cl conv}(\mathcal{X}^i)$. This formulation is notoriously difficult to obtain in practice, and often contains a number of linear inequalities that is exponential in the size of the data needed to describe the problem [41]. Finally, one can visualize the interaction among i and its opponents as the change in the direction of the i -th objective function due to the opponent’s parameters x^{-i} .

5.2.3 NASPs

We recall the definition of *NASPs* problems from [31]. In this game, each player $i \in \{1, \dots, n\}$ solves the optimization problem $P^i(x^i, x^{-i})$ in x^i (for brevity, P^i) in the form of (5.4), where c^i is a real-valued vector of dimension n_i , and Q_o^i is a $n_o \times n_i$ real-valued matrix encapsulating the interactions between any two distinct players i and o . Any leader i has m_i followers, each of which solves a convex continuous optimization problem as in (5.4c). For a given leader i , and its respective follower $j \in \{1, \dots, m_i\}$, $f^{i,j}$, $e^{i,j}$ and $D^{i,j}, E^{i,j}, F^{i,j}, G^{i,j}, H^{i,j}$ are respectively vectors and matrices of conformable dimensions. Each feasible set $\mathcal{X}^i$ is given by the constraints in (5.4), namely a set of linear constraints (5.4b) and bilevel ones (5.4c). Furthermore, the variables x^i are partitioned into the leader's variables w^i , and the followers' ones $y^i = (y^{i,1}, \dots, y^{i,m_i})$ as of (5.4d). Thus, the mathematical program reads as

$$\min_{x^i} (c^i)^\top x^i + \sum_{o=1, o \neq i}^n (x^o)^\top Q_o^i x^i \quad (5.4a)$$

$$\text{s.t.} \quad A^i x^i = b^i \quad (5.4b)$$

$$\begin{aligned} y^{i,j} \in \arg \min_{y^{i,j}} \{ & (\frac{1}{2}(y^{i,j})^\top D^{i,j} y^{i,j} + (f^{i,j} + \sum_{k=1, k \neq j}^{m_i} (y^{i,k})^\top E^{i,j}) y^{i,j} + \\ & (F w^i)^\top y^{i,j} : G^{i,j} w^i + H^{i,j} y^{i,j} \leq e^{i,j}, y^{i,j} \geq 0\} \\ & \forall j \in \{1, \dots, m_i\} \end{aligned} \quad (5.4c)$$

$$x^i \geq 0, x^i = (w^i, y^i). \quad (5.4d)$$

Any single problem P^i is a linear *Stackelberg Game* [139] parametrized in x^{-i} , while a *NASP* $P = (P^1, \dots, P^n)$ is a tuple of n Stackelberg Games. Leaders interact through their objectives, while followers can interact only with their respective leader and followers. Thus, leaders are simultaneously deciding their strategies while ensuring optimality conditions for their followers. For any given leader i , its feasible region $\mathcal{X}^i$ (5.4b)-(5.4d) is a finite union of polyhedra [31], and hence *NASPs* are polyhedrally-representable *RBGs*. One can rewrite $\mathcal{X}^i$ as

$$\mathcal{X}^i = \left\{ x^i : \begin{array}{l} A^i x^i \leq b^i \\ z^i = M^i x^i + q^i \\ 0 \leq x_j^i \perp z_j^i \geq 0 \\ x^i \geq 0 \end{array} \quad \forall j \in \mathcal{C}^i \right\} \quad (5.5)$$

where $\mathcal{C}^i$ is a set of indexes for the complementarity equations. With such notation, we rewrite

the followers' convex quadratic linear problems into identical complementarity conditions derived from the *KKT* conditions. We point the reader to Colson et al. [40] for a tutorial in the context of bilevel problems, and to Carvalho et al. [31] for a tutorial on *NASPs*. We can further rewrite (5.5) as

$$\mathcal{X}^i = \underbrace{\left\{ \begin{array}{l} A^i x^i \leq b^i \\ z^i = M^i x^i + q^i \\ x^i \geq 0, z^i \geq 0 \end{array} \right\}}_{O_0^i} \bigcap_{j \in \mathcal{C}^i} (\{z_j^i = 0\} \cup \{x_j^i = 0\}). \quad (5.6)$$

In this last reformulation, we explicitly express $\mathcal{X}^i$ as a finite union of polyhedra. For any leader i , we refer to the first polyhedron in (5.6) as the *polyhedral relaxation* O_0^i . In other words, this is the polyhedron containing the leader constraints, the definitions for z^i , and the non-negativity constraints. For Theorem 4, given a leader i , if $\sigma^i \in \mathcal{X}^i$ then σ^i is a pure-strategy for i , otherwise, i.e., if $\sigma^i \in \text{cl conv}(\mathcal{X}^i) \setminus \mathcal{X}^i$, σ^i is a mixed-strategy. Hence, we optimize the leaders' objective functions over $\text{cl conv}(\mathcal{X}^i)$ rather than $\mathcal{X}^i$ (for any i). We can then consider $\text{cl conv}(\mathcal{X}^i)$ by exploiting the well-known extended formulation of the union of polyhedra from Balas' theorem [8, 9].

5.3 Algorithmic Scheme

First, we briefly sketch the ideas behind our algorithm. In a nutshell, the *CnP* computes an *MNE* for an *RBG* instance by computing the *MNE* of a sequence of “easier” convex games, namely what we define as *approximate games*.

Definition 10 (Approximate Game). *Given an RBG instance G , $\tilde{G}$ is an approximate game for G if and only if: (i.) G and $\tilde{G}$ have the same number of players, and their payoff functions are equal (ii.) for each $i = 1, \dots, n$, let $\mathcal{X}^i$ and $\tilde{\mathcal{X}}^i$ be the feasible region of player i in G and $\tilde{G}$, respectively: then, $\tilde{\mathcal{X}}^i \supseteq \mathcal{X}^i$ ($\tilde{\mathcal{X}}^i$ is an outer approximation of $\mathcal{X}^i$). Further, $\tilde{G}$ is a Polyhedral Approximate Game (PAG) of G if $\tilde{\mathcal{X}}^i$ is a polyhedron for each $i = 1, \dots, n$ in $\tilde{G}$.*

In Definition 10 we let $\tilde{\mathcal{X}}^i$ be an outer approximation – possibly polyhedral in *PAGs*– of $\text{cl conv}(\mathcal{X}^i)$, namely the mixed-strategy space for player i . The *CnP* will compute *MNEs* for a sequence of *PAGs*, eventually refining one or more of the $\tilde{\mathcal{X}}^i$ by either adding cutting planes or branching on general disjunctions. This process evokes the same scheme one would use to solve a *MIP* via Branch and Cut [120], where – instead of a game and an approximate game – one deals with a *MIP* and its (linear) relaxation. As the $\tilde{\mathcal{X}}^i$ -sets are polyhedral, one can

formulate an *LCP* to determine a *PNE* in the *PAG*. Since *LCP*s are $\mathcal{NP}$ -hard problems, at each iteration of the *CnP*, we will solve a difficult problem. This is not surprising, given that both deciding on the existence of an *MNE* for *IPGs* and *NASPs* are Σ_2^P -hard problems, for instance. In *MIP*, a feasible solution to the original problem is always feasible for its linear relaxation, and there is a relationship between the bounds of the two. However, this may not be the case in games.

Optimization, Relaxations and Games. One refines the player's feasible regions as one would refine a series of increasingly accurate relaxations of an integer program. A feasible solution to the original problem is always feasible for its relaxation, and often there is a relationship between the bounds of the two. However, when dealing with games and Nash equilibria, such a strong relationship does not necessarily hold, as we illustrate in Example 4. Carvalho et al. [32] provides a similar example in the context of *IPGs*. A feasible *MNE* for the original problem may not be an *MNE* for the *PAG*, since the latter may introduce (infeasible) profitable deviations for some players, i.e., strategies that are not feasible but prevents the existence of an equilibrium. As a consequence, there is no concept of bound on the players' objective functions. When players have possibly unbounded feasible regions, a *PAG* may not even have an equilibrium, whereas the original game has one (Example 4). In this sense, one loses information by outer approximating one or more of the players' $\tilde{\mathcal{X}}^i$ sets.

Example 4. Consider an *RBG* with $n = 2$: Player 1 solves $\min_x \{\xi x : x \in \mathbb{R}, x \geq 1\}$ while Player 2 solves $\min_\xi \{x\xi : \xi \in \mathbb{R}, \xi \in [1, 2]\}$. This game has an *MNE*: $(x, \xi) = (1, 1)$. Consider now the *PAG* where Player 2's feasible region is $\{\xi \in \mathbb{R} : \xi \in [-1, 2]\}$. Then, this *PAG* has no *MNE* despite the original game has one. Second, if Player 2 objective becomes $-x\xi$, then the original problem does not have an *MNE*, while the *PAG* has the *MNE* $(x, \xi) = (1, 2)$.

Algorithmic outline. Through this section, we provide an abstract rationale for the *CnP* algorithm. While the scheme works on any *RBG* and it is general in this sense, we will later contextualize it for *IPGs* and *NASPs* in Section 5.5. We start from a game in the form of (5.1), assuming to have access to an initial *PAG*. With *NASPs*, a natural choice is the polyhedral relaxation $\mathcal{O}_0^i$ for any i , while for *IPGs* is the linear relaxation of each player's integer program. For any given player i , let $\tilde{\mathcal{X}}_t^i = \{\tilde{A}_t^i x^i \leq \tilde{b}_t^i, x^i \geq 0\}$ be the increasingly accurate outer approximation of its feasible region $\mathcal{X}^i$ at step t of the cutting plane algorithm. We compute an equilibrium for the *PAG* by building an *LCP* where each player i solves an optimization problem over $\tilde{\mathcal{X}}_t^i$ (and the original objective function for the input *RBG*). Let M_t , and q_t be a vector and a matrix defined as in

$$q_t = \begin{bmatrix} c^1 \\ \tilde{b}_t^1 \\ \vdots \\ c^n \\ \tilde{b}_t^n \end{bmatrix}, M_t = \begin{bmatrix} C^1 x^{-1} & \tilde{A}_t^{1\top} \\ -\tilde{A}_t^1 & 0 \\ \vdots & \\ C^n x^{-n} & \tilde{A}_t^{n\top} \\ -\tilde{A}_t^n & 0 \end{bmatrix}. \quad (5.7)$$

The objects in (5.7) model the *KKT* conditions for each player's convex program associated with the outer approximation of its optimization problems. In other words, they are the optimality conditions associated with the objective function of the original game and the outer approximated feasible regions. The solutions to the *LCP* defined as $0 \leq \sigma \perp z = (M_t \sigma + q_t) \geq 0$ provide all the *PNEs* $\sigma = (\sigma^1, \dots, \sigma^n)$ for the *PAG* at step t . Similarly to *MIP*'s relaxations, we refine $\tilde{\mathcal{X}}_t^i$ to $\tilde{\mathcal{X}}_{t+1}^i$ in two ways: by *cutting* or by a *branching* decision. For *cutting*, we point to the addition of cutting planes that are valid for $\text{cl conv}(\mathcal{X}^i)$. As for *branching*, we borrow the term from *MIP* to refer to the inclusion of a general disjunction over one or more variables.

5.3.1 The Cut-and-Play Algorithm

We present the *CnP* algorithm in Algorithm 3. The input is a polyhedrally-representable *RBG* instance G (a numerical tolerance ϵ), while the output is either an *MNE* $\hat{\sigma}$ or a certificate of non-existence. We assume to have access to an initial *PAG* $\tilde{G}$. For *IPGs*, the most natural choice is the linear relaxation of each player's integer program. For any given player i , let $\tilde{\mathcal{X}}_0^i$ be the starting outer approximation for the i -th feasible region at the starting iteration $t = 0$. In $\tilde{G}$, the feasible sets are polyhedra, and hence the *MNE* of $\tilde{G}$ are all *PNEs* (if any). We determine if $\tilde{G}$ has *PNEs* by solving the *LCP* defined as $0 \leq \sigma \perp z = (M_t \sigma + q_t) \geq 0$ with M_t and q_t defined as in (5.7). If $\tilde{G}$ has no *PNE*, we cannot infer that G has no *MNE* (Example 4). This non-existence condition triggers when there is at least one i with an unbounded $\tilde{\mathcal{X}}_t^i$. The only viable option is to further improve $\tilde{G}$ by refining at least one $\tilde{\mathcal{X}}_t^i$ (for some i) with the *Branch-and-Cut* subroutine, where the algorithm branches and may add a valid inequality for $\text{cl conv}(\mathcal{X}^i)$ (Step 15). The inequality here is a discretionary operation, and does not affect finite termination (yet, it may affect performance). In specific, since this step only occurs when there is no *PNE* for $\tilde{G}$, there is not even a point to cut off. The only viable option is, in fact, to refine the approximation. Assume $\mathcal{X}^i = Y^i \cup Z^i$ where Y^i, Z^i are two arbitrary sets so that $\text{cl conv}(\mathcal{X}^i) = \text{cl conv}(Y^i \cup Z^i)$. If at a step t , there exists a $\tilde{\sigma}^i \in \tilde{\mathcal{X}}_t^i \setminus \text{cl conv}(\mathcal{X}^i)$, then the branching operation accounts to finding a Y_{t+1}^i and Z_{t+1}^i so that

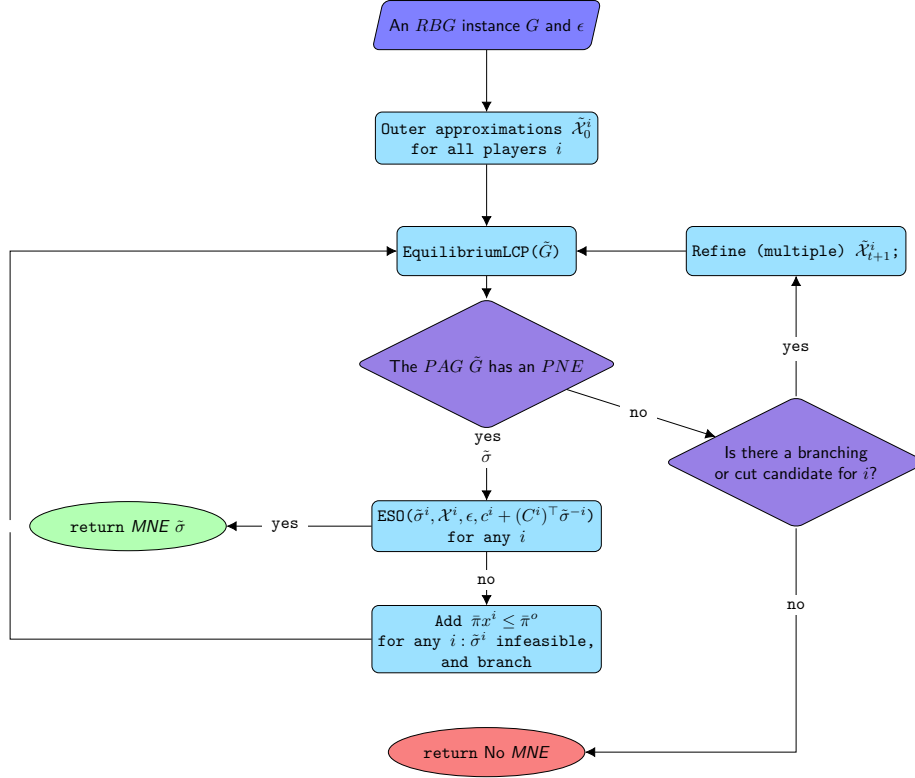


Figure 5.1 A graphical representation of the CnP algorithm.

MIP and Equilibria Selection. In order to determine if a given $PAG \tilde{G}$ has an MNE , we reformulate the game as an LCP as of Step 4. One can either solve directly the LCP via a specialized solver (i.e., $PATH$ by [56, 70]), or via a MIP reformulation. In the latter case, one can get a MIP program from Step 4 by reformulating the complementarity conditions (i.e., with $SOS1$, bgM , or indicator constraints [93]). Although a MIP reformulation hides the underlying complementarity structure that a solver such as $PATH$ may exploit, it has two main benefits for our algorithmic machine. The MIP solver can optimize an arbitrary objective function $w : \prod_{i=1}^n x^i \rightarrow \mathbb{R}$ (up to the given solver's capability) to select an MNE in the $PAG \tilde{G}$ accordingly. While the resulting MNE may not minimize w among all the MNE s of G , it necessarily minimizes w among the MNE s of $\tilde{G}$ for w . In this regard, the algorithmic scheme may incorporate some equilibria selection through MIP , and integrate with existing methodological advancements in the context of MIP and complementarity constraints [147].

Theorem 5. *Given a polyhedrally-representable RBG instance G , Algorithm 3 terminates finitely and (i.) if it returns $\hat{\sigma} = (\hat{\sigma}^1, \dots, \hat{\sigma}^n)$ then $\hat{\sigma}$ is an MNE for G , and (ii.) if it returns failure, then G has no MNE .*

Proof. First, we prove the algorithm terminates finitely.

Termination. The calls to solve the *LCP* in Step 4, and to the *ESO* in Step 7 finitely terminate (see Proposition 4). The only loop that could potentially not terminate is the one starting in Step 2. We restrict to the case where the set $\mathcal{X}^i$ is bounded for any player i . Then, any *PAG* $\tilde{G}$ will necessarily have finitely many *PNEs* with finite support in Step 4. Furthermore, since there always exists a *PNE* for $\tilde{G}$, the algorithm will never enter in Step 12. Thus, Step 9 and Step 10 are the only two steps refining the sets $\tilde{\mathcal{X}}_t^i$ for some i . Then: (i.) the *ESO* terminates finitely (we defer the appropriate proof to the next section), and (ii.) there are finitely many *PNEs* in $\tilde{G}$, and (iii.) the branching step along with the *ESO* will necessarily find $\text{cl conv}(\mathcal{X}^i)$ for any i at some point. Thus, the algorithm will terminate finitely whenever $\mathcal{X}^i$ is bounded for any i . Whenever at least one set $\mathcal{X}^i$ is unbounded, then a *PNE* for a given *PAG* $\tilde{G}$ may not exist. Thus, the algorithm may enter Step 12. Then, Step 15 will necessarily find $\text{cl conv}(\mathcal{X}^i)$ at some step. Then either (i.) there exists a *PNE* in $\tilde{G}$ that is also an *MNE* for G , and the algorithm returns it, or (ii.) there exists no *PNE* in $\tilde{G}$, and the algorithm returns $\emptyset$. Thus, the algorithm terminates finitely.

Proof of statements (i) and (ii). We show that $\hat{\sigma}$ is an *MNE* for G . If the algorithm returns $\hat{\sigma}$, then there exists an approximate game $\tilde{G}$ in Step 3 that outputed a *PNE* $\tilde{\sigma} = \hat{\sigma}$. Let this last iteration be denoted with $t = \theta$. For each player i , its feasible region in the approximate game is $\tilde{\mathcal{X}}_\theta^i$, and the following equilibrium inequalities hold

$$f^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \leq f^i(\bar{\sigma}^i, \hat{\sigma}^{-i}) \quad \forall \bar{\sigma}^i \in \tilde{\mathcal{X}}_\theta^i. \quad (5.8)$$

Namely, no player i has an incentive to deviate from $\hat{\sigma}^i$ to any other strategy bar $\sigma^i \in \tilde{\mathcal{X}}_\theta^i$ in the approximate game. Since $\text{cl conv}(\mathcal{X}^i) \subseteq \tilde{\mathcal{X}}_\theta^i$ for any i , the following holds

$$f^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \leq f^i(\bar{\sigma}^i, \hat{\sigma}^{-i}) \quad \forall \bar{\sigma}^i \in \text{cl conv}(\mathcal{X}^i). \quad (5.9)$$

Since there cannot exist a strategy $\check{\sigma}^i \in \tilde{\mathcal{X}}_\theta^i \setminus \text{cl conv}(\mathcal{X}^i)$ by construction, $\hat{\sigma}$ is also an *MNE* for G . $\square$

Remark 2. Algorithm 3 extends to RBGs that are not necessarily polyhedrally-representable, as long as one can optimize a linear function over each player's feasible set, after intersecting them with some polyhedra. Up to modifications in the choices of the convex hull of the players' feasible set, and of the branching and cutting steps (e.g., disjunction on integer variables or bilevel sets), the algorithm generalizes to any RBGs up to an ϵ numerical precision if the associated *ESO* terminates finitely for every $\text{cl conv}(\mathcal{X}^i)$. Further, one can heuristically (i.) avoid branching in Step 10 and only add a valid cut in the previous step (ii.) skip the cutting in Step 15. Further, a MIP solver can handle the LCPs of Step 4 with a non-void

objective function $w : \prod_{i=1}^n \mathcal{X}^i \rightarrow \mathbb{R}$ in the players variables, and select the PNE of $\tilde{G}$ that minimizes w . Finally, one may add other types of valid inequalities for any $\text{clconv}(\mathcal{X}^i)$ after Step 10, e.g., families of MIP inequalities in an IPG.

5.4 Implementing the *ESO*

Given a set $\mathcal{X} \in \mathbb{R}^d$ and a point $\bar{x} \in \mathbb{R}^d$ as inputs – and assuming to have access to a blackbox to optimize a linear function over $\mathcal{X}$ – the *ESO* will either:

- (i.) outputs **yes** and (V, α) if $\bar{x} \in \text{clconv}(\mathcal{X})$, with $V \subseteq \mathcal{X}$, and $\alpha \in \mathbb{R}^{|V|}$ being the coefficients of the convex combination of elements in V (i.e., $\bar{x} \in \text{conv}(V) \subseteq \text{clconv}(\mathcal{X})$),
or
- (ii.) outputs **no** and a tuple $(\bar{\pi}, \bar{\pi}_0)$ so that $\bar{\pi}^\top x \leq \bar{\pi}_0$ with $x \in \mathcal{X}$ is a separating hyperplane for $\bar{x}$ and $\text{clconv}(\mathcal{X})$.

Compared to a standard separation oracle [81, 91], here $\mathcal{X}$ is not convex, and we separate from $\text{clconv}(\mathcal{X})$. The separation from $\text{clconv}(\mathcal{X})$ is crucial for its applicability to *RBGs*, as of Theorem 4. Since any *RBG* has an equivalent convex representation (i.e., where each player plays on $\text{clconv}(\mathcal{X}^i)$), one would expect, given a feasible input set $\mathcal{X}^i$ and a point $\tilde{\sigma}^i$, to determine that either: (i.) $\tilde{\sigma}^i$ is a mixed-strategy and $\text{supp}(\tilde{\sigma}^i) = V$ with α being the probabilities of each strategy in V , or (ii.) $\tilde{\sigma}^i \notin \text{clconv}(\mathcal{X}^i)$, and $\bar{\pi}^\top x^i \leq \bar{\pi}_0$ is a separating hyperplane for the set of mixed strategies $\text{clconv}(\mathcal{X}^i)$.

Practical and game-theoretical considerations. A theoretical version of this *ESO* would include polynomially-many runs of the ellipsoid algorithm, which would make it rather intractable in practice. We provide a $\mathcal{V}$ -polyhedral implementation of it, where we explicitly require $\text{clconv}(\mathcal{X})$ to be a polyhedron. This allows us to decompose $\text{clconv}(\mathcal{X})$ as conic combination of its rays $\text{rec}(\text{clconv}(\mathcal{X}))$ and convex combination of its extreme points $\text{ext}(\text{clconv}(\mathcal{X}))$ (the $\mathcal{V}$ -polyhedral representation). The *ESO* will iteratively build an inner approximation of $\text{clconv}(\mathcal{X})$ by identifying (and storing) its rays and vertices. If the input point $\bar{x}$ cannot be expressed by the incumbent inner approximation of $\text{clconv}(\mathcal{X})$, the oracle either tries to recover new vertices and rays or outputs a separating hyperplane. More importantly, this implementation further exploits two fundamental game-theoretical interpretations of the underlying optimization problems that players solve. First, any strategy supporting an *MNE* must be a pure best-response, thus any vertex should be a pure-strategy best-response. One can visualize this by taking a player i and its opponents' strategies

σ^{-i} , and noticing that the best-response σ^i is then the solution of a parametrized – in σ^{-i} – mathematical program. This implies the actual set of extreme points needed to describe the subset of $\text{cl conv}(\mathcal{X}^i)$ needed for an *MNE* may be smaller – in practice – than $\text{cl conv}(\mathcal{X}^i)$ itself. Second, Theorem 3 provides that the utilities of any of the pure strategies in the support of the *MNE* must be equal to the one of the *MNE* itself.

The Value Cuts. Starting from these two observations, we develop an optimization-based test to possibly diagnose the infeasibility of a given strategy profile for an approximated game $\tilde{G}$ with respect to the original *RBG* G . We directly exploit Theorem 3, which provides a necessary condition on the players' payoffs in an *MNE*. Given a set of strategies $\tilde{\sigma} = (\tilde{\sigma}^1, \dots, \tilde{\sigma}^n)$ for $\tilde{G}$, we can check if any of the $\tilde{\sigma}^i$ have a payoff that improves (e.g., is less than) the one of a pure best-response to $\tilde{\sigma}^{-i}$ in G . For the i -th player, consider the mathematical program $\bar{z}^i = \min_{x^i} \{f^i(x^i, \tilde{\sigma}^{-i}) : x^i \in \mathcal{X}^i\}$, which can be optimized by assumption. If $\bar{z}^i > f^i(\tilde{\sigma}^i, \tilde{\sigma}^{-i})$, then a valid separating hyperplane for $\text{cl conv}(\mathcal{X}^i)$ is $f^i(x^i, \tilde{\sigma}^{-i}) \geq \bar{z}^i$. We call these simple linear inequalities as *value cuts*, and formalize them in Proposition 3. Clearly, such inequalities are valid for the $\text{cl conv}(\mathcal{X}^i)$ -sets of each player.

Proposition 3. *Consider an RBG G , and an arbitrary game approximation $\tilde{G}$ of G . Then, given a strategy profile $\tilde{\sigma} = (\tilde{\sigma}^1, \dots, \tilde{\sigma}^n)$ for $\tilde{G}$, for any i ,*

$$f^i(x^i, \tilde{\sigma}^{-i}) \geq \inf_{x^i} \{f^i(x^i, \tilde{\sigma}^{-i}) : x^i \in \mathcal{X}^i\}$$

*is a (supporting) valid inequality for $\text{cl conv}(\mathcal{X}^i)$ if $\inf_{x^i} \{f^i(x^i, \tilde{\sigma}^{-i}) : x^i \in \mathcal{X}^i\} = z^i < \infty$. If $z^i > f^i(\tilde{\sigma}^i, \tilde{\sigma}^{-i})$, then we call the inequality a *value cut* for $\text{cl conv}(\mathcal{X}^i)$ and $\tilde{\sigma}^i$.*

The proof, which follows from the definition of infimum, is in (appendix A.2).

5.4.1 The $\mathcal{V}$ -Polyhedral *ESO*

In this section, we discuss the $\mathcal{V}$ -polyhedral implementation of the *ESO*, including matters concerning cutting planes generation and numerical stability. Algorithm 4 sketches the implementation we will refer through this section. We have two additional inputs: (i.) a numerical tolerance ϵ , and (ii.) an (optional) real-valued vector c having the same length of the input point $\bar{x}$ (to perform the test of Proposition 3). For instance, in Step 7 of Algorithm 3, we call the *ESO* with $c = c^i + (C^i)^\top \tilde{\sigma}^{-i}$, the set $\mathcal{X}^i$ of i , $\bar{x} = \tilde{\sigma}^i$, and ϵ . We also expect one to store and access the elements of V, R across different *ESO*'s calls, and we (initially) assume $V = R = \emptyset$.

Algorithm 4: Enhanced Separation Oracle

Data: A point $\bar{x}$, a set $\mathcal{X}$, a tolerance ϵ , astorage of V, R , (a vector c).

Result: Either: (i.) **yes** and (V, R, α, β) if $\bar{x} \in \text{cl conv}(\mathcal{X})$, or (ii.) **no** and a separating hyperplane $\bar{\pi}^\top x \leq \bar{\pi}_0$ for $\text{cl conv}(\mathcal{X})$ and $\bar{x}$.

```

1  $\tilde{x} \leftarrow \arg \min_x \{c^\top x : x \in \mathcal{X}\}$ , with  $\bar{z} = c^\top \tilde{x}$ . ;
2 if  $c^\top \bar{x} < \bar{z}$  then return no and  $(-c, -\bar{z})$  ;
3 if  $\bar{x} = \tilde{x}$  up to  $\epsilon$  then return yes and  $(\{\bar{x}\}, \emptyset, (1), ())$  ;
4 while true do
5    $\mathcal{W} \leftarrow \text{conv}(V) + \text{cone}(R)$ . PRLP:  $\bar{x} \in \mathcal{W}$ ? (up to  $\epsilon$ )
6   if  $\bar{x} \in \mathcal{W}$  then return yes and  $(V, R, \alpha, \beta)$  ;
7   else
8     Separating hyperplane  $\bar{\pi}^\top x \leq \bar{\pi}_0$  for  $\bar{x}$  and  $\mathcal{W}$  ;
9      $\mathcal{G} \leftarrow \max_x \{\bar{\pi}^\top x : x \in \mathcal{X}\}$  /* Blackbox */
10    if  $\mathcal{G}$  is unbounded then
11       $R \leftarrow R \cup \{r\}$ , where  $r$  is an extreme ray of  $\mathcal{G}$  ;
12    else
13      if  $\bar{\pi}^\top \nu < \bar{\pi}^\top \bar{x}$  then return no and  $(\bar{\pi}, \bar{\pi}^\top \nu)$  ;
14      else  $\nu \leftarrow \arg \max_x \{\bar{\pi}^\top x : x \in \mathcal{X}\}$ , and  $V \leftarrow V \cup \{\nu\}$  ;

```

As a first (and optional) step, we check if there is any violated value cut by solving the optimization problem $\bar{z} = \min_x \{c^\top x : x \in \mathcal{X}\}$ of Step 1. Specifically, we compare the value of $c^\top \bar{x}$ to the one of $\bar{z}$ in Step 1. Let $\tilde{x}$ be the minimizer yielding $\bar{z}$. If (i.) $c^\top \bar{x} < \bar{z}$ (up to ϵ), then the *ESO* returns a (value) cut (Step 2), or (ii.) $\bar{x} = \tilde{x}$, then the *ESO* returns **yes** (Step 3). Otherwise, let $\mathcal{W}$ of Step 5 be so that $\mathcal{W} = \text{conv}(V) + \text{cone}(R)$. The task is now to determine if $\bar{x} \in \mathcal{W}$. The task is then to determine if $\bar{x} \in \mathcal{W}$.

The Point-Ray Separator. In order to decide if $\bar{x} \in \mathcal{W}$, we formulate the separation problem through a linear program. The task is to express $\bar{x}$ as the sum of a convex combination of elements in V and a conic combination of elements in R . Let α (β) be the convex (conic) coefficients for any element in V (R). Then, $\bar{x} \in \mathcal{W}$ if and only if there exists a solution (α, β) to

$$\sum_{k=1}^{|V|} v_k^\top \alpha_k + \sum_{j=1}^{|R|} r_j^\top \beta_j = \bar{x}, \quad \sum_{k=1}^{|V|} \alpha_k = 1, \quad \alpha \geq 0, \quad \beta \geq 0. \quad (5.10)$$

By linear programming duality, (5.10) has no solution if there exists a solution $(\bar{\pi}, \bar{\pi}_0)$ to (5.11) so that $\bar{\pi}^\top \bar{x} - \bar{\pi}_0 > 0$.

$$\begin{aligned}
\pi v_k^\top - \pi_0 &\leq 0 & \forall v_k \in V \\
\pi r_j^\top &\leq 0 & \forall r_j \in R
\end{aligned} \tag{5.11}$$

Starting from (5.11), we require our separator to have two additional features: (i.) to maximize the violation $\bar{x}^\top \pi - \pi_0$, and (ii.) to normalize the coefficients π so that $\|\pi\|_1 = 1$. We write the final program as in (5.12) and define it as the Point Ray Linear Program (*PRLP*).

$$\max_{\pi, \pi_0} \quad \bar{x}^\top \pi - \pi_0 \tag{5.12a}$$

$$\text{s.t.} \quad \pi v_k^\top - \pi_0 \leq 0, \quad \forall v_k \in V \tag{5.12b} \quad (\alpha)$$

$$\pi r_j^\top \leq 0, \quad \forall r_j \in R \tag{5.12c} \quad (\beta)$$

$$\pi + u - v = 0 \tag{5.12d} \quad (\gamma)$$

$$e^\top(u + v) = 1 \tag{5.12e} \quad (\delta)$$

$$u, v \geq 0 \tag{5.12f}$$

This *PRLP* has a similar formulation to the ones of [38, 122]. Each vertex $v_k \in V$ (resp., ray $r_j \in R$) requires constraints such as (5.12b) (resp., (5.12c)), while the non-negative variables u, v represent the L_1 -norm of π . The normalization constraints (5.12d) and (5.12e) truncate the cone of the *PRLP* by setting such L_1 -norm to 1. Let $\bar{\pi}, \bar{\pi}_0$ be the optimal values of π, π_0 in the *PRLP*. On the one hand, if the *PRLP* admits an optimal solution with objective of 0, the oracle returns **yes** (Step 6) since $\bar{x} \in \mathcal{W} \subseteq \text{cl conv}(\mathcal{X})$. The convex multipliers α (resp., conic multipliers β) are the dual values of (5.12b) (resp., (5.12c)). On the other hand, if $\bar{x}^\top \bar{\pi} - \bar{\pi}_0 > 0$, then $\bar{\pi}^\top x \leq \bar{\pi}_0$ is a separating hyperplane for $\bar{x}$ and $\mathcal{W}$. In order to determine if $\bar{\pi}^\top x \leq \bar{\pi}_0$ is also a separating hyperplane for $\bar{x}$ and $\text{cl conv}(\mathcal{X})$, in Step 9 we optimize the inequality over $\mathcal{X}$. If $\mathcal{G} = \max_x \{\bar{\pi}^\top x : x \in \mathcal{X}\}$ is unbounded, then its extreme ray r is a new ray for the set R . Conversely, if $\mathcal{G}$ admits an optimal solution ν , the latter is necessarily a new vertex for the set V (Step 14). Furthermore, if $\bar{\pi}^\top \nu < \bar{\pi}^\top \bar{x}$, then $\bar{x}$ is necessarily infeasible. De facto, this means $\bar{x}$ is separated from $\text{cl conv}(\mathcal{X})$ by $\bar{\pi}^\top x \leq \bar{\pi}^\top \nu$, and the *ESO* returns **no**. If this is not the case, the *ESO* necessarily identified a new vertex (or ray), and the process restarts from Step 5.

Similarly to Perregaard and Balas [122], one can modify Step 9 of Algorithm 4 to retrieve multiple vertices and rays violating $\bar{\pi}^\top x \leq \bar{\pi}_0$, and subsequently add them in Step 14 and

Step 11. In this way, the $\mathcal{W}$ inner approximation tends to build faster without significantly impacting the computational overhead.

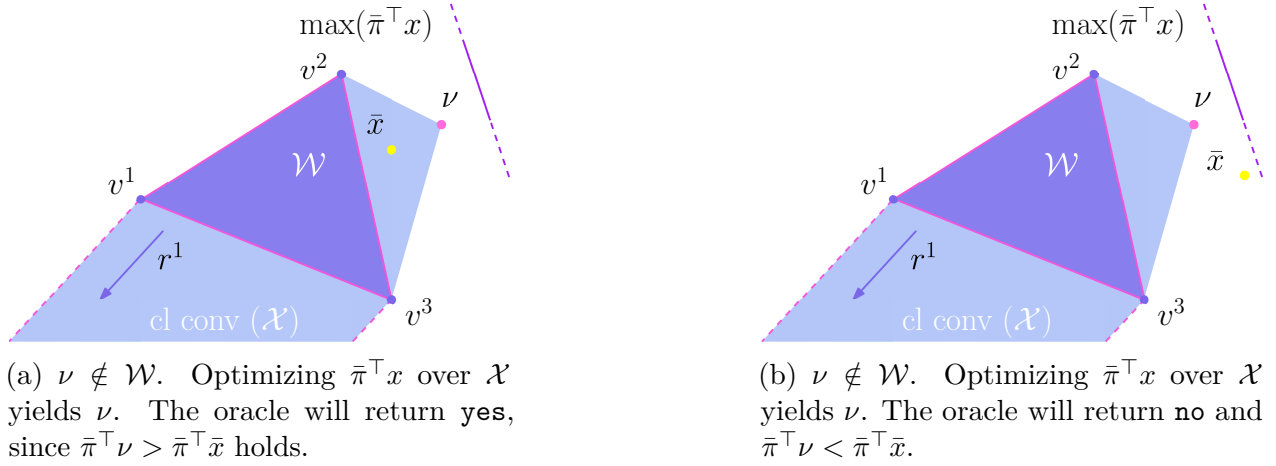


Figure 5.2 A 2-dimensional example of Algorithm 4 trying to separate $\bar{x}$ from $\text{cl conv}(\mathcal{X})$. Here, $\mathcal{X} = \{\text{conv}(v^2, \nu)\} \cup \{\text{conv}(v^1, v^3) + \text{cone}(r^1)\}$. In light blue, $\text{cl conv}(\mathcal{X})$, while in dark blue its inner approximation $\mathcal{W} = \text{conv}(v^1, v^2, v^3)$ at a given iteration of the *ESO*.

Normalizations and termination. The normalizations of the *PRLP* in (5.12) are decisive in practice, since they affect the algorithm's overall stability (and convergence) through the generated cutting planes. Often, normalizations dramatically affect the generators' performance [16, 55, 71, 122]. Hence, normalize (5.12) with (5.12d). Also, for any new ray r added in Step 11, we require $\|r\|_2 = 1$. In practice, we observed that such precautions often produce reasonably sparse and numerically stable cuts. Finally, we show this *ESO* implementation terminates in a finite number of steps with Proposition 4.

Proposition 4. *The ESO terminates in a finite number of steps whenever $\text{cl conv}(\mathcal{X})$ is a polyhedron.*

Proof. The *ESO* inner approximate $\text{cl conv}(\mathcal{X})$ with its $\mathcal{V}$ -representation, which is made finitely many rays and vertices for any given polyhedron. Hence, we have to prove that the *ESO* will never find a vertex v_k (ray r_j) in Step 14 (Step 11) so that v_k was already in V (r_j was already in R) in the previous Step 5. This will necessarily imply that the loop in Step 4 terminates. The inequality in Step 8 is valid for $\mathcal{W}$ if and only if $\bar{\pi}^\top v_k \leq \bar{\pi}_0$ for any $v_k \in V$, and $\bar{\pi}^\top r_j \leq 0$ for any $r_j \in R$ as of (5.12b) and (5.12c). Also, it is a separating hyperplane between $\mathcal{W}$ and $\bar{x}$, thus $\bar{\pi}^\top \bar{x} > \bar{\pi}_0$. Yet, it may not necessarily be a valid inequality for any element in $\text{ext}(\text{cl conv}(\mathcal{X}))$ and $\text{rec}(\text{cl conv}(\mathcal{X}))$. Consider now $\mathcal{G}$ in Step 9. On the one hand, if $\mathcal{G}$ is bounded, let ν be its optimal solution. Then, either (i.) $\bar{\pi}^\top \nu < \bar{\pi}^\top \bar{x}$ with $\bar{\pi}^\top x \leq \bar{\pi}^\top \nu$

being a separating hyperplane between $\text{cl conv}(\mathcal{X})$ and $\bar{x}$. Thus the algorithm terminates and returns **no**, or (ii.) $\bar{\pi}^\top \nu \geq \bar{\pi}^\top \bar{x}$, and ν is necessarily a vertex of $\text{ext}(\text{cl conv}(\mathcal{X})) \setminus V$ violating $\bar{\pi}^\top x \leq \bar{\pi}_0$. Then, the algorithm updates $V \leftarrow V \cup \{\nu\}$. On the other hand, if $\mathcal{G}$ is unbounded, then there exists a extreme ray r so that $\bar{\pi}^\top r > 0$. Then, ν is necessarily in $\text{rec}(\text{cl conv}(\mathcal{X})) \setminus R$, and $\bar{\pi}^\top \nu > \bar{\pi}_0$. The algorithm updates $R \leftarrow R \cup \{r\}$ and go back to Step 5. Since there are finitely many rays and vertices, the algorithm will then terminate. $\square$

Eliminating the conic coefficients. Given a **yes** and (V, R, α, β) from Algorithm 4, one may need to determine an equivalent proof of inclusion where the set R is void. When using the *CnP*, this is equivalent to finding an explicit description of a mixed-strategy $\tilde{\sigma}^i$ as a convex combination of pure strategies in $\mathcal{X}^i$, thus determining $\text{supp}(\tilde{\sigma}^i)$. Proposition 5 proves we can always convert a proof (V, R, α, β) to a proof $(\tilde{V}, \tilde{\alpha})$.

Proposition 5. *Let $\bar{x}$ and $\mathcal{X}$ be the inputs of Algorithm 4, and $\text{cl conv}(\mathcal{X})$ be a polyhedron. Assuming the algorithm returns **yes** and (V, R, α, β) , one can always convert the proof of inclusion (V, R, α, β) to an equivalent one $(\tilde{V}, \tilde{\alpha})$, where – for any element $v \in \tilde{V}$ – $v \in \mathcal{X}$.*

Proof. Let $S_v = \text{ext}(\text{cl conv}(\mathcal{X}))$ and $S_r = \text{rec}(\text{cl conv}(\mathcal{X}))$ be the set of extreme points and extreme rays for $\text{cl conv}(\mathcal{X})$, respectively. Without loss of generality, we restrict to the case where $\bar{x} = v^* + \lambda^* r^*$ and $r^* \in V$ and $r^* \in R$, namely when $|V| = |R| = 1$. The proof naturally generalizes when $|V| > 1$ or $|R| > 1$. We will write $\bar{x}$ as the limit of a set of points resulting from convex combinations of points in $\mathcal{X}$. Since $r^* \in S_r$, it is also an extreme direction of a given cone of C so that there exists a set $B \subseteq \mathcal{X}$ and $\text{conv}(B) = C$. Let $\bar{v} \in C$ be an arbitrary point in C . We define the new point $\bar{x}_\epsilon = \bar{x} - (v^* - \bar{v})\epsilon$. By definition, we have that $\lim_{\epsilon \rightarrow 0} \bar{x}_\epsilon = \bar{x}$. We can now rewrite $\bar{x}_\epsilon$ as a convex combination of v^* and a point $\bar{v} + \lambda_\epsilon r^*$, where the latter belongs to the cone C for any $\lambda_\epsilon \geq 0$.

$$\bar{x}_\epsilon = \eta v^* + (1 - \eta)(\bar{v} + \lambda_\epsilon r^*) \quad \eta \in [0, 1] \quad (5.13)$$

By substituting the definition of $\bar{x}_\epsilon$ in (5.13) we have that

$$\bar{x} - \epsilon v^* + \epsilon \bar{v} = \eta v^* + (1 - \eta)(\bar{v} + \lambda_\epsilon r^*), \quad (5.14)$$

and by plugging the definition of $\bar{x}$ in (5.14) we have that

$$v^* + \lambda^* r^* - \epsilon v^* + \epsilon \bar{v} = \eta v^* + (1 - \eta)(\bar{v} + \lambda_\epsilon r^*) \Rightarrow \quad (5.15)$$

$$v^*(1 - \epsilon) + \epsilon(\bar{v} + \frac{\lambda^*}{\epsilon} r^*) = \eta v^* + (1 - \eta)(\bar{v} + \lambda_\epsilon r^*). \quad (5.16)$$

The latter equality holds when $\eta = 1 - \epsilon$ and $\lambda_\epsilon = \frac{\lambda^*}{\epsilon}$. $\square$

5.5 Tailoring the *CnP*

Although both Algorithm 3 and Algorithm 4 are correct and generalize for any polyhedrally-representable *RBG*, one may exploit the special structure of the players' feasible sets to improve the algorithms. We present two applications to *NASPs* and *IPGs*.

5.5.1 *CnP* for *NASPs*

In *NASPs*, we reformulate each bilevel program by incorporating the lower-level follower problems as complementarity constraints. Since the followers' problems are convex-quadratic, this step translates in writing the *KKT* conditions associated with each followers' convex quadratic program. Such constraints are non-linear and are in the form of complementarity equations. Thus, each of the reformulated problems is a set of the linear leader constraints and the followers' complementarity conditions as in (5.5). Since the reformulated leaders' problems include the followers' ones, we refer to these reformulated bilevel programs as the players of this game. In Algorithm 3, we start by omitting *all* the complementarity constraints in the initial relaxation, vastly enlarging the action space for every player. Thus, the relaxation in Step 1 is the polyhedral relaxation $\tilde{\mathcal{X}}_0^i = O_0^i$, for each player i . Since the description of (5.5) only needs a finite number of complementarity conditions in $\mathcal{C}^i$, the algorithm should track the number of included complementarities at any iteration t via a set J_t^i . Then, Step 15 will include in the next $\tilde{\mathcal{X}}_{t+1}^i$ set one (or more) complementarity j for some player i so that $j \notin J_t^i$ and $j \in \mathcal{C}^i$. Thus, the branching step in Step 15 accounts to finding (one or more) leader i for which $J_t^i \neq \emptyset$, and include a complementarity $j \in J_t^i$ in its refined $\tilde{\mathcal{X}}_{t+1}^i$ set. This boils down to the computation of $\tilde{\mathcal{X}}_{t+1}^i$ as the union of a two-sided disjunction $\{\tilde{\mathcal{X}}_t^i \cap x_{j^i}^i \leq 0\}$ and $\{\tilde{\mathcal{X}}_t^i \cap z_{j^i}^i \leq 0\}$, where $x_{j^i}^i, z_{j^i}^i$ are the terms involved in the j -th complementarity of i . If the algorithm ends at step t with $J_t^i = \mathcal{C}^i$ for any i , then we implicitly solved the game via its exact formulation (i.e., we obtained $\text{cl conv}(\mathcal{X}^i)$ for any i). In practice, one runs Algorithm 3 with a finite time limit and expects the algorithm to exhibit a reliable converging behavior towards a solution. Two decisive steps of Algorithm 3 are the *branching* in Step 15, and the procedure **EquilibriumLCP** in Step 4. Within *MIP*, the “dark side” heuristic nature of some decisions plays a pivotal role in solvers [107]. With the same spirit, we elaborate an additional ingredient to the algorithm.

Branching step. We focus on the branching task in Step 15. We devised two simple branching rules working whenever there exists an *MNE* $\tilde{\sigma}$ in Step 4 at iteration t . Given

a generic player i , its branching candidates at t are in the set $\mathcal{C}^i \setminus J_t^i$, which we assume to contain strictly more than one element.

- (i.) *Hybrid branching:* for any candidate j in this set, we optimize $\min_{\lambda^i} \{(\lambda^i)^2 : \tilde{\sigma}^i \in \text{cl conv}(\{\tilde{\mathcal{X}}_t^i \cap x_{j^i}^i \leq 0\} \cup \{\tilde{\mathcal{X}}_t^i \cap z_{j^i}^i \leq 0\})\}$, where λ are the constraint violations, whose sum of squares we minimize. We select the complementarity with the largest constraint violation among the candidates.
- (ii.) *Deviation branching:* we compute the best-response for i to $\tilde{\sigma}^{-i}$ by solving the bilevel problem of i . We then select the first (given an arbitrary order) complementarity j necessary to encode part of the polyhedron containing such best-response.

5.5.2 *CnP* for *IPGs*

We tailor Algorithm 3 by exploiting some standard techniques of *MIP*. We present four ideas regarding tailoring.

- (i.) Since each player's objective direction changes according to the other players' decisions, we speculate the number of cuts the *CnP* uses may grow faster than it would in a single *MIP*. Since $\text{cl conv}(\mathcal{X}^i)$ is the perfect formulation, any family of valid inequalities for an integer program is also valid for the each player's (parametrized) integer program. For instance, in our tests, we use Gomory Mixed-Integer (*GMI*s), Mixed-Integer Rounding (*MIR*s), and Knapsack Cover (*KP*s) Cuts.
- (ii.) At step $t = 0$ of the *CnP* – instead of taking the linear relaxation of any $\mathcal{X}^i$ – one can already start from a strengthened version of these programs. In our tests, we strengthen the coefficients in each A^i matrix and add some valid inequalities (e.g., *root node* cuts). Dual presolve routines and fixed-costs fixing are generally discouraged since the objective function is parametrized in x^{-i} .
- (iii.) If the *ESO* in Step 7 returns **no**, one can always add additional valid inequalities to strengthen further the next $\tilde{\mathcal{X}}_{t+1}^i$ set. Furthermore, some value cuts may not exhibit a well-behaved numerical behavior (e.g., long fractional coefficients). One can always separate a valid inequality that cuts off the incumbent solution (for the given player i) and avoid using the value cut.
- (iv.) The branching step in Step 15 triggers only when at least one of the players in $\tilde{G}$ has an unbounded feasible set $\tilde{\mathcal{X}}^i$. One can always find a disjunction on one or more integer variables for the players and add it to the next $\tilde{\mathcal{X}}_{t+1}^i$ set.

5.6 Computational experiments

In this section, we present some implementation considerations and the numerical results from our tests. We use an *Intel Xeon Gold 6142*, with 128GB of RAM and *Gurobi 9.2* as *MIP* (blackbox) solver, and *PATH* [56, 70] as an alternative *LCPs* solver in the *IPGs* tests. The entirety of the time-related results is reported with a shifted geometric mean (shift of 10 seconds) to shield against outlier values. The implementation code is in [58]. Although the *CnP* is quite generic, it manages to outperform game-specific methods on both *IPGs* and *NASPs*.

NASPs tests. We configure the *ESO* (Algorithm 4) with $\epsilon = 10^{-5}$, a *CnP* timelimit of 300 seconds, and we use up to 12 cores for the *CnP* and the baseline. The testbed is the set of instances *InstanceSet B* [31] (each instance has 7 Stackelberg leaders with up to 3 followers each), and we introduce an even harder *InstanceSet H7* (7 leaders up with 7 followers each). As a baseline, we use the *inner approximation* (*Inn*), a problem-specific algorithm proposed in [31]. In Table 5.1 we provide 3 different (geometric) means for the computing times in the *GT(s)* columns. The first column is the type of algorithm: the baseline (*Inn-S*) (see [31]), or the *CnP* (*Out*), with either the deviation branching (*DB*) or the hybrid one (*HB*). The second and third columns are the instance set (*Inst*) and its cardinality ($\#$), respectively. The three subsequent pairs of columns report the computing time and the number of instances for which the algorithm either (i.) found an *MNE* (*EQ*), (ii.) concluded no *MNE* exists (*NO_EQ*), or (iii.) terminated without numerical issues (*ALL*). The last two columns report the number of numerical issues ($\#NI$) and time limits hits ($\#TL$). Large *NASP* instances, in particular the set *H7*, are generally badly scaled and are thus useful to test the numerical stability of the algorithms. A clear pattern in Table 5.1 is the systemic failure of *Inn-S-1* on the set *H7*, where the algorithm fails due to the size of the descriptions of $\text{clconv}(\mathcal{X}^i)$. *Inn* tends to exhibit significant numerical issues. On the contrary, the *CnP* performs consistently, especially in the hard set *H7*.

5.6.1 IPGs tests

We configure the *ESO* with $\epsilon = 3 \times 10^{-5}$, and the *CnP* timelimit being 300 seconds. The baseline is the *mSGM* algorithm proposed in [32], which also provides the instances of the *Random Knapsack Game*.

Instances. We use the *Random Knapsack Game* instances from Carvalho et al. [32], where each player i solves a knapsack problem with m items as in (5.17). The objects v^i, w^i are

Table 5.1 *NASPs* summary results.

Algo	Inst	#	GT (s)	#	GT (s)	#	GT (s)	#N	#NI	#TL
			NASH_EQ		NO_EQ		ALL			
Inn-S-1	B	50	6.22	49	69.76	1	6.56	50	0	0
Inn-S-3	B	50	4.94	49	23.96	1	5.12	50	0	0
Out-HB	B	50	7.47	46	29.37	1	7.71	47	3	0
Out-DB	B	50	9.45	46	11.81	1	9.50	47	3	0
Inn-S-1	H7	50	-	0	-	0	300.00	46	4	46
Inn-S-3	H7	50	-	0	-	0	-	0	50	0
Out-HB	H7	50	53.79	41	-	0	73.45	50	0	9
Out-DB	H7	50	52.58	35	-	0	88.92	50	0	15

integer vectors corresponding to the profits and weights. Also, W^i is the knapsack capacity, while C^i is a diagonal $m \times (n-1)m$ matrix with integer entries. The elements on the diagonal are the interaction coefficients associated with each of the $(n-1)$ other players in the game and their m decision variables (in the lexicographic order given by each player's index). We remark that all such parameters are integer-valued, yet, they are not required to be positive. In this game, each player i solves a knapsack problem where its profits v^i may be decreased or increased by the interactions given in C^i . Since this latter is a diagonal matrix, players are mutually interacting only for corresponding items, for instance x_j^i interacts with any x_j^{-i} for $j \in \{1, \dots, m\}$. A positive interaction coefficient between x_j^i and x_j^p in C^i – where p is another player – boils down to a positive incentive for i when both x_j^i and x_j^p are equal to 1, namely both players are selecting the item. Analogously, if the interaction coefficient is negative, i may be penalized in picking j if also p picks it. Formally, the model reads as

$$\max_{x^i} (v^i)^\top x^i + (x^{-i})^\top C^i x^i \quad (5.17a)$$

$$\text{s.t.} \quad w^i x^i \leq W^i, x^i \in \{0, 1\}^m. \quad (5.17b)$$

Since each *PAG* has an *MNE* ($\mathcal{X}^i$ is finite for any i), the *CnP* never branches but adds *MIP* inequalities. Namely, the *CnP* purely acts as a cutting plane algorithm without ever branching.

Setup. We possibly strengthen each player's $\tilde{\mathcal{X}}^i$ with multiple rounds of *GMI*s, *MIR*s, and *KPs* cuts (using [39]), and add other valid inequalities whenever the *ESO* returns **no**. We aim to show *MIP* cuts are a pivotal ingredient of the *CnP*, and their integration improves the algorithm's performance. We test 4 levels of *MIP* cuts *aggressiveness*: -1 with no *MIP* cuts, and $0, 1, 2$ for more cutting planes at each iteration. We solve the *LCP*s with either:

(i.) *PATH*, thus computing a feasible *MNE*, or (ii.) *Gurobi*, optimizing the quadratic *social welfare* (*SW*), namely the players' payoffs sum.

Results. In Table 5.3 we present the computational results for our experiments, with Table 5.4 being a similar table with percent changes concerning the baseline of *m-SGM*. The first column is the algorithm's type as defined previously. For the *CnP*, we also report which solver handled the *EquilibriumLCP*. The second column (*O*) is the objective type, either *F* for feasibility (e.g., an *MNE*) or *Q* for the quadratic objective. In particular, we use a quadratic objective given by the sum of all players' payoff, namely the so-called social-welfare (*SW*). The third column (*C*) reports the aggressiveness of the additional *MIP* cutting planes generated. The column is set to -1 if we add no additional *MIP* cuts, and 0 if we add such cuts whenever a value cut was numerically unstable (e.g., we could not transform the coefficients to integer, or the coefficients ranged from 10^{-3} to 10^3). Otherwise, 1 is the higher increasing levels of aggressiveness, namely where the *CnP* adds (multiple) *MIP* cuts even when *ESO* cuts were added. For each set of instances, we report the number of players n , and the number of items in each knapsack m . The fourth column reports the computing times geometric mean (*GeoT*) and the fifth column ($\#F$) reports the number of time-limit hits for the associated algorithm. The remaining columns are mean average results for a series of statistics. Specifically, in the sixth and seventh columns, we report the average social welfare (SW^*) and the average number of iterations ($\#IT^*$), respectively. The last four columns are the average numbers of: (i.) all cuts added ($Cuts^*$), (ii.) The cuts added by the *ESO*, namely the $\mathcal{V}$ -polyhedral cuts (VP^*) in Step 13 of Algorithm 4, (iii.) value cuts added by the *ESO* (VC^*) in Step 2 of Algorithm 4, (iv.) generic *MIP* cuts added (MIP^*).

Highlights. A first clear pattern is an increase in the *SW* – in almost any instance set – with the *CnP* algorithm. The *m-SGM* algorithm does find only an equilibrium, and so does the *CnP* with *PATH*. Since the *CnP* exploits game's relaxations, one can expect it will find a possibly more favorable *MNE*. Whenever a *MIP* solver optimizes the welfare function, there are dramatic improvements in the welfare. However, this may come at a cost in terms of computing times, as highlighted in Table 5.4. In general, *MIP* solvers do not exploit the underlying structure of *LCPs*, and we speculate this may cause such computing time increases. In general, with *PATH* there are significant computing time improvements in all of the instance sets except $n = 2, m = 20$. Furthermore, the more the cuts, the fewer iterations of the *CnP* are required to converge to a feasible *MNE*. This seems to be the case for all the instance sets. Interestingly, a greater aggressiveness of cuts tends to reduce the number of $\mathcal{V}$ -polyhedral and value cuts in favor of more *MIP* cuts in almost all the instance sets.

A plausible explanation is that the *MIP* cuts are generally generated by routines that are more likely to result in *stronger* cuts for the mixed-integer hull than the *ESO* cuts. Finally, whenever *MIP* cuts were completely disabled (*C* columns set to -1) we generally observe an increase in computation time and average numbers of iterations.

Table 5.2 *IPGs* summary results.

Algo	C	t (s)	MM	T	SW%	OSW	#It	Cuts	MIP
m-SGM		0.73	21.43	0	0.0%	27.0%	8.43	-	-
CnP-MIP	Q/-1	6.58	287.52	0	13.5%	37.0%	7.80	9.57	0.00
CnP-MIP	Q/0	6.13	287.01	0	12.9%	37.0%	5.73	6.47	2.30
CnP-MIP	Q/1	6.31	287.52	0	13.3%	37.0%	3.50	9.60	7.47
CnP-PATH	F/-1	0.36	10.54	0	1.8%	27.0%	7.60	10.2	0.00
CnP-PATH	F/0	0.05	0.19	0	2.9%	27.0%	5.27	5.90	2.07
CnP-PATH	F/1	0.04	0.19	0	4.9%	27.0%	3.23	8.87	7.10
m-SGM		20.86	300.00	6	0.0%	25.0%	18.58	-	-
CnP-MIP	Q/-1	61.08	294.50	0	22.5%	40.0%	13.70	17.00	0.00
CnP-MIP	Q/0	57.85	299.45	1	19.6%	40.0%	11.62	12.62	3.45
CnP-MIP	Q/1	68.20	299.04	0	22.3%	38.0%	9.48	16.80	10.32
CnP-PATH	F/-1	6.68	80.89	0	15.7%	28.0%	13.55	16.35	0.00
CnP-PATH	F/0	4.48	74.37	0	15.7%	28.0%	9.62	10.25	2.42
CnP-PATH	F/1	4.32	75.88	0	15.9%	28.0%	8.22	14.35	8.43

Finally, Table 5.2 provides an overview of the results by splitting the instances: the small instances (with $mn \leq 80$) in rows 2 – 8, and the large ones (with $mn > 80$) in rows (9 – 15). The first two columns are the algorithm’s name, and the objective type and the *MIP* cut aggressiveness as above. In column order, we report: the geometric mean time ($t(s)$), the difference among the maximum and the minimum of time (*MM*), the number of time limits (*T*), and the improvement in social welfare with respect to *m-SGM* (*SW%*). We provide the percentage of instances (*OSW*) where the *MNE*’s *SW* is at least 80% the optimal social outcome (e.g., a solution, possibly not an *MNE*, where a planner independently decides to maximize the *SW*). In the last three columns, we report the average number of iterations ($\#IT$), the total number of cuts (*Cuts*), and the separate number for *MIP* cuts (*MIP*).

Table 5.3 *IPGs* results in absolute values.

Algo	O	C	GeoT (s)	#F	SW*	#It*	Cuts*	VP*	VC*	MIP*
n=3 m=10										
m-SGM	-	-	2.11	0	632.99	10.00	-	-	-	-
CnP-MIP	Q	-1	0.47 (0.23)	0	812.48	4.50	5.0	2.0	3.0	0.0
CnP-MIP	Q	0	0.31 (0.14)	0	812.98	4.60	4.8	2.0	1.1	1.7
CnP-MIP	Q	1	0.20 (0.08)	0	820.71	2.60	7.2	0.5	1.1	5.6
CnP-PATH	F	-1	0.02	0	706.66	5.00	5.9	2.0	3.9	0.0
CnP-PATH	F	0	0.02	0	718.13	4.50	4.9	2.0	1.5	1.4
CnP-PATH	F	1	0.03	0	742.87	2.00	5.4	0.3	0.7	4.4
n=2 m=20										
m-SGM	-	-	0.01	0	658.31	5.40	-	-	-	-
CnP-MIP	Q	-1	0.96 (0.25)	0	684.19	6.40	6.3	4.4	1.9	0.0
CnP-MIP	Q	0	0.93 (0.29)	0	683.91	6.10	5.9	3.0	1.2	1.7
CnP-MIP	Q	1	0.75 (0.18)	0	682.69	3.70	7.6	1.4	0.9	5.3
CnP-PATH	F	-1	0.05	0	645.44	5.30	5.5	3.1	2.4	0.0
CnP-PATH	F	0	0.04	0	664.44	4.90	4.7	1.8	1.2	1.7
CnP-PATH	F	1	0.03	0	656.44	3.10	6.2	1.2	0.4	4.6
n=3 m=20										
m-SGM	-	-	0.20	0	1339.98	9.90	-	-	-	-
CnP-MIP	Q	-1	29.74 (1.49)	0	1488.96	12.50	17.4	7.0	10.4	0.0
CnP-MIP	Q	0	27.22 (0.66)	0	1473.46	6.50	8.7	4.0	1.2	3.5
CnP-MIP	Q	1	29.61 (0.61)	0	1476.85	4.20	14.0	2.0	0.5	11.5
CnP-PATH	F	-1	1.04	0	1327.47	12.50	19.2	6.3	12.9	0.0
CnP-PATH	F	0	0.08	0	1325.23	6.40	8.1	3.4	1.6	3.1
CnP-PATH	F	1	0.07	0	1361.74	4.60	15.0	2.2	0.5	12.3
n=2 m=40										
m-SGM	-	-	1.26	0	1348.56	13.70	-	-	-	-
CnP-MIP	Q	-1	27.87 (5.11)	0	1433.13	16.70	21.9	11.1	10.8	0.0
CnP-MIP	Q	0	25.58 (3.53)	0	1434.09	12.80	13.4	8.2	1.1	4.1
CnP-MIP	Q	1	29.72 (2.16)	0	1405.30	10.50	18.7	6.4	0.7	11.6
CnP-PATH	F	-1	0.89	0	1355.26	16.80	20.7	9.5	11.2	0.0
CnP-PATH	F	0	0.70	0	1355.01	10.00	9.9	7.1	0.8	2.0
CnP-PATH	F	1	0.62	0	1355.21	7.80	14.1	5.1	0.3	8.7
n=3 m=40										
m-SGM	-	-	27.04	2	2339.79	20.10	-	-	-	-
CnP-MIP	Q	-1	140.33 (5.49)	0	2991.76	20.20	28.5	13.2	15.3	0.0
CnP-MIP	Q	0	128.74 (3.06)	0	3016.22	11.60	15.6	8.9	1.9	4.8
CnP-MIP	Q	1	162.20 (2.58)	0	2980.69	9.30	21.9	6.7	0.9	14.3
CnP-PATH	F	-1	2.35	0	2882.45	17.60	24.9	12.6	12.3	0.0
CnP-PATH	F	0	0.87	0	2906.33	10.80	14.0	8.8	1.4	3.8
CnP-PATH	F	1	0.79	0	2898.04	9.00	21.1	6.6	0.8	13.7
n=2 m=80										
m-SGM	-	-	14.97	1	2676.52	19.40	-	-	-	-
CnP-MIP	Q	-1	29.83 (11.47)	0	3127.96	7.60	6.7	5.4	1.3	0.0
CnP-MIP	Q	0	27.02 (7.27)	0	3127.97	7.80	7.0	5.3	0.7	1.0
CnP-MIP	Q	1	36.71 (10.06)	0	3124.63	6.10	8.6	3.6	0.5	4.5
CnP-PATH	F	-1	7.71	0	2914.36	8.80	8.1	6.7	1.4	0.0
CnP-PATH	F	0	5.45	0	2926.82	7.00	6.1	4.5	0.4	1.2
CnP-PATH	F	1	4.93	0	2936.52	5.80	7.4	3.4	0.4	3.6
n=2 m=100										
m-SGM	-	-	77.13	3	2861.20	21.10	-	-	-	-
CnP-MIP	Q	-1	102.57 (36.29)	0	3750.38	10.30	10.9	7.4	3.5	0.0
CnP-MIP	Q	0	105.97 (33.07)	1	3454.41	14.30	14.5	9.4	1.2	3.9
CnP-MIP	Q	1	107.04 (30.86)	0	3771.62	12.00	18.0	6.3	0.8	10.9
CnP-PATH	F	-1	23.02	0	3496.86	11.22	11.67	8.33	3.33	0.0
CnP-PATH	F	0	14.46	0	3488.44	10.70	11.0	7.1	1.2	2.7
CnP-PATH	F	1	14.56	0	3507.71	10.30	14.8	6.4	0.7	7.7

Table 5.4 *IPGs* results (in percentage) with respect to the *m-SGM*. For *GeoT(s)*, and *#It*, the lower the better. As of *SW**, the higher the better.

Algo	O	C	GeoT (s)	#F	SW*	#It*	Cuts*	VP*	VC*	MIP*
m=3 n=10										
m-SGM	-	-	0.00%	0	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	-77.68%	0	28.35%	-55.00%	5.0	2.0	3.0	0.0
CnP-MIP	Q	0	-85.45%	0	28.44%	-54.00%	4.8	2.0	1.1	1.7
CnP-MIP	Q	1	-90.77%	0	29.66%	-74.00%	7.2	0.5	1.1	5.6
CnP-PATH	F	-1	-98.89%	0	11.64%	-50.00%	5.9	2.0	3.9	0.0
CnP-PATH	F	0	-99.18%	0	13.45%	-55.00%	4.9	2.0	1.5	1.4
CnP-PATH	F	1	-98.73%	0	17.36%	-80.00%	5.4	0.3	0.7	4.4
m=2 n=20										
m-SGM	-	-	0.00%	0	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	7247.90%	0	3.93%	18.52%	6.3	4.4	1.9	0.0
CnP-MIP	Q	0	7084.17%	0	3.89%	12.96%	5.9	3.0	1.2	1.7
CnP-MIP	Q	1	5634.50%	0	3.70%	-31.48%	7.6	1.4	0.9	5.3
CnP-PATH	F	-1	268.26%	0	-1.95%	-1.85%	5.5	3.1	2.4	0.0
CnP-PATH	F	0	214.60%	0	0.93%	-9.26%	4.7	1.8	1.2	1.7
CnP-PATH	F	1	149.61%	0	-0.28%	-42.59%	6.2	1.2	0.4	4.6
m=3 n=20										
m-SGM	-	-	0.00%	0	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	14958.58%	0	11.12%	26.26%	17.4	7.0	10.4	0.0
CnP-MIP	Q	0	13681.22%	0	9.96%	-34.34%	8.7	4.0	1.2	3.5
CnP-MIP	Q	1	14891.54%	0	10.21%	-57.58%	14.0	2.0	0.5	11.5
CnP-PATH	F	-1	424.87%	0	-0.93%	26.26%	19.2	6.3	12.9	0.0
CnP-PATH	F	0	-57.18%	0	-1.10%	-35.35%	8.1	3.4	1.6	3.1
CnP-PATH	F	1	-63.36%	0	1.62%	-53.54%	15.0	2.2	0.5	12.3
m=2 n=40										
m-SGM	-	-	0.00%	0	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	2111.71%	0	6.27%	21.90%	21.9	11.1	10.8	0.0
CnP-MIP	Q	0	1929.94%	0	6.34%	-6.57%	13.4	8.2	1.1	4.1
CnP-MIP	Q	1	2258.44%	0	4.21%	-23.36%	18.7	6.4	0.7	11.6
CnP-PATH	F	-1	-29.26%	0	0.50%	22.63%	20.7	9.5	11.2	0.0
CnP-PATH	F	0	-44.36%	0	0.48%	-27.01%	9.9	7.1	0.8	2.0
CnP-PATH	F	1	-50.86%	0	0.49%	-43.07%	14.1	5.1	0.3	8.7
m=3 n=40										
m-SGM	-	-	0.00%	2	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	418.88%	0	27.86%	0.50%	28.5	13.2	15.3	0.0
CnP-MIP	Q	0	376.04%	0	28.91%	-42.29%	15.6	8.9	1.9	4.8
CnP-MIP	Q	1	499.77%	0	27.39%	-53.73%	21.9	6.7	0.9	14.3
CnP-PATH	F	-1	-91.31%	0	23.19%	-12.44%	24.9	12.6	12.3	0.0
CnP-PATH	F	0	-96.78%	0	24.21%	-46.27%	14.0	8.8	1.4	3.8
CnP-PATH	F	1	-97.07%	0	23.86%	-55.22%	21.1	6.6	0.8	13.7
m=2 n=80										
m-SGM	-	-	0.00%	1	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	99.30%	0	16.87%	-60.82%	6.7	5.4	1.3	0.0
CnP-MIP	Q	0	80.53%	0	16.87%	-59.79%	7.0	5.3	0.7	1.0
CnP-MIP	Q	1	145.29%	0	16.74%	-68.56%	8.6	3.6	0.5	4.5
CnP-PATH	F	-1	-48.49%	0	8.89%	-54.64%	8.1	6.7	1.4	0.0
CnP-PATH	F	0	-63.56%	0	9.35%	-63.92%	6.1	4.5	0.4	1.2
CnP-PATH	F	1	-67.08%	0	9.71%	-70.10%	7.4	3.4	0.4	3.6
m=2 n=100										
m-SGM	-	-	0.00%	3	0.00%	0.00%	-	-	-	-
CnP-MIP	Q	-1	32.99%	0	31.08%	-51.18%	10.9	7.4	3.5	0.0
CnP-MIP	Q	0	37.40%	1	20.73%	-32.23%	14.5	9.4	1.2	3.9
CnP-MIP	Q	1	38.79%	0	31.82%	-43.13%	18.0	6.3	0.8	10.9
CnP-PATH	F	-1	-70.16%	0	22.22%	-46.81%	11.67	8.33	3.33	0.0
CnP-PATH	F	0	-81.25%	0	21.92%	-49.29%	11.0	7.1	1.2	2.7
CnP-PATH	F	1	-81.12%	0	22.60%	-51.18%	14.8	6.4	0.7	7.7

5.7 Conclusions

The defining boundaries of this work are game theory and mathematical programming and how they can effectively interact to improve the theory and practice of equilibria computation. We strongly believe the joint endeavor between the disciplines can widen their theoretical understanding and practical impact equilibria computation through a Branch and Cut algorithm. We employ the concept of game’s approximation and extend it to a game by building an increasingly tight sequence of approximations that eventually lead to an equilibrium (or a non-existence proof). The amusing element of this approach is its interoperability with standard mathematical programming tools, such as – and not restrictively – valid inequalities, relaxations, disjunctions. Our approach does not necessarily exploit the specific structure of any of the players’ problems, nor the game, and is in this sense generic. However, it stands on the shoulder of the many “giants” ideas – theoretical and practical – that mathematical programming offers. The *CnP*– even when compared to standard sample generation algorithms or problem-specific ones – offers an appealing alternative for the efficient computation of equilibria. We do believe the results reported should not constitute a barrier. On the contrary, we prudently believe improvement opportunities lie ahead. The integration of existing mathematical programming tools may indeed significantly advance this algorithmic rationale. Moreover, and even more importantly, we hopefully foresee an increased interest from this community towards developing new tools to tackle equilibria computation. The open questions and possible extensions of this work are several. Nevertheless, we are quite optimistic about such opportunities. This paper’s ultimate goal is to showcase – in the context of *MIP*– that existing theory is of tremendous relevance for equilibria computation, also from a practical standpoint. Among the many questions that arose through the making of this work, we present four.

The polyhedral assumption. In the implementation of the *ESO* in Section 5.4.1, we heavily leveraged what we called the *polyhedral assumption* on each player $\text{clconv}(\mathcal{X}^i)$. In specific, we use this assumption to handle unboundedness through polyhedral cones. However, we believe this should not be – in theory – restrictive. An improved separation oracle could – for instance – leverage on second-order cones and drop the polyhedral assumption. For the scope of this work, we showcased *IPGs* and *NASPs*, which are archetypical forms of *RBGs*. Nevertheless, the polyhedral assumption may be dropped in favor of other well-structured sets as soon as one may guarantee a finite termination of the associated enhanced separation oracle.

Inequalities. Except value-cuts, we mainly employed $\mathcal{V}$ -polyhedral inequalities and generic *MIP* ones. However, we speculate there is room for developing new *game-theoretical* inequalities, i.e., inequalities that account for more than one player’s variables. This may well exploit some special or general structure of the game, and they are an auspicious direction of research in the *MIP* context.

Rationality. A pivotal solution concept in game theory is the one of *rationalizability*, introduced independently by Bernheim [14], Pearce [121]. The concept grounds in two main assumptions: (i.) each player views its opponents’ choices as uncertain events, and thus probabilistically assesses them (ii.) the players are *individually* rational, or in the context of this paper they seek to optimize their payoff (objective function) as much as they can. Whenever player i has to decide which strategy to pick, it faces uncertainty about other players’ choices. Thus, the choices of i must somehow reflect some beliefs the player has concerning the strategies of its opponents. For instance, i may rule out the possibility of playing a strategy that is never the best-response to any opponents’ strategy profile. However, i shall play strategies that are best-responses to some opponents’ strategies and verify the validity of these latter ones. Namely, i should also assess the strategies played by its opponents are best-responses. In other words, we define a strategy as rationalizable only if it is a best-response to some opponents’ beliefs, which in their turn are best-responses to some other opponents’ beliefs, and so on.

In practice, a strategy is rationalizable for i if it can be rationalized with a sequence of rationalizable behaviors of the opponents of i . In this sense, i holds a *belief* on the other players’ strategies. namely, it associates a subjective probability distribution to its opponents’ strategies. *MNEs* themselves are rationalizable strategies were also the beliefs player i associates to its opponents are rationalizable and exact. We turn our attention to the *ESO*, and without loss of generality, we consider a generic player i and – for the sake of explanation – we will assume $\mathcal{X}^i$ is bounded (although this assumption is not restrictive). As previously mentioned, the *ESO* builds an increasingly accurate description $\mathcal{W}^i$ of $\text{cl conv}(\mathcal{X}^i)$ by storing the extreme points $\text{ext}(\text{cl conv}(\mathcal{X}^i))$ in V^i . These points correspond to some best-responses for i given its beliefs about the other players. One may wonder whether the strategies in V^i are rationalizable or not. The answer is generally negative. Consider an *MNE* $\tilde{\sigma}$ in Algorithm 3, and the related call to the *ESO* in Step 7 for player i . The best-responses in V^i may not be rationalizable, disregarding the fact that the EO sampled them through the solution of a series of parametrized mathematical programs for i . The issue here concerns the values of the parameters $\tilde{\sigma}^{-i}$ of such programs, which may not be themselves rationalizable. Although the best-response computation for i returns a feasible pure-strategy, the $\tilde{\sigma}^{-i}$ beliefs

plugged in as parameters may themselves not be feasible mixed-strategies. In turn, at each iteration, t of Algorithm 3, the strategies in support of $\tilde{\sigma}$ are rationalizable for the relaxation $\tilde{G}$. Can we design specific routines to “select” rationalizable supports in an efficient way? This question is somehow related to the works of Carvalho et al. [32], Porter et al. [124], which in fact propose algorithms to sample rationalizable best-responses. These two approaches seek to achieve *feasibility* (e.g., an *MNE*) by constructing a smaller approximation of the original games where only a few strategies are present. Dichotomically, the *CnP* tries to refine a loose approximation of the original game. In this context, better identification of best-responses may lead to faster convergence and the development of new cutting planes based on assumptions of rationality, somehow connecting to the first idea on inequalities. In this sense, another important ground would be to establish what “valid” means in the context of games and inequalities.

Tree and bounds. There is a clear connection between *CnP* and the Branch and Cut algorithm of Padberg and Rinaldi [120]. Instead of solving a linear program – which is a $\mathcal{P}$ -problem at least in theory –, the *CnP* leverages a series of *LCPs* which are well-known $\mathcal{NP}$ -hard problems. This is not surprising for either *IPGs* or *NASPs*, which are Σ_2^P -hard in practice. A natural extension of this work would be the creation of a search tree, where leaves spring from the disjunctions in Step 15 of Algorithm 3. This approach hopes to find a feasible *MNE* by possibly solving more constrained *LCPs*, and eventually enumerate equilibria. A feasible *MNE* constitutes a valid upper bound for the problems. Nevertheless, it is not clear how to use this information in the context of game relaxations. An efficient way to identify non-improving – and thus fathomed – nodes would significantly improve the capabilities of the *CnP*. The algorithm may then select the most favorable equilibrium among the possibly many in a given game. However – in contrast with the Branch and Cut – we believe such bounds relationships may be problem-specific and not general.

Acknowledgement

We would like to thank Federico Bobbio, Didier Chételat, Aleksandr Kazachkov, Rosario Scatamacchia, and Mathieu Tanneau for the insightful discussions we had concerning this work.

CHAPTER 6 ARTICLE 1: WHEN NASH MEETS STACKELBERG

Authors: Margarida Carvalho, Gabriele Dragotto, Felipe Feijoo, Andrea Lodi, and Sriram Sankaranarayanan.

Submitted to Management Science¹.

Abstract We analyze a class of simultaneous non-cooperative games among the leaders of Stackelberg Games (*NASP*) and their application in energy markets. In a *NASP*, each leader solves a linear bilevel program with quadratic convex followers subject to the standard optimistic assumption. From a complexity theory perspective, we prove it is Σ_2^P -hard to decide both if the game has a pure-strategy (*PNE*) or a mixed-strategy Nash equilibrium (*MNE*). We provide a finite algorithm that computes exact *MNEs* for *NASPs* or returns a non-existence certificate if no *MNE* exists. We enhance this algorithm with an inner approximation hierarchy that increasingly grows the description of each Stackelberg leader’s feasible region. Furthermore, we extend the algorithmic framework to retrieve a *PNE*, if one exists. Finally, we provide extensive computational tests on a range of *NASPs* instances inspired by international energy trades and a real-world study on a simplified version of the Chilean-Argentinian energy market.

6.1 Games, Definitions, and Overview

Optimization frameworks embedding Game Theory dynamics can model complex interactions among multiple agents and are powerful tools for real-world applications. Their effectiveness relies on two key ingredients. First, their modeling capabilities and the ease of interpretability of such models. Second, the efficiency of the underlying algorithmic arsenal available to solve these models. In this paper, we provide models, algorithms, and theoretical insights for a class of *non-cooperative, simultaneous games* between the leaders (i.e., the first-level players) of bilevel programs with an optimistic followers’ response. In other words, Stackelberg games’ leaders are playing a Nash game among themselves with complete information. We call such problems *Nash Games among Stackelberg Leaders (NASPs)*, schematically represented in Figure 6.1. *NASPs* are part of the well-known family of *Equilibrium Problems with Equilibrium Constraints (EPECs)* that has a wide variety of applications in energy markets. A concise representation of an elementary (or *trivial*, as more formally defined in Definition 17) *NASP* between a *Latin* Stackelberg game and a *Greek* Stackelberg game is given by

¹A pre-print is available in [31].

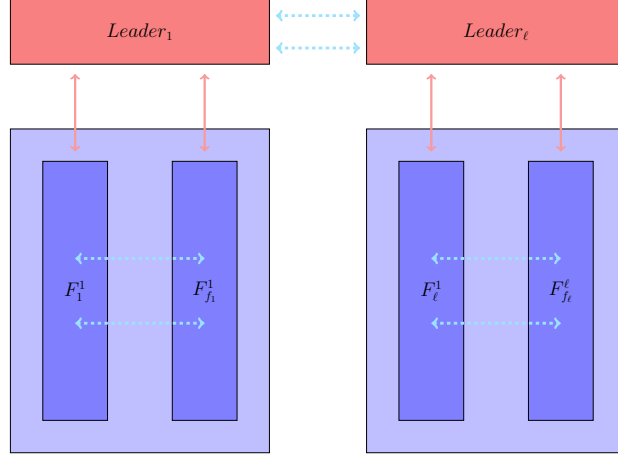


Figure 6.1 A schematic representation of a *NASP*. The vertical arrows are Stackelberg interactions (i.e, sequential decisions), while the horizontal ones are Nash interactions (i.e, simultaneous decisions).

Latin Leader

$$\min_{x,y} : c^T x + d^T y + \left(G \begin{pmatrix} \xi \\ \chi \end{pmatrix} \right)^T \begin{pmatrix} x \\ y \end{pmatrix} \quad (6.1a)$$

$$\text{subject to} \quad Ax + By \leq b \quad (6.1b)$$

$$y \in \arg \min_y \{ f^T y : Qy \leq g - Px \} \quad (6.1c)$$

Greek Leader

$$\min_{\xi,\chi} : \alpha^T \xi + \beta^T \chi + \left(\Gamma \begin{pmatrix} x \\ y \end{pmatrix} \right)^T \begin{pmatrix} \xi \\ \chi \end{pmatrix} \quad (6.1d)$$

$$\text{subject to} \quad \Phi \xi + \Psi \chi \leq \rho \quad (6.1e)$$

$$\chi \in \arg \min_{\chi} \{ \phi^T \chi : \Omega \phi \leq \gamma - \Pi \xi \}. \quad (6.1f)$$

On the one hand, the Stackelberg leaders interact through their objective functions, as in (6.1a) and (6.1d). On the other hand, within each Stackelberg game, each leader anticipates the reaction of their distinct followers, each of which solves a lower-level parametric linear program. In general, *NASPs* can (as we will later show): (i) have more than two leaders, (ii) have more than one follower per leader, (iii) have the followers of each leader interact in a Nash game, and (iv) enforce each follower to solve a convex quadratic program as opposed to linear programs like in (6.1c) and (6.1f). In Game Theory, a dominant solution concept is the one of Nash equilibrium — namely, when each player cannot profitably and unilaterally deviate from the prescribed equilibrium strategy. In this paper, we provide theory and

algorithms concerning the Nash equilibria of *NASPs*.

Applications. *NASPs*, in their full generality, could solve a wide range of problems. We outline three different potential applications related to energy, vaccines, and insurances. In this work, we are primarily motivated to model international energy markets with climate change-aware regulatory authorities and profit-maximizing domestic energy producers, and we provide a game-theoretic framework to analyze this problem. In this game, energy producers – namely the Stackelberg followers – compete in the domestic market and are usually subject to restrictions in the form of tax and caps from the regulatory authorities. The regulatory authorities – namely the Stackelberg leaders – negotiate environmental-conscious agreements for energy trade, thus engaging in a Nash game. The *NASP* theoretical abstraction models this problem and provides a general framework to analyze games, in and outside the energy domain, when there are multiple Stackelberg leaders, each with their set of followers, playing a Nash game with each other.

Similarly – yet in a different context – *NASPs* can model a complex drug trade system. For instance, at the time of writing, the COVID-19 vaccine production and trade situation pose severe threats to the world’s immunization programs and may lessen inter-country cooperation with the so-called vaccine nationalism [145]. In several cases, countries threatened and successfully blocked vaccine exports while also imposing strict regulations on indigenous producers [17, 18, 73]. In this scenario, the homogeneous good would be the vaccine, and in analogy to emission factors, we would see efficacy properties. Countries act as Stackelberg leaders, regulating vaccines’ trade and incentivizing indigenous producers (followers). Further, the leaders’ objectives could model a wide variety of tactical requirements, e.g., prioritize the production of some doses reserved for vulnerable classes, incentivize the exports of prioritized doses to neighboring countries, prioritize more effective vaccines.

Finally, as a third example, we draw attention to the insurance business. Users of a given good may be subject to disruptions (e.g., cyberattacks for energy generators) and may need to contract insurance services [34, 80]. Insurers – namely Stackelberg leaders – provide such services at a cost to their clients – or followers. The *NASP* framework extends this hierarchical model to a multi-insurer setting, introducing a mechanism of *re-insurance*. In plain English, the leaders mutually protect their insurances’ portfolios, shielding them against large-scale disruptions (e.g., natural disasters). With these motivating examples, we now detail our primary contributions.

Primary Contributions

First, we characterize the computational complexity of deciding if a given instance of *NASP* has a pure-strategy Nash equilibrium (*PNE*). Even with restrictive assumptions – such as single follower and bounded strategy sets for all players – we show it is Σ_2^P -hard to decide if the instance has a *PNE*. Namely, even with oracle access to solve $\mathcal{NP}$ -hard problems instantaneously, there would be exponentially many calls to such an oracle to decide the existence of a *PNE* for a given instance of *NASP*. In other words, without substantial consequences in complexity theory, this translates to a bound of $\Omega(2^{2^n})$ elementary operations required to solve the problem, where n is the size of the representation of the corresponding decision problem. This is quite surprising since, in most literature cases, one can either prove that all games in a considered category have a *PNE* or prove sufficiently fast that a given instance has no *PNE*. Second, we consider the computational complexity of deciding the existence of a mixed-strategy Nash equilibrium (*MNE*) for *NASPs*. We demonstrate that with exactly one follower for each leader and boundedness in every player’s problem, an *MNE* always exists (Corollary 1). However, if *at least one* of the leaders has an unbounded feasible set, it is again Σ_2^P -hard to decide the existence of an *MNE*.

Third, given these lower bounds to computationally find *PNE* or *MNE* for *NASPs*, we provide a finite-time algorithm to do so. It retrieves an *MNE* for an instance of *NASP* when it exists and provides a (double exponentially-large) proof of infeasibility when an *MNE* does not exist. To the best of our knowledge, this is the first algorithm to identify *MNE* or *PNE* for a game of this type. Fourth, we provide an enhancement to the algorithm to exclusively seek *PNEs*, or provide proof of infeasibility. This is the case of interest if mixed-strategies are not implementable in practice. Fifth, we provide another enhancement to the algorithms to find *MNEs* and *PNEs*, with an iterative inner-approximation procedure that proves to be considerably faster in practice. We also remark that the negative results (Σ_2^P -hard complexity) are for the easier version of the problem (the latter defined trivial *NASP*), and our positive algorithmic results extend, on the contrary, for the harder version of the problem with multiple followers. Besides, we also present several observations, for instance, Remarks 3 and 5, that shed light on equilibria for Nash games where players solve non-convex optimization problems. We believe that the above contributions, both from the complexity and algorithmic (computational) sides, establish a solid benchmark for future progress.

Literature Review

Nash [116, 117] introduced the concept of Nash Equilibrium in the context of simultaneous

n -person games, also known as Nash games in the optimization literature. A Nash equilibrium always exists in games with a finite number of players and a finite number of strategies. By definition, these equilibrium strategies ask that no player has an incentive to unilaterally deviate from the prescribed strategy. Generally, we distinguish between the pure strategy Nash equilibrium (*PNE*) and the mixed strategy one (*MNE*). The latter generalizes the pure one since each strategy in the support of the equilibrium has an associated probability of being played. The Nash equilibrium concept extends to games where players have an uncountable set of strategies. From an application perspective, interactions within economic markets extensively adopt Nash Games as a modeling paradigm. For instance, gas market bilevel formulations usually involve players solving convex optimization problems parametrized in other players variables [61, 62, 68, 69, 88, 134, 141]. On the other side, the cross-border kidney exchange model [28], competitive lot-sizing models [30, 104], and the fixed charge transportation model [131] feature players solving non-convex problems.

In contrast with Nash games, sequential ones partition the players into different groups, and each group decides in a round – or level. If the rounds are two, then the game is known as Stackelberg game [23, 139]. Here, the agents playing in the first round are the *leaders*, while the ones playing afterward are called *followers*. When each Stackelberg player solves an optimization problem, then we have a bilevel program. In general, bilevel formulations can model interactions where leaders have specific advantages over the followers, such as government taxing companies. Indeed, bilevel formulations allure a nourished community of researchers. Bard et al. [11, 12] model tax credits strategies in the context of biofuel production, and Brotcorne et al. [22], Labbé and Violin [99] create bilevel pricing problems. Feijoo and Das [66], Gabriel and Leuthold [74], Hobbs et al. [87] model pricing and environmental policies for energy markets, where power generators are leaders, and network operators are followers.

When multiple leaders – each with possibly multiple followers – seek an equilibrium between each other, we fall into the category of *EPECs*. Thereby leaders often have a common set of followers, and the equilibrium of interest is *PNE*. Sherali [136] introduced *EPECs* where both leaders and followers produce a homogeneous commodity, and followers adopt a *reaction curve*. Gabriel et al. [75] provides a Gauss-Seidel iteration technique to find *PNEs* for a restricted class of *EPECs*, where followers from distinct leaders can interact. Ralph and Smeers [125], and Hu and Ralph [89] extend the analysis on the existence of a *PNE* to specialized classes of *EPECs* arising in electricity markets. Leyffer and Munson [103] introduces a weaker solution concept based on a nonlinear programming reformulation. DeMiguel and Xu [52] craft the concept of stochastic multi-leader Stackelberg-Nash-Cournot equilibrium for a particular form of investment-production interaction between the players. More recently, Kulkarni and Shanbhag [97, 98] considered *EPECs* with shared constraints, presenting solution concepts

and algorithms starting from the potentiality of players' objectives.

Complexity of Equilibria. As previously mentioned, Nash [116, 117] proved that a Nash equilibrium for finite games always exists, and thus the associated decision problem is trivial. However, since the proof is non-constructive, it already unveils the difficulty of computing an equilibrium. Indeed, even for two-players finite games in strategic form, the problem of determining an equilibrium is PPAD-complete [36]. Furthermore, even for games where equilibria are guaranteed to exist, many variations of associated decision problems are known to be $\mathcal{NP}$ -complete [76]. A few illustrative examples are the existence of two equilibria or the existence of an equilibrium where a player's payoff exceeds a given threshold. Besides, Carvalho et al. [29, 32] proved the existence of *PNE* and *MNE* for games where players solve parametrized non-convex problems to be Σ_2^P -hard. Under this setting, if players' strategies are bounded, then an *MNE* always exists. For congestion games, another widely studied class of Nash games, *PNEs* always exist due to their potential nature [128]. [51] focus on congestions games where totally unimodular matrices describe the players' strategies. Within this context, the authors prove that if players have the same feasible set of strategies, a *PNE* can be computed in polynomial time. In any other case, the problem is PLS-complete. For what concerns Stackelberg games' complexity, the seminal result of Jeroslow [90] enlightens the matter. It proves that sequential games' computational complexity rises one layer up in the polynomial hierarchy for every additional round, even for linear problems. Thereupon, the classification of the computational complexity for *NASPs* becomes almost natural.

Paper Organization. We organize the manuscript as follows. Section 6.2 provides definitions and restates some known results. Section 6.3 provides the complexity results regarding *NASPs*. Section 6.4 presents an algorithm to find *MNE* for *NASP*, proving its finiteness and correctness. Section 6.5 builds on top of the developed algorithm by extending it with an inner approximation hierarchy and introduces a heuristic for computing *PNE*. Section 6.6 presents computational tests, and, finally, Section 6.7 draws conclusions.

6.2 Preliminaries

In this section, we provide definitions, notations and recall some known results in the context of polyhedral theory, Nash games, and Stackelberg games.

6.2.1 Definitions

Nash Games. When players decide simultaneously, and with complete information, we have a Nash Game. As a standard notation in game theory, let the operator $(\cdot)^{-i}$ denote $(\cdot)$ except i .

Definition 11 (Nash games). *A Nash game P among n players is a finite tuple of optimization problems $P = (P^1, \dots, P^n)$, where each P^i is the problem of the i^{th} player. Simultaneously, each player i solves an optimization problem of the form $\min_{x^i \in \mathcal{F}_i} \{f^i(x^i; x^{-i}) : x^i \in \mathcal{F}_i\}$, where f^i and $\mathcal{F}_i$ are their objective function and the feasible set, respectively.*

We can further characterize a Nash game as (i) *simple* if, for every player i and for some positive semi-definite matrix Q^i , a real vector c^i , and a real matrix C^i of appropriate dimensions, the objective function is in the form of $f^i(x^i; x^{-i}) = \frac{1}{2}x^{iT}Q^ix^i + (c^i + C^ix^{-i})x^i$, (ii) *linear*, if $Q^i = 0$ for all i , namely each leader has a linear objective function, (iii) *facile*, if the game is *simple*, and $\mathcal{F}_i$ is a polyhedron for all $i = 1, 2, \dots, n$.

Definition 12 (Simple parameterization). *An optimization problem in y has a simple parameterization with respect to $x \in \mathbb{R}^{n_\ell}$ if the problem is in the form of $\min_{y \in \mathbb{R}^{n_f}} \{f(y) + (Cx)^T y : y \in \mathcal{F}, Ax + By \leq b\}$, where $f : \mathbb{R}^{n_f} \rightarrow \mathbb{R}$, and C, A, B, b are matrices and vectors of appropriate dimensions, and $\mathcal{F} \subseteq \mathbb{R}^{n_f}$.*

A Nash game $P = (P^1, \dots, P^n)$ has a simple parameterization with respect to $x \in \mathbb{R}^{n_\ell}$ if each optimization problem $P^1(x), \dots, P^n(x)$ has a simple parameterization with respect to x .

Definition 13 (Mixed and Pure-strategy Nash equilibria). *Let $\nu = (\nu^1, \dots, \nu^n)$ where ν^i is a Borel probability distribution on $\mathcal{F}_i$ with finite support. Then, ν is a MNE if $\mathbb{E}(f(\nu^i, \nu^{-i})) \leq \mathbb{E}(f(\tilde{x}^i, \nu^{-i}))$ for any player i and $\tilde{x}^i \in \mathcal{F}_i$. If all the distributions have a singleton support, then the set of strategies is referred to as PNE.*

PNE is a strong notion of equilibrium, and even relatively trivial games – such as rock-paper-scissors – may not possess one. In contrast, an *MNE* always exists for finite games [116, 117].

Stackelberg Games. A Stackelberg game is a multi-level game with 2 rounds of decisions. First, the *leader* decides, optimizing their objective subject to some constraints. Then, the *followers* decide, with their objective and constraints now depending upon the leader's decision [24].

Definition 14 (Stackelberg game). *Let $P(x)$ be a Nash game with a simple parametrization with respect to x , let $\text{SOL}(P(x))$ denotes its solution set, and define $f : \mathbb{R}^{n_\ell + n_f} \rightarrow \mathbb{R}$. Then, a Stackelberg game is an optimization problem of the form $\min_{x \in \mathbb{R}^{n_\ell}; y \in \mathbb{R}^{n_f}} \{f(x, y) : (x, y) \in \mathcal{F}, y \in \text{SOL}(P(x))\}$.*

In a Stackelberg game, the set $\text{SOL}(P(x))$ is parametrized given the leader's strategy x . Namely, given an upper-level strategy x , the followers should play optimally. Each Stackelberg game's solution is then a *Subgame Perfect Nash-Equilibrium (SPNE)*. For the purposes of this work, we only consider *SPNEs*. The previous definition implies the Stackelberg game to be *optimistic*. Namely, if the game has multiple optimal solutions $\text{SOL}(P(x))$, then y takes the value among $\text{SOL}(P(x))$ benefitting the leader the most. Suppose $P(x)$ is an optimization problem (i.e., one follower). In that case, the optimistic assumption is natural: the leader – by incentivizing the follower with an arbitrarily small amount (e.g., a payment) – can persuade the follower to choose the most favorable solution (according to the leader). In general, when there are multiple followers, this optimistic assumption could be a strict restriction since the followers' equilibrium may not be unique. However, the optimistic assumption on the bilevel solution selection is not restrictive in our work, since the followers' equilibrium is unique. Since the followers of each *NASP*'s leader play a Nash-Cournot game with strictly convex objective functions, each lower-level equilibrium is unique.

Definition 15 (Simple Stackelberg game). *A Stackelberg game P is simple if $P(x)$ is a facile Nash game with a simple parameterization with respect to the upper-level variables x , $\mathcal{F}$ is a polyhedron, and $f(x, y)$ is a linear function.*

Definition 16 (NASP). *A NASP is a linear Nash game $N = (P^1, \dots, P^k)$ where for each i , $P^i(x^i)$ is a simple Stackelberg game.*

Combining Definition 11, 12 and 14 to 16, a *NASP* refers to the following game. There is a set L of players called the *leaders*, each of which has a set F of second-level players called the *followers*. We will use the term *players* to point to the Stackelberg games associated to each leader. Each follower $f \in F$ has a unique leader $\ell(f) \in L$, such that the objective function and the feasible set of the follower f depends only upon the decision variables of $\ell(f)$ and other followers $f' \in F$ such that $\ell(f') = \ell(f)$. In other words, each follower interacts with the followers having the same leader, and not with followers from other leaders. We assume both leaders' and followers' parameters are common information. First, with complete anticipation of their followers' behavior and simultaneously, every leader $\ell \in L$ chooses their decision to maximize their utility. Then, every follower f observes their respective leader's (i.e., $\ell(f)$'s) decision, and every follower simultaneously chooses their decision by maximizing their utility.

We assume an *optimistic* behavior from the followers in the sense that – if there are multiple optimal strategies for the followers over which they are indifferent – they will choose the strategy which benefits their leader the most.

Definition 17 (Trivial NASP). *A trivial NASP is a NASP where $k = 2$, and P^1 and P^2 are simple bilevel games whose lower levels are linear programs with a simple parameterization with respect to the upper-level variables.*

The additional assumptions holding on a trivial *NASP* (as of Definition 17) compared to a general *NASP* (as of Definition 16) are seemingly strong. We require that each leader has precisely one follower – as opposed to finitely many followers – and that each follower solves a linear program – as opposed to a quadratic program – with a simple parameterization with respect to the upper-level variables. For instance, the game between the *Latin* and *Greek* leaders presented in (6.1) is an example of trivial *NASP*. Also, in *NASPs* with lower-level facile Nash Games, the feasible region for the followers are convex, and the leaders' objective functions are convex in x^i . As a consequence, the existence of a *PNE* is guaranteed whenever the feasible regions are compact [50, 65, 78]. Therefore, one can solely search for *PNEs* among the followers despite considering both *MNEs* and *PNEs* among the leaders.

In the optimization literature, Nash games often reformulate as *Linear Complementarity Problems* (*LCPs*). This reformulation leverages the complementarity conditions induced by the optimality conditions (i.e., the *KKT* conditions) of the players' optimization problems. *LCPs* have a rich theoretical basis [46, 63], and can be formulated as mixed-integer programs (*MIPs*). Following the usual notation, let operator $x \perp y$ be equivalent to $x^T y = 0$.

Definition 18 (Linear complementarity problem). *Given $M \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$, the linear complementarity problem (LCP) asks to find a $x \in \mathbb{R}^n$ so that $0 \leq x \perp Mx + q \geq 0$, or to show that no such x exists. We denote as the feasible set induced by the LCP the set of all x satisfying the condition of the LCP.*

Simplifying assumptions and limitations. We summarize the assumptions made on *NASPs*. First, a *NASP* is a game among leaders of Stackelberg games, each with a specific set of followers. The actions of the followers of a given leader do not directly affect another leader, or followers of a different leader. Second, we assume optimistic behavior by the leader. Given a leader's decision, should there be multiple equilibria for the followers, the leader will choose the most favorable equilibrium from its standpoint. This also has consequences for the upcoming results about the absence of equilibria. When we state that a *NASP* does not have an equilibrium (either a *PNE* or an *MNE*), our claim is as follows: should the leaders

always select the most favorable lower-level equilibrium, no equilibrium exists among the leaders. Thus, if the leader does not select the most favorable equilibrium among the followers, an equilibrium among the leaders *might* exist. In this sense, we consider non-optimistic equilibrium selection by the leaders to be beyond the scope of this work.

6.2.2 Existing Results

We introduce a series of known results we will later use to provide theoretical and algorithmic contributions. Cottle et al. [46] proves that a facile Nash game can be restated as an *LCP*.

Theorem 6 (Cottle et al. [46]). *Let P be a facile Nash game. Then, there exists a matrix M and a vector q such that every solution to the LCP defined by M, q is a PNE for P and every PNE of P solves the LCP.*

Basu et al. [13], with Theorem 7, provide an extended formulation for the feasible region of a simple Stackelberg game. This result is a critical ingredient of our contribution, since it will enable us to provide a polyhedral characterization *NASPs*.

Theorem 7 (Basu et al. [13]). *Let S be the feasible set of a simple Stackelberg game. Then, S is a finite union of polyhedra. Conversely, let S be a finite union of polyhedra. Then, there exists a simple Stackelberg game with $P(x)$ containing exactly 1 player such that the feasible region of the simple Stackelberg game provides an extended formulation of S .*

Finally, the celebrated Theorem 8 from Balas [8] provides an extended formulation for the closure of the convex hull for the union of a finite set of polyhedra.

Theorem 8 (Balas [8]). *Given k polyhedra $S_i = \{x \in \mathbb{R}^n : A^i x \leq b^i\}$ for $i = 1, \dots, k$, then $\text{cl conv}(\bigcup_{i=1}^k S_i)$ is given by the set $\{x \in \mathbb{R}^n : \exists(x^1, \dots, x^k, \delta) \in (\mathbb{R}^n)^k \times \mathbb{R}^k : x \in \{A^i x^i \leq \delta_i b^i, \sum_{w=1}^k x^w = x, \sum_{w=1}^k \delta_w = 1, \delta_i \geq 0, \forall i \in [k]\}\}$.*

6.3 Hardness of Finding a Nash equilibrium

In what follows, we characterize the computational complexity of *NASPs*. We formalize the intuition stemming from Jeroslow [90] with a reduction from the **SUBSET SUM INTERVAL** problem. The main results are summarized below.

Theorem 9. *It is Σ_2^P -hard to decide if a trivial NASP has a PNE.*

Corollary 1. *If each player's feasible set in a trivial NASP is a bounded set, an MNE exists.*

Theorem 10. *It is Σ_2^P -hard to decide if a trivial NASP has an MNE.*

In what follows, we will provide the proof of Theorems 9 and 10. First, we formally introduce the SUBSET SUM INTERVAL.

Definition 19 (SUBSET SUM INTERVAL). *Given $q_1, \dots, q_k, p, t, k \in \mathbb{Z}_+$, with none of them equal to zero, and $\log_2(t - p) \leq k$, does there exist a $s \in \mathbb{Z} : p \leq s < t$, so that for all $I \subseteq \{1, 2, \dots, k\}$ then $\sum_{i \in I} q_i \neq s$.*

In other words, we seek – within an interval of integers – for a number s that cannot be expressed as a sum of a subset of $\{q_1, \dots, q_k\}$ or alternatively show that no such s exists. Here, $t - p$ can be chosen as a power of 2. For instance, we may ask if there exist an r in $\mathbb{Z}_+$ such that $2^r = t - p$. Eggermont and Woeginger [60] proven that, given r in $\mathbb{Z}_+$ such that $t - p = 2^r$, the problem is Σ_2^P -hard.

Theorem 11 (Eggermont and Woeginger [60]). *Given that there exists $r \in \mathbb{Z}_+$ such that $t - p = 2^r$, SUBSET SUM INTERVAL is Σ_2^P hard.*

Proof of Theorem 9. To show the hardness of NASP, we will rewrite SUBSET SUM INTERVAL as a trivial NASP of comparable size. Then, appealing to Theorem 11, we establish the hardness of a trivial NASP. Finally, we claim that NASP is only a generalization of trivial NASP, which could not be any easier.

Consider a trivial NASP as of in Definition 17. For the sake of clarity, we call the two Stackelberg games associated with the trivial NASP the *Latin*, and *Greek* game, respectively. The decision variables of the *Latin* game's leader are x , and their follower controls y variables. Similarly, the decision variables of the *Greek* game are ξ , and χ for their follower. As for the SUBSET SUM INTERVAL, we stick to the notation introduced in Definition 19.

Let $b_1, \dots, b_r \in \{0, 1\}$ as the unique r -bit binary representation of $s - p$: for instance, $\{b_i\}_{i=1}^r$ satisfies $s - p = \sum_{i=1}^r b_i 2^{i-1}$. Then, let $P = k + 2r$, $Q = \sum_{i=1}^k q_i$, and $T = t - 1 + rQ$, where both can be computed in polynomial time with respect to the data in SUBSET SUM INTERVAL.

The Latin Game.

$$\max_{\substack{x_0, x_1, \dots, x_{2P} \\ \in \mathbb{R} \\ y_0, y_1, \dots, y_{2P} \\ \in \mathbb{R}}} : (T-1)\xi_0 x_0 + \sum_{i=1}^k q_i \xi_i x_{P+i} + Q \sum_{i=k+1}^P \xi_i x_{P+i} \quad (6.2a)$$

$$\text{subject to} \quad x_i = 0 \quad i = 1, \dots, k \quad (6.2b)$$

$$y_i \geq 0 \quad i = 1, \dots, 2P \quad (6.2c)$$

$$x_i \geq 0 \quad i = 1, \dots, 2P \quad (6.2d)$$

$$\sum_{i=k+1}^P x_i \leq r \quad (6.2e)$$

$$x_i + x_{P+i} \leq 1 \quad i = 1, \dots, P \quad (6.2f)$$

$$x_0 + x_{P+i} \leq 1 \quad i = 1, \dots, P \quad (6.2g)$$

$$(y_0, \dots, y_{2P}) \in \arg \min_y \left\{ \sum_{i=0}^{2P} y_i : \begin{array}{l} y_i \geq -x_i \\ y_i \geq x_i - 1 \end{array} \forall i = 0, \dots, 2P \right\} \quad (6.2h)$$

The Greek Game.

$$\max_{\substack{\xi_0, \xi_1, \dots, \xi_P \\ \in \mathbb{R} \\ \chi_0, \dots, \chi_P \\ \in \mathbb{R}}} : (T-1)\xi_0 + \sum_{i=1}^k q_i \xi_i (1 - x_{P+i}) + Q \sum_{i=k+1}^P \xi_i (1 - x_i - x_{P+i}) \\ + \sum_{i=k+1}^{k+r} 2^{i-k-1} \xi_i (1 - x_i - x_{P+i}) - \sum_{i=k+1}^P T (x_i \xi_i + (1 - x_i)(1 - \xi_i - \xi_0)) \quad (6.2i)$$

$$\text{subject to} \quad \xi_i \geq 0 \quad \forall i = 0, \dots, P \quad (6.2j)$$

$$\xi_i \leq 1 \quad \forall i = 0, \dots, P \quad (6.2k)$$

$$\chi_i \geq 0 \quad \forall i = 0, \dots, P \quad (6.2l)$$

$$\sum_{i=k+1}^P \xi_i + r \xi_0 \geq r \quad (6.2m)$$

$$T \geq T \xi_0 + \sum_{i=1}^k q_i \xi_i + Q \sum_{i=k+1}^P \xi_i + \sum_{i=k+1}^{k+r} 2^{i-k-1} \xi_i \quad (6.2n)$$

$$(\chi_0, \dots, \chi_P) \in \arg \min_{\chi} \left\{ \sum_{i=0}^P \chi_i : \begin{array}{l} \chi_i \geq -\xi_i \\ \chi_i \geq \xi_i - 1 \end{array} \forall i = 0, \dots, 2P \right\} \quad (6.2o)$$

We claim the game in (6.2) has a *PNE*, if and only if the **SUBSET SUM INTERVAL** instance has a decision YES.

Claim 1. *The game defined in (6.2) is a trivial NASP.*

Claim 2. *The region in the space of x defined by (6.2c) and (6.2h) is the Cartesian product of $(\{x_i : x_i \leq 0\} \cup \{x_i : x_i \geq 1\})$, for $i = 0, \dots, 2P$. Similarly, the region in the space of ξ defined by (6.2l) and (6.2o) is the Cartesian product of $(\{\xi_i : \xi_i \leq 0\} \cup \{\xi_i : \xi_i \geq 0\})$, for $i = 0, \dots, P$.*

We refer the reader to appendix B.1 for the proofs of Claims 1 and 2.

Claim 3. *If $((\bar{x}, \bar{y}), (\bar{\xi}, \bar{\chi}))$ is a PNE for (6.2), then $\bar{\xi}_0 \neq 0$.*

Proof of Claim. First, observe that $\xi \neq 0$, since setting $\xi_0 = 1$ is a feasible profitable deviation for the *Greek* leader, regardless of the *Latin* leader's decision. Suppose $\bar{\xi}_0 = 0$ and for some $\emptyset \neq L \subseteq \{1, \dots, P\}$, $\xi_\ell \neq 0$. Note that the *Latin* leader has no incentive to keep $\bar{x}_0 = 1$, which forces an objective value of 0. Instead, it can choose $\bar{x}_0 = 0$, and $\bar{x}_{P+\ell} = 1$ for all $\ell \in L$ and any feasible value for $\bar{x}_{P+\ell}$ for $\ell \in \{1, \dots, P\} \setminus L$. One can check that this is feasible and optimal for the the *Latin* leader, given $\bar{\xi}_0 = 0$. This also means that the *Greek* leader's objective is 0, as each of the summands in their objective vanishes, and $\bar{\xi}_0 = 0$ makes the first term vanish. Hence, this cannot be a Nash equilibrium since the Greek leader has a profitable deviation by setting $\xi_0 = 1$ and $\xi_i = 0$ for all $i \neq 0$, which is feasible and yields an objective value of $T - 1 > 0$.

Claim 4. *If SUBSET SUM INTERVAL has decision YES, then (6.2) has a PNE.*

Proof of Claim. Suppose there exists $s \in \mathbb{Z}_+$ such that $p \leq s \leq t - 1$, and for all $I \subseteq \{1, \dots, k\}$, $\sum_{i \in I} q_i \neq s$. Also, recall the unique r -bit binary representation of $s - p$, namely $b_1, \dots, b_r \in \{0, 1\}$. Consider the following strategy:

$$x_0 = 1 \tag{6.3a}$$

$$x_i = 0 \quad \forall i = 1, \dots, k \tag{6.3b}$$

$$x_i = b_{i-k} \quad \forall i = k + 1, \dots, k + r \tag{6.3c}$$

$$x_i = 1 - b_{i-k-r} \quad \forall i = k + r + 1, \dots, P = k + 2r \tag{6.3d}$$

$$x_i = 0 \quad \forall i = P + 1, \dots, 2P \tag{6.3e}$$

$$y_i = 0 \quad \forall i = 0, \dots, 2P \tag{6.3f}$$

$$\xi_0 = 1 \tag{6.3g}$$

$$\xi_i = 0 \quad \forall i = 1, \dots, P \tag{6.3h}$$

$$\chi_i = 0 \quad \forall i = 1, \dots, P \tag{6.3i}$$

It is easy to check that the strategy in (6.3) is feasible. Given ξ , observe that the strategy is optimal for the *Latin* leader as follows. Due to the choice $\xi_i = 0$ for $i \neq 0$, all but the first term of the *Latin* leader vanish. The largest value the first term can take corresponds to $x_0 = 1$. The remaining terms do not affect the *Latin* leader's objective, as long as they are feasible.

For what concerns the *Greek* leader, the current objective is $T - 1$. We show there exist no deviation which can improve their objective. With $\xi_0 = 1$, clearly no other deviation is feasible. Consider the deviation $\xi_0 = 0$: with such strategy the first term in the objective vanishes. Let $M = \{i \in \{k + 1, \dots, k + 2r\} : \bar{x}_i = 1\}$. Observe that $|M| = r$, and let $L = \{k + 1, \dots, k + 2r\} \setminus M$. Notice that we require $\xi_\ell = 1$ for $\ell \in L$, otherwise the fifth term in the objective would be a large negative quantity. Hence, the objective would not exceed the value of $T - 1$. With such a choice of ξ_ℓ for $\ell \in L$, the fifth term in the objective evaluates to 0, and the fourth term evaluates to $\sum_{\ell \in L} 2^{\ell-k-i} = \sum_{i=k+1}^{k+r} (1-b_{i-k})2^{i-k-1} = 2^r - 1 + p - s = t - 1 - s$. Therefore, the objective value is $t - 1 + rQ - s$. However, since it is a YES instance of **SUBSET SUM INTERVAL**, the deficit s in the objective value can never be made up by any choice of ξ_i for $i = 1, \dots, k$ and by making the second term equal to s . If such ξ_i are chosen to exceed s , then (6.2n) is violated if it is strictly less than s , and the objective cannot exceed $T - 1$. Hence, this is no longer a valid deviation. Thus (6.3) is indeed a Nash equilibrium. $\square$

Claim 5. *If **SUBSET SUM INTERVAL** has decision NO, then (6.2) has no PNE.*

Proof of Claim. We prove the result by contradiction. In order to establish the latter, assume that the **SUBSET SUM INTERVAL** instance has an answer NO, and there exists a *PNE* $((\bar{x}, \bar{y}), (\bar{\xi}, \bar{\chi}))$ for (6.2), with $\bar{\xi}_0 = 1$. From Claims 3-2, any *PNE* necessarily has $\xi_0 = 1$. From (6.2n), $\bar{\xi}_0 = 1$ enforces that $\bar{\xi}_i = 0$ for $i = 1, \dots, T$, and hence the *Greek* leader has an objective value of $T - 1$. Therefore, with $\bar{\xi} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}$, observe that the *Latin* leader's objective is $(T - 1)x_0$. Thus, we necessarily have $\bar{x}_0 = 1$. From (6.2g), we deduce $\bar{x}_{P+i} = 0$ for $i = 1, \dots, P$, while from (6.2e) we obtain $\bar{x}_i \leq \frac{r}{r+1}$ for $i = 1, \dots, k$. The only value of $\bar{x}_i$ that satisfies this condition along with (6.2h) is $\bar{x}_i = 0$ for $i = 1, \dots, k$. That only leaves $\bar{x}_i$ for $i = k + 1, \dots, k + 2r = P$. We can now show that – for any value of $\bar{x}_i$ – the *Greek* leader has a profitable deviation. Namely, it can get an objective strictly greater than $T - 1$. Let $M = \{i \in \{k + 1, \dots, k + 2r\} : \bar{x}_i = 0\}$. From (6.2e), we have $|M| \geq r$. We choose some $L \subseteq M$ such that $|L| = r$, and for $i \in L$, we set $\bar{\xi}_i = 1$. Since $|L| = r$, and $L \subseteq M$, the third term in the *Greek* leader's objective evaluates to rQ . The fourth term is in between 0 and $2^r - 1$, and the fifth term vanishes. Keeping in mind that $\bar{\xi}_0 = 0$, the objective now evaluates to a number between $\sum_{i=1}^k q_i \xi_i + rQ$ and $\sum_{i=1}^k q_i \xi_i + rQ + 2^r - 1$. In other words, the objective is $T - s + \sum_{i=1}^k q_i \xi_i$ and $p \leq s \leq t - 1$. Since this is a NO instance of **SUBSET SUM INTERVAL**,

$\exists I \subseteq \{1, \dots, k\}$ such that $\sum_{i \in I} q_i \xi_i = s$. Set $\bar{\xi}_i = 1$ if $i \in I$, and $\bar{\xi}_i = 0$ if $i \in \{1, \dots, k\} \setminus I$. This is feasible, and makes the objective value equal to T , which is a profitable deviation from $T - 1$. Therefore $((\bar{x}, \bar{y}), (\bar{\xi}, \bar{\chi}))$ is not a Nash equilibrium. $\square$

From Theorem 9, we have a direct implication of Corollary 2.

Corollary 2. *Consider a linear Nash Game $N = (P^1, \dots, P^n)$ where each P^i is an MIP. It is Σ_2^P -hard to decide if N has a PNE.*

Proof of Corollary 2. Bounded and continuous bilevel programs can be reformulated as bounded integer programs of polynomial-size [13]. The *Greek* and the *Latin* leaders' problems defined in (6.2) are bounded bilevel programs, where each variable necessarily takes value in $[0, 1]$. $\square$

Furthermore – under an assumption of boundedness – we prove Corollary 1, showing that an *MNE* always exists.

Proof of Corollary 1. Let $\mathcal{F}_i$ be the feasible region of the i -th player (leader), namely a bounded set. Given x^{-i} , the objective of its optimization problem is linear. Hence, an optimal solution always exists, which is an extreme point of $\text{conv}(\mathcal{F}_i)$. However, given that $\mathcal{F}_i$ are feasible sets of bilevel linear programs, we know that the feasible region of the leaders is a finite union of polyhedra from Theorem 7. It follows that $\text{conv}(\mathcal{F}_i)$ is a polyhedron. Since we also assume boundedness, $\text{conv}(\mathcal{F}_i)$ is indeed a polytope. Thus, the i -th player's strategy is the set of extreme points of this polytope, finite in number. Since the same reasoning holds for each player, this is a Nash game with finitely many strategies. From Nash [116, 117], such a game has an *MNE*. $\square$

From Corollary 1, deciding on the existence of an *MNE* is trivial if each player has a bounded feasible set. We extend this result with Theorem 10, showing that even if one player's feasible region is unbounded, then deciding on the existence of an *MNE* is Σ_2^P -hard.

Before proving Theorem 10, we introduce the technical Theorem 12. While Theorem 7 shows that *any* finite union of polyhedra can be written as a feasible region of a bilevel problem in a lifted space, Theorem 12 explicitly describes this set for a given union of two polyhedra.

Lemma 12. *Consider the set $\mathcal{S}$ defined as the union of two polyhedra, namely*

$$\mathcal{S} = \left\{ (h, y, x) \in \mathbb{R}_+^3 : h = x; y = 1 \right\} \cup \left\{ (h, y, x) \in \mathbb{R}_+^3 : h = 0; y = 0 \right\} \quad (6.4)$$

$\mathcal{S}$ has an extended formulation as a feasible set of a simple bilevel program.

From Theorem 12 we can further derive Theorem 13.

Lemma 13. *Suppose $S \subseteq \mathbb{R}^{n_1}$ and $T \subseteq \mathbb{R}^{n_2}$ have an extended formulation as bilevel programs. So does $S \times T$.*

Therefore, with Theorems 12 and 13, we can then prove Theorem 10. Both the proofs for these two lemmas can be found in appendix B.1.

Proof of Theorem 10. We reduce **SUBSET SUM INTERVAL** into a problem of deciding the existence of an *MNE* for a trivial *NASP*. Let $Q = \sum_{i=1}^k q_i$. Also, as of Theorem 9, let the *Latin* game and the *Greek* game have Latin and Greek terms, respectively.

Latin Game. The *Latin* game is a Stackelberg game. The variables of the leader and the follower are denoted by Latin alphabets x and y , respectively.

$$\begin{aligned} \max_{\substack{x_0, \dots, x_{k+3r+1} \\ \in \mathbb{R} \\ y_0, \dots, y_k \\ \in \mathbb{R}}} : & \quad \frac{x_0}{2} + \sum_{i=1}^k q_i x_i + 2(Q+1)\xi_{r+1}x_{k+3r+1} \\ & \quad - (Q+1) \left(\sum_{i=1}^r 2^{i-1} x_{k+i} + p x_{k+3r+1} \right) \end{aligned} \quad (6.5a)$$

$$\text{subject to} \quad x_i \geq 0 \quad \forall i = 0, \dots, k \quad (6.5b)$$

$$y_i \geq 0 \quad \forall i = 0, \dots, k \quad (6.5c)$$

$$x_i \geq 1 \quad \forall i = 0, \dots, k \quad (6.5d)$$

$$x_{k+3r+1} = x_{k+2r+i} \quad \forall i = 1, \dots, r \quad (6.5e)$$

$$x_{k+3r+1} = p + \sum_{i=1}^r 2^{i-1} x_{k+r+i} \quad (6.5f)$$

$$\frac{x_0}{2} + \sum_{i=1}^k q_i x_i \leq x_{k+3r+1} \quad (6.5g)$$

$$(x_{k+i}, x_{k+r+i}, x_{k+2r+i}) \in \mathcal{S} \quad (\text{as in (6.4)}) \quad \forall i = 1, \dots, r \quad (6.5h)$$

$$(y_0, \dots, y_k) \in \arg \min_y \left\{ \sum_{i=0}^k y_i : \begin{array}{l} y_i \geq -x_i \\ y_i \geq x_i - 1 \end{array} \quad \forall i = 0, \dots, k \right\} \quad (6.5i)$$

Greek Game. Similarly, the *Greek* game is a Stackelberg game, where the leader and the follower variables are denoted by Greek alphabets ξ and χ , respectively.

$$\max_{\substack{\xi_0, \dots, \xi_{r+1} \\ \in \mathbb{R} \\ \chi_1, \dots, \chi_r \\ \in \mathbb{R}}} : (1 - x_0)\xi_0 \quad (6.5j)$$

$$\text{subject to } \xi_i \geq 0 \quad \forall i = 1, \dots, r \quad (6.5k)$$

$$\chi_i \geq 0 \quad \forall i = 1, \dots, r \quad (6.5l)$$

$$\xi_i \leq 1 \quad \forall i = 1, \dots, r \quad (6.5m)$$

$$p + \sum_{i=1}^r 2^{i-1} \xi_i = \xi_{r+1} \quad (6.5n)$$

$$(\chi_1, \dots, \chi_r) \in \arg \min_{\chi} \left\{ \sum_{i=1}^r \chi_i : \begin{array}{l} \chi_i \geq -\xi_i \\ \chi_i \geq \xi_i - 1 \end{array} \forall i = 0, \dots, r \right\} \quad (6.5o)$$

Claim 6. *The game defined in (6.5) is a trivial NASP.*

Claim 7. *The region of space for x – defined by (6.5c) and (6.5i) – is the Cartesian product of $(\{x_i : x_i \leq 0\} \cup \{x_i : x_i \geq 1\})$ for $i = 0, \dots, k$. Similarly the region of the space for ξ – defined by (6.5l) and (6.5o) – is the Cartesian product of $(\{\xi_i : \xi_i \leq 0\} \cup \{\xi_i : \xi_i \geq 0\})$ for $i = 1, \dots, k$.*

The proof of this claim is analogous to the ones of Claims 2-1.

Claim 8. *x_{k+3r+1} takes integer values only.*

Proof of Claim. From (6.5h), each x_{k+r+i} for $i = 1, \dots, r$ can take a value of either 0 or 1, depending upon which of the two polyhedra (in the definition of S) the variable falls in. Moreover, since in (6.5f) the RHS is a sum of integers, the LHS x_{k+3r+1} is also an integer.

Claim 9. *$(x_{k+3r+1})^2 = \sum_{i=1}^r 2^{i-1} x_{k+i} + p x_{k+3r+1}$ holds for the Latin game's feasible set.*

Proof of Claim. Consider the set S defined in (6.4). For a point $h = x$ and $y = 1$ in the first polyhedra, one can write $h = xy$. Similarly, for a point $h = 0$ and $y = 0$ in the second polyhedron, then $h = xy$. Thus, the nonlinear equation $h = xy$ is valid for the set S . By

multiplying both sides of (6.5f) with x_{k+3r+1} , one gets

$$\begin{aligned}
(x_{k+3r+1})^2 &= px_{k+2r+1} + \sum_{i=1}^r 2^{i-1} x_{k+r+i} x_{k+3r+1} \\
&= px_{k+3r+1} + \sum_{i=1}^r 2^{i-1} x_{k+r+i} x_{k+2r+i} \\
&= px_{k+3r+1} + \sum_{i=1}^r 2^{i-1} x_{k+i}
\end{aligned}$$

The second equality follows from (6.5e), and the third equality from the fact that $h = xy$ is valid for S and (6.5h).

Claim 10. *Given some $\xi_{r+1} \in \mathbb{Z}$ between p and $t - 1$, the Latin player has a profitable unilateral deviation for any feasible strategy with $x_{k+3r+1} \neq \xi_{r+1}$.*

Proof of Claim. Note that if ξ_{r+1} is between p and $t - 1$, then $x_{k+3r+1} = \xi_{r+1}$ is feasible for the *Latin* game. Observe the last two terms of the objective function. From Claim 9, we can rewrite them as $(Q + 1)(2\xi_{r+1}x_{k+3r+1} - x_{k+3r+1}^2)$. By focusing just on the last two terms, these reach a maximum value for the *feasible* choice of $x_{k+3r+1} = \xi_{r+1}$. We can now argue that the player can never be optimal by choosing $x_{k+3r+1} \neq \xi_{r+1}$. As established in Claim 8, x_{k+3r+1} is restricted to take integer values, and for any other choice x_{k+3r+1} , the deficit in objective value is at least $Q + 1$. However, even if each of the other terms take their maximum possible value, the largest value they can add to is $0.5 + Q < Q + 1$. the claim follows.

Claim 11. *If SUBSET SUM INTERVAL has decision YES, then (6.2) has a PNE (and hence an MNE).*

Proof of Claim. Let s be an integer such that $p \leq s < t$ and $\forall I \subseteq \{1, \dots, k\}$, $\sum_{i \in I} q_i \neq s$, and let $b_1, \dots, b_r \in \{0, 1\}$ be the unique r -bit binary representation of $s - p$. Consider the

following pure strategies for the players:

$$x_{k+3r+1} = s \tag{6.6a}$$

$$x_{k+2r+i} = s \quad i = 1, \dots, r \tag{6.6b}$$

$$x_{k+r+i} = b_i \quad i = 1, \dots, r \tag{6.6c}$$

$$x_{k+i} = b_i s \quad i = 1, \dots, r \tag{6.6d}$$

$$x_0 = 1 \tag{6.6e}$$

$$\xi_0 = 0 \tag{6.6f}$$

$$\xi_i = b_i \quad i = 1, \dots, r \tag{6.6g}$$

$$\xi_{r+1} = s \tag{6.6h}$$

Finally, choose $x_i \in \{0, 1\}$ for $i = 1, \dots, k$ such that $\sum_{i=1}^k q_i x_i$ is the largest value not exceeding s . Since it is a YES instance of **SUBSET SUM INTERVAL**, $\sum_{i=1}^k q_i x_i \leq s - 1$, and thus the strategy is indeed feasible for both the players. The *Latin* player has no feasible profitable deviation. This follows from the fact that x_{k+3r+1} cannot be chosen differently due to Claim 10. Moreover, the first two terms in the above strategy already take the largest possible value not violating (6.5g). Thus the *Latin* player has no profitable deviation. Now for the *Greek* player, since $x_0 = 1$, the objective value is always zero, and cannot be improved. Thus, the strategy in (6.6) is indeed a *PNE*.

Claim 12. *If **SUBSET SUM INTERVAL** has decision NO, then (6.2) has no MNE.*

Proof of Claim. Recall x_{k+3r+1} is forced to be an integer between p and $t - 1$. For any choice of x_{k+3r+1} , $x_0 = 0$ is selected and $x_1, \dots, x_k$ are so that (6.5g) holds with equality. There is no incentive to choose $x_0 = 1$, which will contribute to only 0.5 in the objective. However, with $x_0 = 0$, the *Greek* player can choose arbitrarily large values of ξ_0 . Hence, there is always a larger choice of ξ_0 which constitute a profitable deviation. Thus, no equilibrium exists for the game. $\square$

6.4 An Enumeration Algorithm to find *MNEs* for *NASPs*

First, we introduce Algorithm 5, which enumerates the polyhedra whose union corresponds to each player's feasible region. Then, it finds a pure-strategy Nash equilibrium in the convex hull of each player's feasible regions. We will prove the equivalence between finding a *PNE* over the convex hull and the original problem. Also, we remark that we call player the Stackelberg game associated with a given leader and their followers.

Algorithm 5: Enumeration algorithm to obtain an *MNE* for a *NASP*

Data: A description of *NASP* $N = (P^1, \dots, P^n)$

Result: For each $i = 1, \dots, n$, $\hat{x}_j^i$ is a pure-strategy played with probability p_j^i ,
presenting a mixed-strategy with support size k^i

```

1 for  $i = 1, \dots, n$  do
2   Enumerate the polyhedra whose union defines the feasible set  $\mathcal{F}_i$  of  $P^i$ ;
3    $\tilde{\mathcal{F}}_i \leftarrow \text{cl conv } \mathcal{F}_i$  by applying Theorem 8;
4    $\tilde{P}^i \leftarrow$  objective function of  $P^i$  and a feasible set of  $\tilde{\mathcal{F}}_i$ ;
5 Solve the facile Nash game  $\tilde{N} = (\tilde{P}^1, \dots, \tilde{P}^n)$  to obtain either a PNE,  $(\tilde{x}^1, \dots, \tilde{x}^n)$  or
   show that no PNE exists;
6 if no PNE exists for  $\tilde{N}$  then
7   return no MNE
8 for  $i = 1, \dots, n$  do
9   if  $\tilde{x}^i \in \mathcal{F}_i$  then
10     $\hat{x}_1^i \leftarrow \tilde{x}^i$ ;  $p_1^i \leftarrow 1$ ;  $k^i \leftarrow 1$ ;
11  else
12     $\tilde{x}^i = \sum_{j=1}^{k^i} \eta_j \hat{x}_j^i$  for  $\hat{x}_1^i, \dots, \hat{x}_{k^i}^i \in \mathcal{F}_i$  with  $\eta_j \geq 0$  and  $\sum_{j=1}^{k^i} \eta_j = 1$ ;
13     $p_j^i \leftarrow \eta_j$  for  $j = 1, \dots, k^i$ ;
14 return  $(\hat{x}_j^i, p_j^i)$  for each  $i = 1, \dots, n$  and  $j = 1, \dots, k^i$ 

```

The feasible region. Consider the feasible region of a simple Stackelberg game, given by $\{A'u + B'v \leq b, v \in \text{SOL}(P(u))\}$. Using the *KKT* conditions of the lower-level players in $P(u)$, we can rewrite the Stackelberg game feasible region as in (6.7), which is a union of polyhedra:

$$S = \left\{ x : \begin{array}{l} Ax \leq b \\ z = Mx + q \\ 0 \leq x_i \perp z_i \leq 0, \quad \forall i \in \mathcal{C} \end{array} \right\}. \quad (6.7)$$

Preliminary Enumeration Algorithm. Algorithm 5 exploits the polyhedral structure of each player's feasible region. Step 2 explicitly enumerates all such polyhedra, while Step 3 computes the closure of their convex hull using Theorem 8. Since this convex hull is also a polyhedron, the game $\tilde{N}$ (defined in Step 5) is a facile Nash game, and we can get a *PNE* for the game using Theorem 6.

Let $\tilde{x}$ be a *PNE* of $\tilde{N}$ and $\tilde{x}^i$ be the strategy of the i -th player. If $\tilde{x}^i$ belongs to $\mathcal{F}_i$, then at equilibrium i plays $\tilde{x}^i$ in N . If $\tilde{x}^i$ does not belong to $\mathcal{F}_i$, it is still contained in $\text{cl conv } \mathcal{F}_i$.

Thereby, $\tilde{x}^i$ can be expressed as a convex combination of points – or strategies – in $\mathcal{F}_i$ or a limit of such points. Player i would then play a mixed-strategy where each weight in the convex combination – or δ of Theorem 8 – is the probability of playing the corresponding pure-strategy, as in Step 12 of Algorithm 5. We remark that the *LCP* solved in Step 5 is implemented as a feasibility problem and solved as a *MIP*. Being interested in a specific equilibrium, one can add an objective function to this *LCP* problem, thus allowing the user to perform equilibria selection (if more than one exists). A visualization of the rationale behind the algorithm is in Figure 6.2. We formalize the correctness and finite termination of the above procedure in Theorem 14.

Theorem 14. *Algorithm 5 terminates finitely and (i) if it returns $\hat{x}_j^i, p_j^i$ for each $i = 1, \dots, n$, and $j = 1, \dots, k^i$, then the strategy profile is indeed an MNE for the NASP, (ii) if it returns failure, then N has no MNE.*

Proof of Theorem 14. For the purpose of this proof, we adopt the same notation introduced in Algorithm 5. First, the algorithm terminates in a finite number of steps: all loops in Algorithm 5 are finite loops, Step 2 ends finitely since there are only finitely many polyhedra (see Theorem 7), and Step 3 is also a finite procedure.

Proof of Statement (i) . Observe that if Algorithm 5 does not return failure, then Step 5 finds *PNE* $\tilde{x}$ for $\tilde{N}$. Each player's objective function is linear, and the distribution for the *MNE* has finite support. Therefore, one can observe that - for each player i - the following holds:

$$\mathbb{E} \left((c^i + C^i \hat{x}^{-i})^T \hat{x}^i \right) = \sum_{j'} \sum_{j=1}^{k_i} p_{j'}^{-i} p_j^i (c^i + C^i \hat{x}_{j'}^{-i})^T \hat{x}_j^i = (c^i + C^i \tilde{x}^{-i})^T \tilde{x}^i. \quad (6.8)$$

Assume a generic player i has an unilateral profitable deviation $\dagger \hat{x}_j^i$, and $\dagger p_j^i$ for $i = 1, \dots, \ell^i$ from $\hat{x}^i$ in their $\tilde{P}^i$ problem. Such a deviation is also a mixed-strategy profile. Consider now the pure-strategy for $\tilde{N}$ given by $\sum_{j=1}^{\ell^i} (\dagger p_j^i \dagger \hat{x}_j^i)$. It is feasible for the facile game $\tilde{P}^i$. Therefore, leveraging on the linearity of each player's objective function, we can show that this is also a profitable deviation for $\tilde{P}^i$ in $\tilde{N}$, and hence find a contradiction.

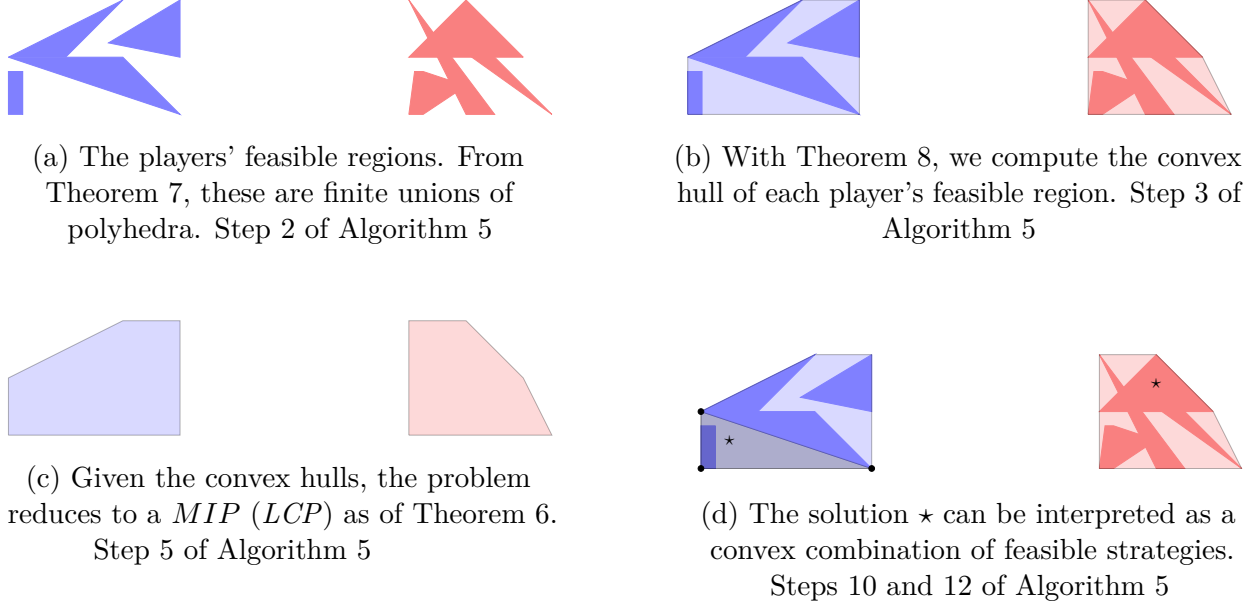


Figure 6.2 A pictorial representation of Algorithm 5.

$$(c^i + C^i \tilde{x}^{-i})^T \tilde{x}^i = \sum_{j'} \sum_{j=1}^{k_i} p_{j'}^{-i} p_j^i (c^i + C^i \hat{x}_{j'}^{-i})^T (\hat{x}_j^i) \quad (6.9)$$

$$\geq \sum_{j'} \sum_{j=1}^{\ell_i} p_{j'}^{-i} \dagger p_j^i (c^i + C^i \tilde{x}_{j'}^{-i})^T (\dagger \hat{x}_j^i) \quad (6.10)$$

$$= \left(c^i + C^i \left(\sum_{j'} p_{j'}^{-i} \tilde{x}_{j'}^{-i} \right) \right)^T \left(\sum_{j=1}^{\ell_i} \dagger p_j^i \dagger \hat{x}_j^i \right) \quad (6.11)$$

$$= (c^i + C^i \tilde{x}^{-i})^T \left(\sum_{j=1}^{\ell_i} \dagger p_j^i \dagger \hat{x}_j^i \right) \quad (6.12)$$

The result of (6.12) follows by plugging the profitable deviation into (6.8), and exploiting its linearity. Since we have a profitable deviation for the mixed strategy for N , a unilateral deviation from $\tilde{x}$ exists for N . This contradicts the fact that $\tilde{x}$ is a *PNE* for N . Therefore, such a deviation cannot exist.

Proof of Statement (ii) . To prove this statement, we prove its contrapositive. Namely, we show that if N has an *MNE*, then Step 5 obtains a *PNE* for $\tilde{N}$ and will not return failure. Therefore, it is sufficient to show that $\tilde{N}$ has a *PNE*. Let the *MNE* of N be given by each player $i \in [n]$ playing $x_1^i, \dots, x_{k_i}^i$ with probability $p_1^i, \dots, p_{k_i}^i$, respectively. Let $\tilde{x}^i = \sum_{j=1}^{k_i} p_j^i x_j^i$ be the

a feasible pure-strategy for player i . It follows that $(\tilde{x}^1, \dots, \tilde{x}^n)$ is a feasible pure-strategy for $\tilde{N}$, and we now show it is indeed a *PNE* for $\tilde{N}$. Given the above *MNE* for N , we know that

$$\sum_{j'} \sum_{j=1}^{k^i} p_{j'}^{-i} p_j^i (C^i x_{j'}^{-i} + c^i)^T x_j^i \leq \sum_{j'} p_{j'}^{-i} (C^i x_{j'}^{-i} + c^i)^T \bar{x}^i, \forall \bar{x}^i \in \mathcal{F}_i.$$

Due to the linearity of the objective function, it follows that:

$$(C^i \tilde{x}^{-i} + c^i)^T \tilde{x}^i \leq (C^i \tilde{x}^{-i} + c^i)^T \bar{x}^i \quad \forall \bar{x}^i \in \mathcal{F}_i. \quad (6.13)$$

If (6.13) holds for all $\bar{x}^i \in \text{cl conv}(\mathcal{F}_i)$, for all i , then $\tilde{x}$ is a *PNE* of $\tilde{N}$ and the proof will be complete. First, we show that (6.13) holds for $\bar{x}^i \in \text{conv}(\mathcal{F}_i)$. Let $\bar{x}^i = \sum_{j=1}^{\ell} \lambda_j \bar{x}_j^i$, where $\bar{x}_j^i \in \mathcal{F}_i$ and $\lambda_j \geq 0$ and $\sum_{j=1}^{\ell} \lambda_j = 1$. Now consider the ℓ inequalities of (6.13), each one for $\bar{x}_j^i$ for $j = 1, \dots, \ell$. Multiply these inequalities by non-negative λ_j on both sides, and add to obtain

$$\begin{aligned} (C^i \tilde{x}^{-i} + c^i)^T \tilde{x}^i &\leq \sum_{j=1}^{\ell} \lambda_j (C^i \tilde{x}^{-i} + c^i)^T \bar{x}_j^i \\ &= (C^i \tilde{x}^{-i} + c^i)^T \bar{x}^i. \end{aligned}$$

In the second instance, to show the same holds for $\bar{x}^i \in \text{cl conv}(\mathcal{F}_i)$, consider a convergent sequence $\bar{x}_1^i, \bar{x}_2^i, \dots$ with each $\bar{x}_j^i \in \text{conv}(\mathcal{F}_i)$ and $\lim_{j \rightarrow \infty} \bar{x}_j^i = \bar{x}^i$:

$$\begin{aligned} (C^i \tilde{x}^{-i} + c^i)^T \tilde{x}^i &\leq (C^i \tilde{x}^{-i} + c^i)^T \bar{x}_j^i \quad \forall j = 1, 2, \dots \\ \implies \lim_{j \rightarrow \infty} (C^i \tilde{x}^{-i} + c^i)^T \tilde{x}^i &\leq \lim_{j \rightarrow \infty} (C^i \tilde{x}^{-i} + c^i)^T \bar{x}_j^i \\ \implies (C^i \tilde{x}^{-i} + c^i)^T \tilde{x}^i &\leq (C^i \tilde{x}^{-i} + c^i)^T \left(\lim_{j \rightarrow \infty} \bar{x}_j^i \right) \\ &= (C^i \tilde{x}^{-i} + c^i)^T \bar{x}^i. \end{aligned}$$

Thus, (6.13) holds for all $\bar{x}^i \in \text{cl conv}(\mathcal{F}_i)$, and $\tilde{x}$ is indeed a *PNE* of $\tilde{N}$. $\square$

Remark 3. Within the proof of Theorem 14, we never exploit any specific properties of simple Stackelberg games. The only assumption we leverage is that the problem is a linear Nash game (i.e., the objective of each player is of the form $(c^i + C^i x^{-i})^T x$). In this case, it is sufficient to solve the problem for *PNE* in the convex hull of each player's feasible set to compute an *MNE* for the original problem. In this spirit, if one can compute the convex hull of the player's feasible region, and if objectives are linear, then every game is a convex game.

6.5 Enhancing the Algorithm

In this section, we present two enhancements of Algorithm 5. In Section 6.5.1 we introduce an iterative procedure to approximate the closure of the convex hull of each player feasible set. Thus, we avoid a possibly costly (and arguably unnecessary) enumeration of all the polyhedra defining the feasible sets. In Section 6.5.2, we tailor the algorithms to specifically retrieve *PNEs*, as opposed to general *MNEs*.

6.5.1 Inner Approximation Algorithm

While Algorithm 5 is guaranteed to terminate and solve the problem, we introduce a procedure that can improve computational tractability. The feasible region of a simple Stackelberg game is a finite union of polyhedra (see Theorem 7), and Theorem 8 gives their convex hull. However, since there may be exponentially many polyhedra, the convex hull description could become untractably large. Algorithm 5 intensively leverage on the complete enumeration of such polyhedra in Step 2. The central intuition is to limit the enumeration by iteratively refining the convex hull's description for each player. This procedure is also valid for an individual Stackelberg game or a bilevel program. However, its importance is more relevant when dealing with *NASPs*, where the computation of this convex hull is essential. The key components of this approach are the *polyhedral relaxation* of the set S defined in (6.7), and the concept of *selected polyhedron*.

Definition 20 (Polyhedral relaxation). *The polyhedral relaxation of the set S defined in (6.7) is given by the set $\mathcal{O}_0 = \{x : Ax \leq b, z = Mx + q, x_i \geq 0, z_i \geq 0 \ \forall i \in \mathcal{C}\}$*

Clearly, this set contains $\text{cl conv}(S)$ and is a polyhedron by construction, even though if S is not generally a polyhedron.

Definition 21 (Selected polyhedron). *Let $b \in \{0, 1\}^{|\mathcal{C}|}$ and let $\mathcal{C} = \{c_1, \dots, c_k\}$. Then, the selected polyhedron corresponding to b is $\mathcal{P}(b) = \{x_{c_i} \leq 0, \forall i \in \{i : b_i = 0\}\} \cap \{[Mx + q]_{c_i} \leq 0, \forall i \in \{i : b_i = 1\}\}$.*

We can then formally define the concept of inner approximation.

Definition 22 (Inner Approximation). *Let $J = \{j^1, \dots, j^\ell\} \subseteq \{0, 1\}^{m_f}$. Then the inner approximation defined by J is $\mathcal{I}_J = \text{cl conv}(\bigcup_{b \in J} \mathcal{P}(b) \cap \mathcal{O}_0)$.*

Remark 4. *The size of the extended formulation of $\mathcal{I}_J$ is bounded by $O(|J|)$. To ensure a perfect description, we need a choice of $J = \{0, 1\}^{|\mathcal{C}|}$. However, $|J| = 2^{|\mathcal{C}|}$ and a description*

of $\text{cl conv}(S)$ will be exponentially large. Unless $\mathcal{P} = \mathcal{NP}$, there cannot be any asymptotical improvements [10].

Algorithm 6 presents the *inner approximation algorithm* – an enhancement to Algorithm 5 – to retrieve an *MNE* for *NASPs*. First, it iteratively constructs an increasingly accurate inner approximation of the players' feasible regions. Then, the algorithm looks for a *PNE* in a restricted game $\tilde{N}$, namely a game where each player's feasible region may be inner approximated (Step 4).

Let $\mathcal{F}_1, \dots, \mathcal{F}_n$, be the inner approximations of the feasible sets of player $1, \dots, n$. One can compute the convex hull's closure for each approximation and solve the associated facile Nash game $\tilde{N}$. If $\bar{x}$ is a Nash equilibrium of $\tilde{N}$, the algorithm checks if $\bar{x}$ – or the associated mixed-strategy implied by $\bar{x}$ (similarly to Step 12 of Algorithm 5) – is a Nash equilibrium for the original game N . If this is the case, then the algorithm terminates and returns the equilibrium. Conversely, if this mixed-strategy is not an *MNE* of N , there exists a profitable deviation $\hat{x}^i$ for some players such that $\hat{x}^i \notin \mathcal{F}_i$. In this case, we refine the inner approximation of i -th player's feasible set by adding a polyhedron containing $\hat{x}^i$. At each iteration of the algorithm, we keep on adding polyhedra containing the profitable deviations. However, $\tilde{N}$ may not have a *PNE* in a given iteration (Step 4). In this case, we gain no additional knowledge about which polyhedra to add to the inner approximation. Therefore, we arbitrarily add one or more polyhedra to the feasible region of *each* player in the problem, keeping the algorithm running. We define as the *extension strategy* the criteria by which we select such polyhedra.

In optimization problems, a point contained in an inner approximation of the feasible set is feasible for the original problem and provides a *primal bound* for the original problem. However, this is not true in the case of a Nash game. In Remark 5 below, we show that the inner approximation game might have an *MNE* while the original game does not. Conversely, we also show that the original game might have an *MNE* while an inner approximation of the game does not.

Remark 5 (Inner approximation $\tilde{N}$ might have an *MNE* but N might not). *There might be cases where the inner approximation has no MNE, but the original NASP does. Consider the following players' problems and their inner approximation.*

$$\textbf{Latin Player:} \quad \min_x \{\xi x : x \in \mathbb{R}, x \geq 0\} \quad (6.14a)$$

$$\begin{aligned} \textbf{Greek Player:} \quad & \min_{\xi, \chi} \{x\xi : \xi \in [-5, 5]; \chi \geq 0; \\ & \chi \in \arg \min_{\chi} \{\chi : \chi \geq \xi - 1; \chi \geq -\xi - 1\}\} \end{aligned} \quad (6.14b)$$

Algorithm 6: Inner approximation to obtain an *MNE* for a *NASP*

Data: A description of *NASP* $N = (P^1, \dots, P^n)$ and $J = (J^1, \dots, J^n)$ where $J^i \subseteq \{0, 1\}^{|\mathcal{C}_i|}$ where $\mathcal{C}_i$ is the set of indices of complementarity ($\perp$) conditions for the i -th player

Result: For each $i = 1, \dots, n$, $\hat{x}_j^i$ is a pure-strategy played with probability p_j^i , presenting a mixed-strategy with support size k^i

```

1 Procedure InnerApproxNash( $N, J$ ) ()
2    $\hat{\mathcal{F}}_i \leftarrow$  inner approximation defined by  $J^i$  and  $\hat{\mathcal{F}}_i \leftarrow \text{cl conv } \hat{\mathcal{F}}_i$ ;
3    $\tilde{P}^i \leftarrow$  objective function of  $P^i$  and a feasible set  $\tilde{\mathcal{F}}_i$ ;
4   Solve the facile Nash game  $\tilde{N} = (\tilde{P}^1, \dots, \tilde{P}^n)$  to obtain solution  $\bar{x}$ ;
   /* May return no  $\bar{x}$  */
5    $\hat{x}^1, \dots, \hat{x}^n \leftarrow \text{getDeviation}(P, \bar{x})$ ;
6   if  $\hat{x}^i = \text{NULL}$  for all  $i = 1, \dots, n$  then
7     return  $\bar{x}$ 
8   for  $i = 1, \dots, n$  do
9     if  $\hat{x}^i \neq \text{NULL}$  then
10       $\tilde{b}^i \leftarrow$  binary encoding of a polyhedron containing  $\hat{x}_i$ .  $J^i \leftarrow J^i \cup \tilde{b}^i$ ;
11 return InnerApproxNash( $N, J$ )

```

Using *KKT* conditions on the follower's problem, the Greek's problem can be rewritten as

$$\min_{\xi, \chi, \mu} \left\{ x\xi : \xi \in [-5, 5]; \mu_1 + \mu_2 = 1; \chi \geq 0; \begin{array}{l} 0 \leq \mu_1 \perp \chi - \xi + 1 \geq 0 \\ 0 \leq \mu_2 \perp \chi + \xi + 1 \geq 0 \end{array} \right\}.$$

The polyhedra $P(b)$ corresponding to $b = (0, 0)$, and $b = (1, 1)$ are empty. The remaining two polyhedra can be projected to the ξ space as $[-5, -1] \cup [1, 5]$. We claim that the problem in (6.14) has no Nash equilibrium. This is because, irrespective of the Latin player's decision, an optimal decision for the Greek player is $\xi = -5$. For such a value of ξ , the Latin player has an unbounded objective. Consider the inner approximation due to the choice $J = \{(0, 1)\}$. The equivalent programs are as follows.

$$\textbf{Latin Player:} \quad \min \{ \xi x : x \in \mathbb{R}, x \geq 0 \} \quad (6.15a)$$

$$\textbf{Greek Player:} \quad \min \{ x\xi : \xi \in \mathbb{R}, \xi \in [1, 5] \} \quad (6.15b)$$

In (6.15), the inner approximation is exact for the Latin player and is a strict inner approximation for the Greek player. However, (6.15) has a PNE $(\xi, x) = (0, 1)$.

Conversely, it can also happen that the original *NASP* has no *MNE*, but the inner approximation does. For such an example, replace the objective of the Greek player in (6.14) with

a minimization of $-x\xi$, and the corresponding inner approximation of the Greek player in (6.15) with $\xi \in [-5, -1]$. This inner approximation game has no Nash equilibrium. However, the original game has a Nash equilibrium of $(\xi, x) = (0, 5)$.

6.5.2 Enhancements for *PNEs*

In specific applications, users tend to prefer deterministic strategies over randomized ones. Thus, one necessarily requires a *PNE* or to show that no *PNE* exists. With this motivation, we alter Algorithm 5 to retrieve *PNEs* specifically or prove no *PNE* exists.

Enumeration for *PNE*. This algorithm is similar to Algorithm 5, hence we assume the same notation. First, the procedure explicitly enumerates all the polyhedra in the feasible region of each player, and computes their convex hull. In addition, it introduces in $\tilde{N}$ a set of binary variables forcing the equilibrium strategy, for each player, to be strictly in the original feasible region rather than solely in the convex hull. From Theorem 7, the feasible region for each *NASP*'s player is a finite union of polyhedra. Let the feasible region of the i -th leader be $\mathcal{F}_i = \bigcup_{j=1}^{g_i} P_j^i$, where $P_j^i = \{A_j^i x \leq b_j^i\}$ is a polyhedron. Moreover, Theorem 8 gives $\text{cl conv}(\mathcal{F}_i)$ as $A_j^i x_j^i \leq b_j^i \delta_j^i$ for $j \in [g_i]$, $x^i = \sum_{j=1}^{g_i} x_j^i$, and $\sum_{j=1}^{g_i} \delta_j^i = 1$. If for some j , $\delta_j^i = 1$, then the projection x is strictly in the polyhedron P_j^i . Since we can reformulate a *NASP* as a MIP feasibility problem, we enforce a new set of constraints in $\tilde{N}$ requiring each δ_j^i to be binary in $\tilde{N}$. Hence, each *PNE* for $\tilde{N}$ is also a *PNE* for N , and if $\tilde{N}$ has no *PNE*, also N has no *PNE*. In addition, for the equivalence between *PNEs* in $\tilde{N}$ and N , the condition of N being a *NASP* can be relaxed. In particular, it is sufficient that leaders' objectives in N are convex – observe that under this case, the reasoning in the proof of statement (ii) for Theorem 14 directly follows. We refer the reader to appendix B.3 for the pseudocode of this procedure.

6.6 Computational Tests

We test our algorithms² with the energy-trade model (B.3) and (B.4).

The Model. We consider different geographical regions, where governments of such regions act as leaders. Governments determine the energy export/import amount (of energy) and the CO₂ taxation scheme imposed on their respective followers (energy producers). Each country seeks to minimize the sum of three components: (i) the product between each

²Full implementation with detailed documentation are available on <https://github.com/ds4dm/ZERO>

follower’ production and the emission cost (e.g., the social cost of carbon, SCC), (ii) the product between import price and quantity to any other country, (iii) the negative product between import price and quantity of any other country, namely the maximization of export revenues. Besides, countries may also include a negative (maximized) tax-revenue term in their objectives, namely the sum of all their respective followers’ taxes. We distinguish between three forms of taxation: (i) *Standard-Taxation*, where each follower has a possibly different tax per unit-energy produced, (ii) *Single-Taxation*, where every follower has the same tax per unit-energy produced, (iii) *Carbon-Taxation*, where every follower has the same tax per unit-emission. The lower-level players are energy producers deciding the amount of production of their plants based on their linear and quadratic unit costs and their leader’s taxation levels. Specifically, followers are playing a Cournot game where the homogeneous good is the amount of energy produced.

Instances. We propose three sets of computational instances (InstanceSet *A*, *B*, and *Insights*), and a case study on a real-world inspired Chile-Argentina instance ³. The goal of our computational tests is twofold. On the one hand, we showcase our algorithms’ compelling computational capabilities and compare their performances. With this respect, we extensively test all our algorithms on the instance sets *A* and *B*. On the other hand, we derive managerial insights from our models’ solutions, focusing on the Chile-Argentina case study and the instance set *Insights*.

Data generation. We synthetically generate our instance sets as follow: (i) *InstanceSet A* contains 150 instances with 3 to 5 countries and up to 3 followers per country, (ii) *InstanceSet B* contains 50 instances with 7 countries and up to 3 followers per country, (iii) *InstanceSet Insights* contains 50 instances with 2 countries and 3 followers per country.

We randomly draw each of the followers from three classes of producers: highly-polluting (e.g., coal, oil), averagely-polluting (e.g., gas), and green (e.g., renewables such as solar, hydro). Their emission costs per unit-energy (e.g., GWh) takes an integer value in the range [300, 500], [100, 200], and [25, 50], respectively. These are USD values of emission assuming a social cost of carbon at USD 25 per tonne of CO₂ equivalent and typical emission values in these technologies. We set linear and quadratic production costs – negatively correlated to the emission factors – in the respective ranges [150, 300], and [0, 0.6] for unit energy. The production capacities are discrete unit-energies in the interval [50, 20000]. We refer the reader to appendix B.4 for a more detailed review of the parameters.

³All instances are available on <https://github.com/ds4dm/EPECInstances>

6.6.1 Strategic Insights

Starting from *InstanceSet Insights*, we solve each instance 4 times by testing a discrete grid of 2 parameters. The first one is the *Carbon-Taxation* revenue in every country’s objective, while the second dictates whether trade among countries is allowed. Table B.4 provides comprehensive results. We attempt to answer the following strategic questions:

- (i) **Tax policy.** Are countries reducing further their emissions if they consider the carbon tax as a source of income?
- (ii) **Trade policy.** How does competitive energy trade among countries affect global emission?

Tax policy. Some literature argues that carbon tax revenues can further help reduce carbon emission, spur greener technologies (e.g., carbon sequestration, electric vehicles), or even meet other governmental expenses [2, 105, 119]. One might instinctively think that an income-hungry (e.g., GDP) government could levy a more aggressive carbon tax policy if that could be a revenue source and help reduce emissions. However, we observe the opposite to be true. We consistently find that when the government’s objective ($b = 1$ in (B.3)) model incomes through a carbon tax, the government is systematically incentivized to impose a smaller tax. With smaller tax rates, coal and natural gas production are more significant. Thus, this increases the governmental revenue, which is the product of production and tax per unit of emission. In summary, decreased carbon tax could give increased revenue for the government. However, emissions are decreased compared to the no-taxation scheme but increased compared to the case when the government does not look for revenue from these taxes.

In particular, in 40 out of the 50 test instances, both countries’ total emission was greater if the individual governments considered the objective’s tax revenue. On an absolute basis, emissions were about 13.5% more on average when governments imposed taxes, keeping the revenue in their objective. A statistical t-test rejects the null hypothesis that the global emissions are equal with and without the countries considering carbon tax as a revenue source with a p -value of 0.00018.

We also observe that the trade is lesser in 30 out of 50 instances and, on average, about 7.8% lesser when the countries consider tax as a revenue source. However, a similar t-test does not suggest enough evidence to reject the null hypothesis (p -value = 0.29) that the traded quantities in the two cases have the same population mean.

Trade policy. Second, we observe that the tax rate is typically lesser when countries can trade. Quantitatively, we find that the average tax rate without trade is about 12.9% higher than when trade exists between the countries. However, we also observe that in 63 of the 100 possible cases (50 instances in *InstanceSet Insights* with two countries each), the tax rate is higher if there is trade between them. In other words, tax is slightly higher in many instances when the trade is enabled. Nevertheless, in those instances where the tax rate is lower with trade enabled, the tax rate is significantly lower.

Next, one might wonder if increased emissions might accompany trade between countries. Since energy trade is an economic activity, one can think it could worsen the externality of emission. However, we observe that emissions are consistently less when countries can trade. Clean means of energy in a different country could fulfill the demand without forcing domestic producers to produce using non-green means of production. Quantitatively, we compare the average emission by both the countries when a trade happens between them instead of no-trade being allowed between them (see Table B.4). We observe that the average emission dropped by about 35.9% when trade was enabled. Further, never in those 50 test instances did the emission ever increase after countries were able to trade. We also note that when countries can trade, emissions could increase in one country, but the decrease in another country is always significant enough to ensure that the total emission decreases while keeping the consumption in both countries roughly the same.

Final comments. Besides our consistent insights that (i) a tax revenue-hungry government might impose a lesser carbon tax than a government inclined to reduce emissions (ii) enabling trade reduces total global emissions, the answers to the more general questions were predominantly instance dependent. In particular, we observed that opening up trade increased domestic carbon taxes in some cases and decreased them in others. We observed similar behaviors for trade with revenue-hungry governments. The answers to these questions were sensitive to the cost, capacity, and emission factors of production units and the domestic energy demand of each country. These observations suggest that one has to solve a *NASP* (or even a more complex model) to identify the specific dynamics for a given situation. Furthermore, in *NASPs*, one can always perform equilibria selection – if multiple *MNEs* exist – by solving the problem with Algorithm 5 and enforcing the *MIP*’s objective to optimize a given criterion.

6.6.2 Case Study – Chile-Argentina Energy Markets

We implemented the model using actual data from Chile’s and Argentina’s electricity markets (specifically from 2018-2019). Electricity trade between these countries started in 2016, with Chile exporting a small amount – close to 1558 MWh – of electricity to Argentina. However, we expect the transfers to increase as both countries signed an energy cooperation agreement in 2019 (for both electricity and gas). These efforts have created some debate regarding electricity prices, which may impact one of the Chilean government’s main goals: make electricity more affordable. Furthermore, both Chile and Argentina have signed the Paris agreement and promised rapid decarbonization of their energy systems. Chile was the first country in Latin America to implement a carbon tax (USD 5/ tCO_2), followed by Argentina, which defined a carbon tax that became operational in 2019. Given this context, this analysis focuses on determining the impacts of an integrated market where electricity trade is viable while each country’s government can define internal carbon tax policies. We model different energy producers in each country. Electricity producers in Chile and Argentina have various technologies. We consider hydro, solar, wind, natural gas, and coal technologies in Chile’s case. Historical data shows that Argentina heavily relies on thermal plants fueled by natural gas and on hydro energy. Technical data for different technologies, obtained from the Chilean *Comision Nacional de Energia* (National Energy Agency) and the US Energy Information Administration, include fuel consumption, capacity factors, and variable costs. We model a stake of coal-based production technology only in Chile and minimal to none in Argentina. We analyzed how the markets react under different renewable sources’ future levels and with/without limits on energy trade imposed between these two countries.

If no trade is allowed (representing current operations), we calibrated the model to match both countries’ historical data. There is a significantly greater demand in Argentina (129 TWh/y) than in Chile (60 TWh/y). Approximately 71% of the generation in Argentina roots in natural gas thermal power plants. Hydro energy fulfills the remaining demand. In Chile’s case, coal and gas power plants have a market share of 42%, hydro accounts for 36%, and renewable sources (solar and wind) supply approximately 15% of the electric demand.

We observe an interesting substitution effect when trade is allowed among countries and install capacities are not varied (existing capacities in both countries). Imports from Argentina replace conventional means of production in Chile (coal and gas). The Chilean government curtails fossil-fueled electricity by increasing the carbon tax faced by such technologies. The opposite effect shows in Argentina, where the government lowers the carbon tax to incentivize electricity generation from natural gas technologies. Such an export hurts the local market. As expected, an increase in exports to Chile yields increased local electricity prices in Argentina,

significantly lowering the indigenous consumption levels.

As observed above, with a possibility for energy trade between the countries, our model predicts that without a significant increase in the renewable capacity on either country or without a significant decrease in carbon's social cost, Argentina's economy could be highly impacted. Therefore, unless cheap (near-zero) renewable sources produce energy in Chile or Argentina, it is expected that trade among countries will remain low.

To assess the likelihood of future trade under large renewable energy deployments, we consider two increased wind and solar capacity cases in Chile. The two scenarios consider capacity additions of 20 GW and 40 GW, respectively [3]. We initially observe that Chile benefits from increased renewable capacity if energy trade is not allowed in these cases. Electricity prices are reduced by 13% when there is an increase of 40 GW, while consumption grows by 20%. Interestingly, Argentina becomes a net importer of electricity when energy trade is allowed as Chile increases its renewable energy capacity. Argentina has net imports of 12 TWh/y when the country installs a 40 GW of renewable capacity. Therefore, Argentina switches from a net exporter (without renewable capacity installed) to a net importer of electricity. This import is a direct result of the availability of cheap energy, which increases the demand.

6.6.3 Speed Analysis

In terms of performance analysis, we focus on *InstanceSet A*, and *B*. An instance is marked a solved if it has an *MNE*, or an algorithm finds a certificate of inexistence, namely, no *MNE* exists. The time limit is $TL = 1800$ seconds. In our implementation, we introduce 3 *extension strategies* for Algorithm 6: given a lexicographic order for each player's polyhedra, k of them are added *sequentially*, *reverse-sequentially*, or *randomly*.

Tables 6.1 and 6.2 summarize the computational results for *InstanceSetA* and *InstanceSetB*, respectively. The upper parts of the tables reports results for the full enumeration Algorithm 5 (*FE*) and Algorithm 6 (*InnerApp*), where an *MNE* solves the instances. In the lower part of the table, we specifically look for *PNEs* with the enhanced algorithm presented in Section 6.5.2 (*FE-P*). In the third column, if the algorithm is the inner approximation, we highlight the *extension strategies*, and the relative parameter k in the following column. Fifth, sixth, and seventh columns are, respectively, average time when: (i) an *MNE* is found (*EQ*), (ii) the algorithm returns a certificate of non-existence (*NO*) and (iii) for all instances. In the eighth and ninth column, we report the number of times the row's algorithm outperforms all the others, namely wins in terms of computing times. Finally, the tenth column reports how many instances do not trigger the time limit.

Table 6.1 Results summary of different algorithmic configurations for *InstanceSetA*.

Algorithm	ES	k	Time (s)			Wins		Solved	
			EQ	NO	All	EQ	NO		
FE	-	-	29.08	0.12	120.21	6	82	140/149	
MNE	$InnerApp$	Seq	1	6.65	0.35	51.33	3	0	145/149
		Seq	3	17.76	0.18	55.82	5	0	145/149
		Seq	5	6.40	0.15	51.08	3	0	145/149
		Rev.Seq	1	7.97	0.36	3.73	26	0	149/149
		Rev.Seq	3	11.29	0.18	53.12	4	0	145/149
		Rev.Seq	5	9.53	0.15	76.41	5	0	143/149
		Random	1	5.22	0.36	26.60	8	0	147/149
		Random	3	32.42	0.18	85.65	5	0	143/149
		Random	5	23.67	0.15	58.26	2	0	145/149
PNE	$FE-P$	-	-	7.25	0.12	328.23	-	-	122/149

Table 6.2 Results summary of different algorithmic configurations for *InstanceSetB*.

			Time (s)			Wins			
Algorithm	ES	k	EQ	NO	All	EQ	NO	Solved	
	FE	-	-	260.29	1.12	1174.32	0	2	20/50
MNE	$InnerApp$	Seq	1	39.26	9.64	672.24	1	0	32/50
		Seq	3	62.66	3.88	616.25	1	0	34/50
		Seq	5	24.03	2.83	733.97	1	0	30/50
		Rev.Seq	1	171.47	9.66	262.74	27	0	47/50
		Rev.Seq	3	13.85	3.86	585.27	4	0	34/50
		Rev.Seq	5	78.57	2.83	798.90	6	0	29/50
		Random	1	34.65	9.65	497.06	0	0	37/50
		Random	3	123.02	3.87	588.03	2	0	36/50
		Random	5	39.18	2.86	711.77	4	0	41/50
PNE	$FE-P$	-	-	7.36	1.12	1441.95	–	–	10/50

For *MNEs*, *InnerApp* achieves better performances than *FE*, being on average 2x faster on all instances, and up to 30x when an *MNE* exists (see *InnerApp*-RevSeq-1 in Table 6.1). Table 6.2 shows the full potential of *InnerApp*, which remarkably reduces computational times compared to *FE*. Especially, *InnerApp* can solve almost all the 50 hard instances compared to the 20 solved by *FE*. Besides, when no equilibrium exists, *InnerApp* will always terminate at its last iteration, namely the one corresponding to *FE*. It is not surprising that *FE* returns a non-existence certificate always faster than *InnerApp*. Both the algorithms *InnerApp* and *FE* – when asked to retrieve a generic *MNE* – may return a *PNE*. This happens 37.6%, and 30.4% within *InstanceSetA* and *InstanceSetB*, respectively. Hence, there is a natural need for *FE-P*.

6.7 Concluding Remarks

Our theoretical and computational framework tackles *NASPs*, where players of a Nash game solve linear bilevel programs, and each leader can have several followers playing a *simple* Nash game among themselves. We show that deciding on the existence of *PNE* and *MNE* for *NASPs* is Σ_2^P -hard, and we provide a family of algorithms to find *MNEs* as well as *PNEs* for the problem. Furthermore, we show it is sufficient to compute an *MNE* over the convex hull of each player's feasible region to retrieve a *MNE* for the original problem. This work expands our knowledge of algorithmic approaches to compute equilibria, in particular *MNEs*, by using theory and tools from Integer Programming and Optimization. In addition to a theoretical characterization of these algorithmic methods, we analyze their practical efficiency, settle their limitations, and opens up new future directions by establishing a solid benchmark for future progress. From an application standpoint, we demonstrated how the *NASPs* framework could help unveil counterintuitive consequences of policymaking within the context of international energy trade.

In terms of future work, both the computation of multiple equilibria or their selection according to some specified criteria are interesting interrogatives. Furthermore, it may be worth developing procedures to prune parts of the feasible regions (e.g., polyhedra) not in the support of any equilibrium. This last direction may considerably speed up the equilibria computation in *NASPs* and Stackelberg games. Finally, advancements on these proposed research lines may lead to further methodological developments to tackle other classes of hierarchical games. For instance, multi-leader multi-followers games where followers from different leaders can directly interact.

CHAPTER 7 ARTICLE 2: ZERO: PLAYING MATHEMATICAL PROGRAMMING GAMES

Authors: Gabriele Dragotto, Sriram Sankaranarayanan, Margarida Carvalho and Andrea Lodi.

Submitted to INFORMS Journal on Computing¹.

Abstract We present *ZERO*, a modular and extensible C++ library interfacing Mathematical Programming and Game Theory. *ZERO* provides a comprehensive toolkit of modeling interfaces and algorithms for Reciprocally Bilinear Games (*RBGs*), i.e., simultaneous non-cooperative games where each player solves a mathematical program with a linear objective in the player’s variable and bilinear in its opponents’ variables. This class of games generalizes the classical problems of Operations Research to a multi-agent setting. *ZERO* modular structure gives users all the elementary ingredients to design new game-theoretic models and algorithms for *RBGs*, and find their Nash equilibria. The library provides additional extended support for integer non-convexities, linear bilevel problems, and linear equilibrium problems with equilibrium constraints. We provide an overview of the software’s key components and showcase a Knapsack Game, i.e., a game where each player solves a binary knapsack problem. Aiming to boost practical methodological contributions at the interplay of Mathematical Programming and Game Theory, we release *ZERO* as open-source software. Source code, documentation and examples are available at www.getzero.one.

7.1 Why Games and Equilibria?

The pioneering book from Morgenstern and Von Neumann [115] and the seminal papers from Nash [116, 117] transformed the scientific perspective on strategic behavior. The ubiquitous concepts of Nash equilibrium and rationality are now cornerstone concepts in Game Theory, with applications ranging from Economics to Social Sciences. The growing interest in game dynamics in the Operations Research community reflects a need to extend classical decision-making frameworks to multi-agent settings that can account for interactions among multiple decision-makers. The community devoted particular interest – to name a few – to bilevel programming (e.g., Basu et al. [13], Caprara et al. [26], DeNegre and Ralphs [54], Fischetti et al. [72], Hu and Ralph [89], Kleinert et al. [94], Labbé and Violin [99]) and its application in electricity markets and network pricing problems [21, 67, 100],

¹A pre-print is available in [58].

equilibrium problems with equilibrium constraints [31, 110], and more recently to integer programming games [29, 32, 33, 47, 57, 82, 95]. On the one hand, such empowering modeling capabilities unquestionably offer a tempting opportunity for extending the domain of influence of Operations Research. Arguably, multi-agent optimization frameworks can help provide enhanced models by contemplating the interactions decision-makers often take by pondering the influence of other stakeholders (e.g., other players). Additionally, they can help embed socially-beneficial outcomes by enlightening the nature of interaction among selfish decision-makers. For instance, Carvalho et al. [31] provide insights on the role of a carbon tax in competitive international energy markets, Carvalho et al. [28] prove that the most rational outcome in their cross-border kidney exchange maximizes the social welfare (e.g., the sum of the objectives of all players). On the other hand, multi-agent models are as helpful as one can efficiently compute equilibria (or equivalent solution paradigms), thus highlighting the importance of theoretical and practical contributions for computing them. We believe that free and open-source software can foster experimentation in both practitioners' and researchers' communities, and hopefully lead to novel methodological advancements in the field.

7.1.1 Background.

In this context, we introduce *ZERO*, a modular C++ package to handle Reciprocally-Bilinear Games (*RBGs*), a special class of Mathematical Programming Games (*MPGs*). An *MPG* is a simultaneous game among n players, each of which solves a mathematical program whose objective function is parametrized in other players' variables, and whose feasible region's description does not include other players' variables. Although *MPGs* are also Nash equilibrium problems (*NEPs*) [63], the *MPGs* taxonomy we propose follows three assumptions: (i.) a set of constraints, for instance, a set of linear constraints and integer requirements, represent each player's moves. This set may be unbounded, contain infinitely or finitely many elements, and generally does not have a special structure. We do not assume the players' feasible sets to be continuous (i.e., in contrast to most of the *NEPs* literature), nor that computing equilibria necessarily requires the solution of a complementarity problem, (ii.) we aim to build a language intersecting both elements of Game Theory and Mathematical Programming, (iii.) we aim to preserve the structure that constraints give to each player's problem. For instance, we may not drop any constraints to simplify the game without damaging its modeling capability. For the above three reasons, we introduce the class of *MPGs* to represent a wide variety of games among optimization problems.

ZERO provides support for a fundamental class of *MPGs*, namely the class of *RBGs*. Let the

operator $(\cdot)^{-i}$ define $(\cdot)$ except i ; e.g., if $x = (x^1, \dots, x^n)$, then $x^{-2} = (x^1, x^3, \dots, x^n)$.

Definition 23 (Reciprocally-Bilinear Game [33]). *A Reciprocally-Bilinear Game (RBG) is an MPG among n players, where each player $i = 1, 2, \dots, n$ solves the optimization problem*

$$\min_{x^i} f^i(x^i, x^{-i}) = (c^i)^\top x^i + (x^{-i})^\top C^i x^i \quad (7.1a)$$

$$\text{s.t. } x^i \in \mathcal{X}^i \quad (7.1b)$$

where $\mathcal{X}^i \subseteq \mathbb{R}^{m_i}$, and C and c are a matrix and a vector of appropriate dimensions, respectively. An RBG is polyhedrally-representable if $\text{cl conv}(\mathcal{X}^i)$ is a polyhedron for each i , and one can optimize a linear function over each $\mathcal{X}^i$.

In RBGs, the i -th player objective function $f^i(x^i, x^{-i})$ – or payoff function for i – is linear in x^i and contains bilinear products with x^i and x^{-i} . Further, since RBGs are MPGs, the description of each player's feasible region $\mathcal{X}^i$ does not contain other players' variables, and the i -th player optimization problem is *parametrized* in x^{-i} , namely plugging x^{-i} as a parameter results in an optimization problem purely in the variables x^i . When $n = 1$, the RBG in Definition 23 is a single optimization problem in x^i . Whenever $n > 1$, RBGs become expressive models extending typical Operations Research tasks – such as resource allocation, scheduling, or routing – to a multi-agent setting. Consider, for instance, the emblematic 0/1 Knapsack Problem; given a set of items, a decision-maker selects some of them to maximize the sum of the profits associated with each item, subject to a capacity constraint. A multi-agent extension of this problem is the so-called Knapsack Game as in Example 5, where n players simultaneously solve a 0/1 Knapsack Problem.

Example 5 (Knapsack Game). *A Knapsack Game is an RBG where each player solves the optimization problem*

$$\max_{x^i} \{(c^i)^\top x^i + (x^{-i})^\top C^i x^i : (a^i)^\top x^i \leq b^i, x^i \in \{0, 1\}^{m_i}\} \quad (7.2)$$

where m_i is the number of items for player i , $b^i \in \mathbb{Z}$, $a^i \in \mathbb{Z}^{m_i}$, $c^i \in \mathbb{Z}^{m_i}$, and C^i is an integer-valued matrix of appropriate size.

In this game, player i has not only to consider a feasible packing of items maximizing the profits associated with the vector c^i , but has to look out for the positive or negative impact of the interaction of its packings with the ones of its opponents (the C^i products). Besides being an RBG, the Knapsack Game is also an Integer Programming Game (IPG), namely an MPG where each player solves a mixed-integer problem [95]. The sets $\text{cl conv}(\mathcal{X}^i)$ are the so-called

integer hulls associated with each player's 0/1 knapsack polytope, and each point $\bar{x}^i \in \mathcal{X}^i$ is a *pure-strategy* for i , namely a solution to the knapsack problem for i . In general, each $\tilde{\sigma}^i \in \text{clconv}(\mathcal{X}^i)$ is a *mixed-strategy*, namely a point inside the 0/1 knapsack polytope. The central question is then to determine what is a solution to the above game. In an optimization problem, we usually search for an optimal solution that maximizes (minimizes) the objective function while fulfilling the constraints. However, in a game, a solution should be *stable*, meaning that it should be mutually optimal for all the players, and not only a subset of them. The most famous solution paradigm in Game Theory is the one of Nash Equilibrium, a solution where each player cannot *unilaterally* deviate from it while improving its payoff. We formally define the concept of Nash equilibrium for RBGs in Definition 24; we remark that in Definition 23 players are minimizing their objective functions, and improving a payoff means decreasing it.

Definition 24 (Pure-Strategy Nash Equilibrium). *A strategy profile $\bar{x} = (\bar{x}^1, \dots, \bar{x}^n)$ is a Pure-Strategy Nash Equilibrium for an RBG as in Definition 23 if, for each player i and strategy $\tilde{x}^i \in \mathcal{X}^i$, then $f^i(\bar{x}^i, \bar{x}^{-i}) \leq f^i(\tilde{x}^i, \bar{x}^{-i})$.*

In other words, at the equilibrium $\bar{x} = (\bar{x}^1, \dots, \bar{x}^n)$, no player i can possibly pick a strategy $\tilde{x}^i \neq \bar{x}^i$ so that $f^i(\bar{x}^i, \bar{x}^{-i}) > f^i(\tilde{x}^i, \bar{x}^{-i})$. In this sense, the equilibrium strategy is resilient to the moves of each player's opponents and provides a mutually-optimal solution. The *Mixed-Strategy Nash equilibrium* relaxes the definition of Pure-Strategy Nash equilibrium by allowing players to select not only pure-strategies, but in general mixed-strategies.

7.2 Our Contributions

ZERO provides advanced and modular C++ toolkits to formulate RBGs and compute their Nash equilibria, with high-level APIs for practitioners and low-level ones for researchers and experienced users. We summarize the most important contributions as follows.

- (i.) ZERO is the first library to support non-cooperative simultaneous games where players solve mathematical programs. Other Game Theory solvers, such as Gambit [113] only support finite games in normal form (games with finitely many players, finitely many strategies and outcomes).
- (ii.) The library has a modular structure designed for allowing extensibility. Each component – or *module* – independently performs a specific task and interacts with the others through well-defined interfaces. For instance, the natively embedded algorithms interface with the base modules allowing the development of sophisticated computational routines.

Users can either use the included algorithms or implement custom ones depending on the desired level of control.

- (iii.) The library is an abstract layer bridging typical Mathematical Programming and Game Theory and focuses on the interaction and orchestration among external libraries and native modules. We delegate most of the standard mathematical programming routines to specialized software, thus integrating popular and well-maintained tools available in the Operations Research community. For instance, we solve mathematical programs through Gurobi [83] and PATH [70], we generate cutting planes with Coin-OR Cgl [109], and we perform linear algebra operations through Armadillo [132].
- (iv.) ZERO can work as an off-the-shelf solver for *RBGs* without the need for a deep technical understanding of the algorithmic details. We provide a series of high-level interfaces designed specifically for some classes of *RBGs*, along with standardized instance file schemes and plug-and-play shell executables. On the one side, ZERO provides high-level APIs for practitioners and industrial parties to experiment with our high-level APIs. On the other side, we target experienced users by offering advanced tools to build sophisticated models and algorithms.

7.3 Overview

We briefly give an overview of ZERO: the detailed documentation for the software is available online at www.getzero.one. Our library currently supports any polyhedrally-representable *RBG*, and further provides additional tools (i.e., high-level modeling APIs) for two specific types of games. First, *IPGs*, namely *MPGs* where each player solves an integer program; in particular, ZERO supports *IPGs* that are also *RBGs*, and hence have a bilinear objective as in Definition 23. Second, Nash games Among Stackelberg Players (*NASPs*), a class of Equilibrium Problems with Equilibrium Constraints among the leaders of continuous bilevel games [31].

Modules and Namespaces. ZERO’s modules are classes defined inside a suitable *namespace*, namely a larger scope grouping modules with similar functions or goals. In the sequel, we provide an overview of the software architecture. The namespace `MathOpt` contains the necessary optimization tools for defining and solving mathematical programs – for instance, `MathOpt::IP_Param` for parametrized mixed-integer linear programs, and `MathOpt::LCP` for linear complementarity problems (*LCPs*) – as well as helper functions (e.g., `MathOpt::convexHull` for computing the convex hull of a union of polyhedra). This

class provides a layer between ZERO and the external solvers such as Gurobi and PATH. Arguably, the most relevant namespace is the one of `Games`, which implements the abstraction of specific *RBGs*, such as `Games::IPG` for *IPGs*, and `Games::EPEC` for *NASPs*. The modules inside this namespace orchestrate a tight integration among all the other modules and provide several low-level APIs to the user. The namespace `Algorithms` contains the algorithms to compute the Nash equilibria for *RBGs*. Such algorithms are inside the modules of this namespace and closely coordinate with the modules in `Games`; for instance, the class `Algorithms::IPG::CutAndPlay` associated with the Cut-And-Play algorithm for *IPGs* and *NASPs* [33] coordinates with both `Games::EPEC` and `Games::IPG`. Other than advanced users, ZERO aims to target practitioners that may only be interested in plug-and-play usage of the software. Thus, in the namespace `Models` we provide high-level APIs allowing users to quickly model and solve off-the-shelves instances of *IPGs* and *NASPs*. Furthermore, we propose a standardized format for instances encoded through the data-interchange format `JSON` [123], and integrate complementary helper functions to manage the input and output files. We also include two shell executables working with standardized instance formats allowing users to deploy the algorithms and solve instances on the run. Finally, the namespace `Utils` provides some simple helper functions for writing and reading files, as well as additional numerical and linear algebra utilities. Figure 7.1 provides a schematic representation of the architecture.

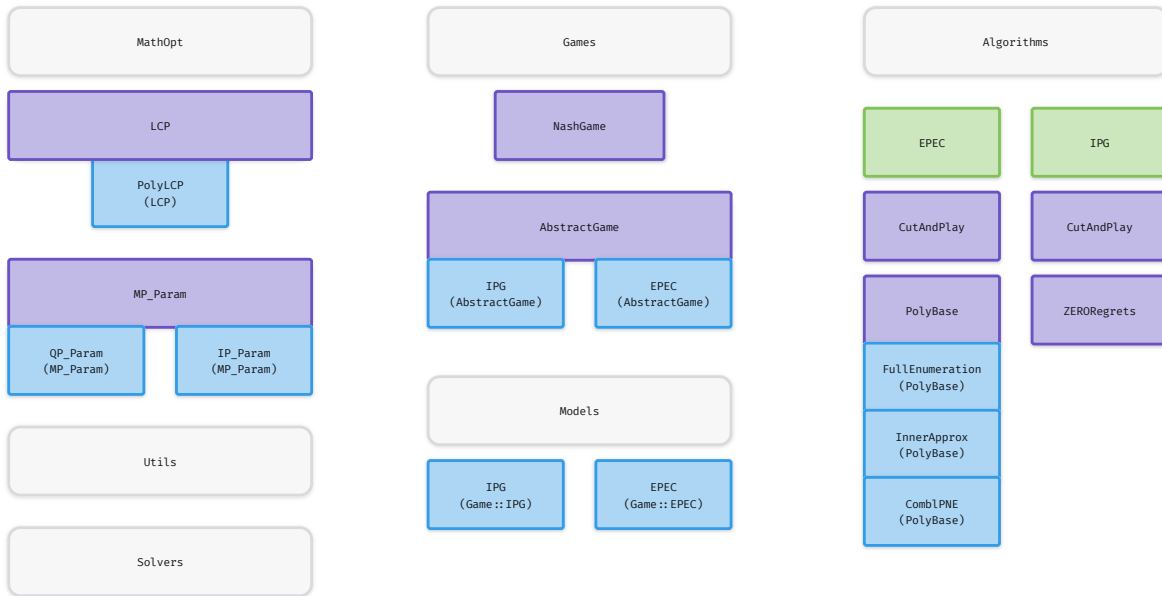


Figure 7.1 A schematic view of ZERO’s modules, 10000 lines of code, 50 files, 40 classes, and 450 functions. The *namespaces* are in gray, and the relative content is grouped below. The primitive classes are in purple, and the associated inheritor classes are in blue. Nested namespaces are in green.

7.4 Modeling the Knapsack Game

We showcase how to model an instance of the Knapsack Game of Example 5 with ZERO. Let *blue* be Player 1 and *red* be Player 2. Each player i seeks to pack $m_i = 2$ items into its knapsack with capacity $b^i = 5$. The optimization problems for blue and red are in (7.3) and (7.4), respectively.

$$\max_{x^1} \quad x_1^1 + 2x_2^1 - 2x_1^1x_1^2 - 3x_2^1x_2^2 \quad (7.3a)$$

$$\text{s.t.} \quad 3x_1^1 + 4x_2^1 \leq 5, x^1 \in \{0, 1\}^2 \quad (7.3b)$$

$$\max_{x^2} \quad 3x_1^2 + 5x_2^2 - 5x_1^2x_1^1 - 4x_2^2x_2^1 \quad (7.4a)$$

$$\text{s.t.} \quad 2x_1^2 + 5x_2^2 \leq 5, x^2 \in \{0, 1\}^2 \quad (7.4b)$$

This problem has 3 Nash equilibria: the Pure-Strategy Nash equilibria $(x_1^1, x_2^1, x_1^2, x_2^2) = (0, 1, 1, 0)$, $(x_1^1, x_2^1, x_1^2, x_2^2) = (1, 0, 0, 1)$, and the Mixed-Strategy Nash equilibrium $(x_1^1, x_2^1, x_1^2, x_2^2) = (\frac{2}{9}, \frac{7}{9}, \frac{2}{5}, \frac{3}{5})$. We attempt to find one of them by using the *Cut-And-Play* algorithm from [33]. Intuitively, this algorithm iteratively refines each players' feasible region starting from its linear relaxations (i.e., the polyhedron given by dropping the integrality constraint in either (7.3) or (7.4)). Specifically, the algorithm iteratively refines the linear relaxations adding cutting planes (some of which generated by Cgl from Coin OR [109]) or by branching until it finds a Nash equilibrium.

Modeling and solving with ZERO. Figure 7.2 demonstrate the use of our high-level API for *IPGs* by modeling the Knapsack Game in (7.3) and (7.4). We start by including the only header file `zero.h` in Step 1 – which contains the specifications for the entire library – and by creating a new Gurobi environment in Step 4. In Step 5 we create a new empty *IPG* instance (`Models::IPG::IPGInstance`), which we will later populate with the programs in (7.3) and (7.4). From Step 7 to Step 12, we create the objects holding the data for the integer programs, for instance, the vector a for the knapsack constraint and the vector `IntegerIndexes` containing the indices of the integer-constrained variables. We fill in the data from (7.3) from Step 14 to Step 21, and create the (parametrized) integer program for player *blue* in Step 24 with a constructor of `MathOpt::IP_Param`. The latter class infers the number of parameters – namely the number other players variables – by counting the number of rows of C^1 ; in this case, the parameters are 2, and they are associated to the choices of Player 2. From Step 26 to Step 29, we iterate this data-filling process for *red*, and eventually add the two parametrized integer programs to the `IPG_Instance` in Steps 32 and 33. In Step 34, we save the instance with the standardized data format for ZERO instances. The solution process starts from line Step 35, where we instantiate – in the object `KnapsackGame`

```

1  #include <zero.h>
2
3  int main(int argc, char **argv) {
4      GRBEnv GurobiEnv;
5      Models::IPG::IPGInstance IPG_Instance; // The IPG Instance
6      int numItems = 2, numPlayers = 2;
7      arma::vec c(numItems); // Profits c in the objective
8      arma::sp_mat C( (numPlayers-1) * numItems, numItems); // C terms in the objective
9      arma::sp_mat a(1, numItems); // LHS for Knapsack constraint
10     arma::vec b(1); // RHS for constraints
11     arma::vec IntegerIndexes(numItems); // The index of the integer variables
12     VariableBounds VarBounds = {{0, 1}, {0, 1}}; // Implicit bounds (LB,UB) on variables.
13
14     //Fill the values in the parameterized integer problem
15     b(0) = 5; // Knapsack Capacity
16     for (unsigned int i = 0; i < numItems; ++i)
17         IntegerIndexes.at(i) = i;
18
19     C(0, 0) = 2; C(1, 1) = 3; // C terms in the objective for player Blue
20     a(0, 0) = 3; a(0, 1) = 4; // Knapsack Constraints
21     c(0) = -1; c(1) = -2; // The standard is minimization, hence minus
22
23     // Create a parametrized Integer Program for player Blue
24     MathOpt::IP_Param PlayerBlue(C, a, b, c, IntegerIndexes, VarBounds, &GurobiEnv);
25
26     // Parametrized Integer Program for player Red.
27     C(0, 0) = 5; C(1, 1) = 4; a(0, 0) = 2; a(0, 1) = 5; c(0) = -3; c(1) = -5;
28
29     MathOpt::IP_Param PlayerRed(C, a, b, c, IntegerIndexes, VarBounds, &GurobiEnv);
30
31     // Add the players to the instance. We can also specify a file path to write the instance
32     IPG_Instance.addIPParam(PlayerBlue, "PlayerBlue_KP");
33     IPG_Instance.addIPParam(PlayerRed, "PlayerRed_KP");
34     IPG_Instance.save("A_Knapsack_Game"); // Save the instance with the standardize format
35
36     Models::IPG::IPG KnapsackGame(&GurobiEnv, IPG_Instance); // Create a model from the
37     // instance
38     // Select the equilibrium to compute a Nash Equilibrium
39     KnapsackGame.setAlgorithm(Data::IPG::Algorithms::CutAndPlay);
40
41     // A few optional settings
42     KnapsackGame.setDeviationTolerance(3e-4); // Numerical tolerance
43     KnapsackGame.setNumThreads(4); // How many threads, if supported by the solver?
44     KnapsackGame.setLCPAlgorithm(Data::LCP::Algorithms::MIP); // How do we solve the LCPs?
45     KnapsackGame.setTimeLimit(5); // Time limit in second
46     KnapsackGame.finalize(); // Lock the model
47     // Run and get the results
48     KnapsackGame.findNashEq();
49     KnapsackGame.getX().at(0).print("Player Blue:"); // Print the solution
50     KnapsackGame.getX().at(1).print("Player Red:");
51 }

```

Figure 7.2 An Example of a C++ instantiation of a 2-player Knapsack Game in ZERO

– an *IPG* model with the data contained in *IPG_Instance*. We employ the constructor of *Models::IPG::IPG* by also specifying a pointer to the Gurobi environment. In Step 37, we instruct *ZERO* to use the *Cut-And-Play* algorithm to solve *KnapsackGame*. In Steps 38 and 43, we set some extra options, and finally start computing the Nash equilibria in Step 45 by calling the method *Models::IPG::IPG::findNashEq()*. We print the Nash equilibrium found by

the *Cut-And-Play* in Steps 46 and 47.

7.5 Conclusions and Future Directions

We introduced ZERO, a multi-purpose C++ library offering the base ingredients to help users model and solve *RBGs*. On the one side, ZERO implements high-level and intuitive APIs to formulate *RBGs* and solve them. On the other side, its modular and extensive design enables advanced users and researchers to build customized algorithms. A current limitation of ZERO is the availability of only two mathematical programming solvers. We plan to extend further the support for other solvers, such as SCIP [77]. Furthermore, we believe future methodological advancements will likely enable us to extend our support to other classes of *MPGs* and *RBGs*. Naturally, this is conditional to the development of the appropriate mathematical tools to do so. Indeed, we release ZERO with the ambition to foster methodological and applied research in this newly developing field at the intersection of Game Theory and Mathematical Programming.

Acknowledgements

This work has been supported by the Canada Excellence Research Chair in Real-time Decision-making (DS4DM).

CHAPTER 8 GENERAL DISCUSSION

The ideas and works we presented in Chapters 4 to 7 constitute an heterogeneous set of contributions to the objectives of this thesis. Throughout the chapters, we often addressed intersecting research questions from different perspectives: we provided results ranging from computational complexity to algorithms to compute equilibria, and from their selection to their application in energy. This section reviews some diagonal themes that matured and emerged through the thesis. We believe the interpretations we provide can serve as useful keys for summarizing our work and further developing it.

In this thesis, we exploited the *MPG* representation to provide theoretical results concerning the properties of equilibria, such as their efficiency, their geometrical structure, and computational complexity results. All such properties are instrumental for the design of efficient algorithms to compute and select equilibria: indeed, from a computational standpoint, we also provided novel algorithms and computational frameworks for computing and selecting equilibria in *MPGs*. In order to derive our theoretical and computational results, we often relied on the structure of the optimization problems associated with each player decision problem, and we employed different algorithmic methods. For instance, while in Chapter 5 we exploited complementarity methods to compute Nash equilibria for *RBGs*, in Chapter 4 we provided a diametrically different approach for *IPGs* based on the iterative solution of a series of integer programs.

In Chapter 4, we introduced ZERO Regrets, a cutting plane algorithm that can select and optimize over the set of *PNEs* in *IPGs*. By devising concepts such as *equilibrium inequality*, *equilibrium separation oracle*, and *equilibrium closure*, archetypical tools of integer programming acquire a game-theoretic role. The approach we proposed stems from the geometrical structure of each player's integer program and characterizes the set of *PNEs* as a polyhedron. The final methodology we provided is general, can handle *PNE* selection, and only assumes each player is solving an integer program with linearizable payoff functions. We also intertwined the aims of *AGT* and *MPGs* by studying the computational aspects of the weighted *NFG*, a paradigmatic problem from *AGT*. We showed the game reformulates through an equivalent *IPG* and that ZERO Regrets provides an efficient way to discriminate among the possibly many *PNEs* in the game. Up to our knowledge, this is the first successful and efficient attempt to select (exact) *PNEs* (if any) in the weighted 3-player version of the *NFG*, where we managed to prove efficient *PNEs* exist (i.e., *PNEs* with a low *PoS*) even in large instances. Although providing bounds on equilibria efficiency is unequivocally necessary,

we believe that computing equilibria approaching those bounds is highly relevant for two main reasons. First, designing efficient equilibria selection algorithms is far from being trivial and often provides new perspectives on the game’s underlying structure. Second, specific applications may benefit from computing efficient and exact equilibria instead of approximated ones. In this sense, we believe an algorithmic perspective on equilibria selection complements the rich theory on equilibria’s existence and efficiency.

In Chapter 4, we also considered the *KPG*, a game expressed as an extension of the binary knapsack problem, a classical combinatorial optimization problem. We analyzed the efficiency of its equilibria and proved that, unfortunately, the *PoS* and *PoA* may be arbitrarily bad. In contrast to the *NFG*, the *KPG* is an extension of a classical combinatorial problem and attracted way less attention than the *NFG*. Although one may re-design the game to fulfill some desired computability and efficiency properties – i.e., a bounded *PoS* or a potential argument for the existence of a *PNE* – we believe this may not always be the case. Indeed, we claim that representing the decision-makers’ complex set of operational requirements through constraints of *MPGs* increases modelization fidelity. Formulating games through the optimization problems of their players is crucial to express a more heterogeneous set of constraints often stemming from an application’s requirement. This is the case for the *NASP*’s energy application in Chapter 6, where regulatory constraints shape the pattern of a complex sequence of hierarchical interactions among energy producers and regulatory agencies. The energy *NASP* does not possess such well-desired computability properties, and in fact, it may not even admit an equilibrium in its general form. Although our model is simple, the inherent complexity of additional regulations and constraints may increase the need for a richer *NASP* model. Finally, even in our simple *NASP*, the equilibria provide valuable insights and new perspectives on the decision-making dynamics in an energy market with environmental incentives and carbon taxes.

The need for computing equilibria in more general games led us to develop *CnP*. Throughout Chapter 5, we provided a methodology leveraging the concept of outer approximation. In optimization, outer approximations of feasible regions – and more specifically relaxations – provide insightful bounds on the optimal solution of the original problem; however, this seemed not to be the case with Nash equilibria and outer approximated games. Following this observation, we devised *CnP* by taking inspiration from the Branch-and-Cut algorithm. *CnP* combines an implicit scheme for enumerating the search space and a cutting plane method. It exploits mathematical and integer programming ingredients – such as separation oracles, disjunctive programming, cutting planes – to efficiently compute equilibria in polyhedrally-representable *RBGs*. Centrally, it allows a better interplay between the existing optimization frameworks and *GT*. Indeed, one of the critical aspects of *CnP* is the interoperability between

the algorithm and the established *MIP* technology.

The order of presentation of our works is willingly anachronistic. Each contribution is the result of a series of interactions with the other works, the first of one being the one in Chapter 6. The characterization of the convex hull of each *NASP*'s leader we provided in Chapter 6 provided the funding element for *CnP* in Chapter 5. Under the assumption of reciprocally-bilinear objective payoffs, we introduced the class of *RBGs* and generalized the result of Theorem 14 to this broader family of games. With *CnP* algorithm, we devised a generic methodology dichotomic to the inner approximation hierarchy we proposed in Chapter 6. The Enhanced Separation Oracle of Chapter 5 mimics, in fact, the inner approximation scheme of Chapter 6. While developing *CnP*, we soon faced a problem of paramount importance. There seemed no easy way to select equilibria. When *CnP* computes a feasible *MNE* for the original game, this equilibrium can be the “best” one (i.e., the *MNE* maximizing a desired property) in the given game's approximation but not necessarily in the original game.

The issue of selection is, in fact, not problematic in what we proposed in Chapter 4. While *ZERO Regrets* does not extend to *RBGs*, it works with discrete variables. The contribution of Chapter 4 started as a study on the theoretical and computational properties of the *KPG* and later extended to the *NFGs* and a larger class of *IPGs*. The connection with equilibria selection became apparent where we generalized a class of inequalities we found for the *NFG* (Proposition 1). In practice, what we proposed in the inequalities of Proposition 1 is a lower bound on the payoff of each player; once we have one of its feasible strategies, we can always compute the bound. The class of inequalities we introduced involves variables from multiple players and motivated the theoretical and practical framework of Chapter 4. Remarkably, these inequalities are sufficient to represent the set of equilibria in *IPGs* with linearizable payoffs, and gave us what we defined as the *perfect equilibrium formulation*. As in a cutting plane method, one does not need the perfect formulation to retrieve a feasible optimal solution or, in our case, a *PNE* maximizing a given property. Instead, to optimize over the *PNEs*, we only need an intermediate polyhedron between the perfect equilibrium formulation and the set of pure strategies for the game. We also presented a solid game-theoretic interpretation of *ZERO Regrets*. The algorithm acts as a central authority by proposing, at each iteration, a collective solution to the players. This solution is an integer-feasible strategy profile that maximizes a given property, but it is not necessarily an equilibrium for the game. The selfish players will then individually ask a trusted rationality blackbox – the separation oracle – whether they should accept the proposed strategy or not. Whenever the rationality blackbox advises a player to refuse a solution, it will also provide an interpretable refusal criterion through a cutting plane. The central authority then computes a new collective solution considering the players' additional suggestions.

Finally, as we stressed the importance of computing equilibria to understand their role, we also advocated for lower barriers to entry in *MPGs* with *ZERO*. As the *MPG* methodology should advance, so should the availability of collaborative and open-source software. In Chapter 7, we aimed to give the community a package to foster experimentation for practitioners, researchers, and industrial players.

CHAPTER 9 CONCLUSIONS AND RECOMMENDATIONS

This thesis proposes new perspectives to better grasp the dynamics of multi-agent strategic decision-making in competitive settings where agents are solving optimization problems. The research we propose lies at the interface of *AGT* and Mathematical Optimization and analyzes the strategic interaction among decision-makers through the lenses of a unified framework integrating elements of the two disciplines. We proposed the taxonomy of *MPGs* and explored algorithmic and theoretical matters concerning Nash equilibria existence, computability, and selection. In the following, we briefly summarize the work and provide some considerations concerning the boundaries of our work and future research directions.

9.1 Summary of Works

In Chapter 4, we designed an algorithmic and theoretical framework to compute and select *PNEs* for a class of *IPGs*. By devising concepts such as *equilibrium inequality*, *equilibrium separation oracle*, and *equilibrium closure*, archetypical tools of integer programming and optimization acquire a game-theoretic role. We introduced ZERO Regrets and proved its practical effectiveness by testing on instances of the *NFG* and the *KPG*. We further characterized the complexity of deciding whether the *KPG* admits a *PNE* and showed that the prices on its equilibria may be arbitrarily bad. Finally, we showcased an extensive set of computational results highlighting the existence of efficient equilibria in both games and demonstrated the viability and performance of our algorithm.

In Chapter 5, we introduced *CnP*, a general-purpose algorithm to compute *MNEs* for *RBGs*. Our methodology employs the concept of outer approximation and leverages the rich theory of integer programming. The algorithm we proposed computes *MNEs* by solving a series of “easier” outer-approximated games and combines an implicit enumeration scheme with a cutting plane method. *CnP* is general in its design and integrates the existing theory and tools from optimization into a unified framework for computing equilibria. We provided extensive computational results on both *NASPs* and *KPGs*.

In Chapter 6, we considered a series of hierarchical interactions among the players of Stackelberg games and introduced the novel class of *NASPs*. The games feature sequential and simultaneous multi-level interactions and are powerful modeling tools for energy markets and cybersecurity insurance. We characterized the computational complexity associated with determining the existence of an equilibrium in the game, and we provided two algorithms

to compute equilibria. From a practical perspective, we presented a case study on the Chilean-Argentinean energy market and derived a set of managerial insights stemming from the associated Nash equilibria.

Finally, in Chapter 7 we introduced ZERO, a software package providing all the necessary ingredients to experiment with *RBGs*. We released an open-source package to lower the barrier to entry in the field, hoping to boost methodological developments and practical applications of *RBGs*.

9.2 Extensions and Future Research

MPGs are powerful modeling tools and can help embed game dynamics inside consolidated optimization models. We think the results reported in this thesis constitute a first effort towards developing a unified framework for the interplay of *AGT* and Mathematical Optimization; indeed, the ultimate goal of this thesis is to promote a better integration of the two disciplines. We are cautiously optimistic about the future of *MPGs*, and we believe significant improvements and future developments lie ahead.

Methodologically, we believe there is room – if not the need – for developing more sophisticated algorithms and theoretical frameworks for *MPGs*. The problem of computing equilibria and selecting them are far from being fully understood, and novel algorithms for these tasks may even provide new perspectives on general equilibria selection. Further, integrating mathematical programming techniques may also propel new methodological and practical advancements. Practically, we believe many industrial and applied tasks may benefit from the frameworks *MPGs* provide. Industrial applications unequivocally promoted developments of integer programming, especially since several applications required integer variables to model indivisible quantities. With the same spirit, we believe that an enhanced effort on attacking applications through *MPGs* can hopefully lead to similar results. Further, the informative content of Nash equilibria can provide socially beneficial and explainable prescriptive policies for decision-makers; in the same fashion of our analysis in Chapter 6, we hope the approach can provide similar insights in other application domains. This type of analysis may confirm or invalidate the role of Nash equilibria as solutions in such contexts and shed new light on alternative solution concepts.

Moreover, this thesis considers the Nash equilibrium as the leading solution concept. We assume complete information and common knowledge of rationality for any player and call a player rational when it has no incentive to accept a lower payoff if a better option exists. However, different equilibria or solution concepts may also suit the scope of *MPGs*. The

ZERO Regrets algorithm of Chapter 4 can, for instance, select solutions with a bounded regret by appropriately modifying each player’s separation oracle. By allowing a deviation bounded by a given constant k , the algorithm selects strategy profiles where each player’s regret is at most k . Naturally, when the regrets are 0 for any player, the profile is also a Nash equilibrium. An interesting question is whether we can further relax the structure of ZERO Regrets, and eventually of the *CnP* algorithm, to compute other types of equilibria.

Equilibria selection will probably be critical for ensuring *MPGs*’ equilibria are competitive with single-minded or centrally-advised solutions, for instance, solutions from single optimization problems. In Chapter 4, we provided an efficient way to select *PNEs* in *IPGs*; however, we did not obtain a similar result for *RBGs* and *MNEs* in Chapter 5, despite many efforts. While we have a selection result in Chapter 6, the algorithm we provided requires a computationally-intense full enumeration of the players’ feasible sets, and hence the approach may be practically hard to extend to other types of problems. Nevertheless, the existence of a methodology for *IPGs* may constitute an encouraging step towards developing more general methods in *RBGs*.

Crucially, we believe there is room for developing new classes of equilibrium inequalities for *MPGs*. Although in Chapter 4 we restricted to *IPGs*, the equilibrium inequalities we presented may generalize to other *MPGs*. Perhaps, there exist stronger inequalities eliminating some action from each player’s set of moves. In particular, inequalities restricting each player’s feasible set to the set of “rationalizable” strategies, i.e., strategies that a player may rationally play given some belief on its opponents. Understanding the geometry of the sets of rationalizable moves in *MPGs* can provide a novel perspective on equilibria selection other than practical improvements on *MPGs* computations.

Finally, and perhaps more importantly, we shall consider an ethical perspective. We believe that embedding fairness paradigms into the decision-making process can be practically impactful. In Chapter 1, we described the decision-makers as selfish and self-interested. In an ideal world, decision-makers are benevolent and altruistic, and the outcomes are always beneficial for the society. Unfortunately (for all of us), this is not often the case with individuals, companies, governments. Understanding the dynamics of strategic interaction through *MPGs* should provide better insights for advocating for a fairer use of optimization. Companies, governments, and in general, organizations are likely to solve optimization problems to address their day-to-day operational matters. *MPGs* embed complex game dynamics and have the potential to provide solutions that are individually optimal, socially beneficial, and that hopefully highlight some of the ethical issues related to the undisciplined and selfish use of optimization.

REFERENCES

- [1] T. Achterberg, “Constraint Integer Programming,” Ph.D. dissertation, TU Berlin, TU Berlin, 2007. [Online]. Available: <https://depositonce.tu-berlin.de/handle/11303/1931>
- [2] D. Amdur, B. G. Rabe, and C. P. Borick, “Public views on a carbon tax depend on the proposed use of revenue,” *Issues in Energy and Environmental Policy*, vol. -, no. 13, 2014.
- [3] P. Amigo, S. Cea-Echenique, and F. Feijoo, “A two stage cap-and-trade model with allowance re-trading and capacity investment: The case of the Chilean NDC targets,” *Energy*, vol. 224, p. 120129, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360544221003789>
- [4] E. Anshelevich, A. Dasgupta, E. Tardos, and T. Wexler, “Near-optimal network design with selfish agents,” in *Proceedings of the thirty-fifth ACM symposium on Theory of computing - STOC '03*. San Diego, CA, USA: ACM Press, 2003, p. 511. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=780542.780617>
- [5] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden, “The Price of Stability for Network Design with Fair Cost Allocation,” *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1602–1623, Jan. 2008. [Online]. Available: <http://epubs.siam.org/doi/10.1137/070680096>
- [6] C. Audet, S. Belhaiza, and P. Hansen, “Enumeration of All the Extreme Equilibria in Game Theory: Bimatrix and Polymatrix Games,” *Journal of Optimization Theory and Applications*, vol. 129, no. 3, pp. 349–372, Dec. 2006. [Online]. Available: <http://link.springer.com/10.1007/s10957-006-9070-3>
- [7] D. Avis, G. D. Rosenberg, R. Savani, and B. von Stengel, “Enumeration of Nash equilibria for two-player games,” *Economic Theory*, vol. 42, no. 1, pp. 9–37, Jan. 2010. [Online]. Available: <http://link.springer.com/10.1007/s00199-009-0449-x>
- [8] E. Balas, “Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems,” *SIAM Journal on Algebraic Discrete Methods*, vol. 6, no. 3, pp. 466–486, Jul. 1985. [Online]. Available: <https://epubs.siam.org/doi/10.1137/0606047>

- [9] —, “Disjunctive programming: Properties of the convex hull of feasible points,” *Discrete Applied Mathematics*, vol. 89, no. 1-3, pp. 3–44, Dec. 1998. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0166218X9800136X>
- [10] J. F. Bard, “Some properties of the bilevel programming problem,” *Journal of Optimization Theory and Applications*, vol. 68, no. 2, pp. 371–378, Feb. 1991.
- [11] J. F. Bard, J. Plummer, and J. C. Sourie, “Determining Tax Credits for Converting Nonfood Crops to Biofuels: An Application of Bilevel Programming,” in *Multilevel Optimization: Algorithms and Applications*, A. Migdalas, P. M. Pardalos, and P. Värbrand, Eds. Boston, MA: Springer US, 1998, pp. 23–50.
- [12] —, “A bilevel programming approach to determining tax credits for biofuel production,” *European Journal of Operational Research*, vol. 120, no. 1, pp. 30–46, 2000.
- [13] A. Basu, C. T. Ryan, and S. Sankaranarayanan, “Mixed-integer bilevel representability,” *Mathematical Programming*, vol. 185, no. 1, pp. 163–197, Aug. 2019. [Online]. Available: <https://doi.org/10.1007/s10107-019-01424-w>
- [14] B. D. Bernheim, “Rationalizable Strategic Behavior,” *Econometrica*, vol. 52, no. 4, p. 1007, Jul. 1984. [Online]. Available: <https://www.jstor.org/stable/1911196?origin=crossref>
- [15] T. Berthold, “Heuristic algorithms in global MINLP solvers,” PhD Thesis, TU Berlin, 2014. [Online]. Available: <http://www.zib.de/berthold/Berthold2014.pdf>
- [16] R. E. Bixby, “Solving Real-World Linear Programs: A Decade and More of Progress,” *Operations Research*, vol. 50, no. 1, pp. 3–15, Feb. 2002. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.50.1.3.17780>
- [17] D. Boffey, “Italy blocks export of 250,000 AstraZeneca vaccine doses to Australia,” *The Guardian*, Jan 2021, accessed on March the 4th 2021. [Online]. Available: <https://www.theguardian.com/world/2021/mar/04/italy-blocks-export-of-250000-astrazeneca-vaccine-doses-to-australia>
- [18] —, “EU threatens to block Covid vaccine exports amid AstraZeneca shortfall,” *The Financial Times*, Jan 2021, accessed on March the 4th 2021. [Online]. Available: <https://www.theguardian.com/world/2021/jan/25/eu-threatens-to-block-covid-vaccine-exports-amid-astrazeneca-shortfall>

- [19] E. A. Boyd, “Fenchel Cutting Planes for Integer Programs,” *Operations Research*, vol. 42, no. 1, pp. 53–64, Feb. 1994. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.42.1.53>
- [20] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge, UK ; New York: Cambridge University Press, 2004.
- [21] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard, “A Bilevel Model for Toll Optimization on a Multicommodity Transportation Network,” *Transportation Science*, vol. 35, no. 4, pp. 345–358, 2001. [Online]. Available: <https://doi.org/10.1287/trsc.35.4.345.10433>
- [22] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard, “Joint Design and Pricing on a Network,” *Operations Research*, vol. 56, no. 5, pp. 1104–1115, 2008.
- [23] W. Candler and R. Norto, “Multi-level programming and development policy,” *Working Paper*, pp. 1–56, 1977.
- [24] W. Candler and R. Townsley, “A linear two-level programming problem,” *Computers & Operations Research*, vol. 9, no. 1, pp. 59–76, 1982.
- [25] A. Caprara, M. Carvalho, A. Lodi, and G. J. Woeginger, “A Study on the Computational Complexity of the Bilevel Knapsack Problem,” *SIAM Journal on Optimization*, vol. 24, no. 2, pp. 823–838, Jan. 2014. [Online]. Available: <http://epubs.siam.org/doi/10.1137/130906593>
- [26] —, “Bilevel Knapsack with interdiction constraints,” *INFORMS Journal on Computing*, vol. 28, no. 2, pp. 319–333, 2016.
- [27] M. Carvalho, “Computation of equilibria on integer programming games,” Ph.D. dissertation, Universidade do Porto, Faculdade de Ciências, 2016.
- [28] M. Carvalho, A. Lodi, J. P. Pedroso, and A. Viana, “Nash equilibria in the two-player kidney exchange game,” *Mathematical Programming*, vol. 161, no. 1, pp. 289–417, Jan 2017.
- [29] M. Carvalho, A. Lodi, and J. P. Pedroso, “Existence of Nash Equilibria on Integer Programming Games,” in *Operational Research*, A. I. F. Vaz, J. P. Almeida, J. F. Oliveira, and A. A. Pinto, Eds., vol. 223. Cham: Springer International Publishing, 2018, pp. 11–23, series Title: Springer Proceedings in Mathematics & Statistics. [Online]. Available: http://link.springer.com/10.1007/978-3-319-71583-4_2

- [30] M. Carvalho, J. P. Pedroso, C. Telha, and M. V. Vyve, “Competitive uncapacitated lot-sizing game,” *International Journal of Production Economics*, vol. 204, pp. 148–159, 2018.
- [31] M. Carvalho, G. Dragotto, F. Feijoo, A. Lodi, and S. Sankaranarayanan, “When Nash Meets Stackelberg,” *arXiv preprint*, p. 29, 2019. [Online]. Available: <https://arxiv.org/abs/1910.06452>
- [32] M. Carvalho, A. Lodi, and J. P. Pedroso, “Computing Nash Equilibria for Integer Programming Games,” *arXiv*, vol. abs/2012.07082, 2020. [Online]. Available: <https://arxiv.org/abs/2012.07082>
- [33] M. Carvalho, G. Dragotto, A. Lodi, and S. Sankaranarayanan, “The Cut and Play Algorithm: Computing Nash Equilibria via Outer Approximations,” *arXiv*, vol. abs/2111.05726, 2021. [Online]. Available: <https://arxiv.org/abs/2111.05726>
- [34] B. Cashell, W. D. Jackson, M. Jickling, and B. Webel, “The economic impact of cyber-attacks,” *Congressional Research Service Documents, CRS RL32331 (Washington DC)*, 2004.
- [35] H.-L. Chen and T. Roughgarden, “Network design with weighted players,” in *Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures - SPAA '06*. Cambridge, Massachusetts, USA: ACM Press, 2006, p. 29. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1148109.1148114>
- [36] X. Chen and X. Deng, “Settling the complexity of two-player Nash equilibrium,” in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006, pp. 261–272.
- [37] V. Chvátal, “Edmonds polytopes and a hierarchy of combinatorial problems,” *Discrete Mathematics*, vol. 4, no. 4, pp. 305–337, Apr. 1973. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0012365X73901672>
- [38] V. Chvátal, W. Cook, and D. Espinoza, “Local cuts for mixed-integer programming,” *Mathematical Programming Computation*, vol. 5, no. 2, pp. 171–200, Jun. 2013. [Online]. Available: <http://link.springer.com/10.1007/s12532-013-0052-9>
- [39] Coin-OR, “The COIN-OR Cut Generation Library,” <https://github.com/coin-or/Cgl>, 2021, accessed: 2021-05-20.

- [40] B. Colson, P. Marcotte, and G. Savard, “An overview of bilevel optimization,” *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, Jun. 2007. [Online]. Available: <http://link.springer.com/10.1007/s10479-007-0176-2>
- [41] M. Conforti, G. Cornuéjols, and G. Zambelli, “Extended formulations in combinatorial optimization,” *4OR*, vol. 8, no. 1, pp. 1–48, Mar. 2010. [Online]. Available: <http://link.springer.com/10.1007/s10288-010-0122-z>
- [42] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming*, ser. Graduate Texts in Mathematics. Cham: Springer International Publishing, 2014, vol. 271. [Online]. Available: <http://link.springer.com/10.1007/978-3-319-11008-0>
- [43] V. Conitzer and T. Sandholm, “Complexity Results about Nash Equilibria,” *arXiv:cs/0205074*, May 2002, arXiv: cs/0205074. [Online]. Available: <http://arxiv.org/abs/cs/0205074>
- [44] —, “New complexity results about Nash equilibria,” *Games and Economic Behavior*, vol. 63, no. 2, pp. 621–641, Jul. 2008. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0899825608000936>
- [45] J. R. Correa, A. S. Schulz, and N. E. Stier-Moses, “Selfish Routing in Capacitated Networks,” *Mathematics of Operations Research*, vol. 29, no. 4, pp. 961–976, Nov. 2004. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/moor.1040.0098>
- [46] R. Cottle, J.-S. Pang, and R. E. Stone, *The linear complementarity problem*, ser. Classics in applied mathematics. Philadelphia: Society for Industrial and Applied Mathematics, 2009, no. 60, oCLC: ocn370388155.
- [47] T. Cronert and S. Minner, “Equilibrium identification and selection in integer programming games,” *SSRN Preprint 3762380*, p. 38, 2021.
- [48] G. B. Dantzig, “Reminiscences about the origins of linear programming,” in *Mathematical Programming The State of the Art*. Springer, 1983, pp. 78–86.
- [49] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The complexity of computing a Nash equilibrium,” *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, Feb. 2009. [Online]. Available: <https://dl.acm.org/doi/10.1145/1461928.1461951>
- [50] G. Debreu, “A Social Equilibrium Existence Theorem,” *Proceedings of the National Academy of Sciences*, vol. 38, no. 10, pp. 886–893, Oct. 1952. [Online]. Available: <http://www.pnas.org/lookup/doi/10.1073/pnas.38.10.886>

- [51] A. Del Pia, M. Ferris, and C. Michini, “Totally Unimodular Congestion Games,” in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '17. USA: Society for Industrial and Applied Mathematics, 2017, p. 577–588.
- [52] V. DeMiguel and H. Xu, “A stochastic multiple-leader Stackelberg model: analysis, computation, and application,” *Operations Research*, vol. 57, no. 5, pp. 1220–1235, 2009.
- [53] S. Dempe and A. Zemkoho, Eds., *Bilevel Optimization: Advances and Next Challenges*, ser. Springer Optimization and Its Applications. Cham: Springer International Publishing, 2020, vol. 161. [Online]. Available: <http://link.springer.com/10.1007/978-3-030-52119-6>
- [54] S. T. DeNegre and T. K. Ralphs, “A Branch-and-cut Algorithm for Integer Bilevel Linear Programs,” in *Operations Research and Cyber-Infrastructure*, J. W. Chinneck, B. Kristjansson, and M. J. Saltzman, Eds. Boston, MA: Springer US, 2009, pp. 65–78. [Online]. Available: http://link.springer.com/10.1007/978-0-387-88843-9_4
- [55] S. S. Dey, M. Molinaro, and Q. Wang, “Approximating polyhedra with sparse inequalities,” *Mathematical Programming*, vol. 154, no. 1-2, pp. 329–352, Dec. 2015. [Online]. Available: <http://link.springer.com/10.1007/s10107-015-0925-y>
- [56] S. P. Dirkse and M. C. Ferris, “The PATH solver: a nonmonotone stabilization scheme for mixed complementarity problems,” *Optimization Methods and Software*, vol. 5, no. 2, pp. 123–156, Jan. 1995. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/10556789508805606>
- [57] G. Dragotto and R. Scatamacchia, “ZERO Regrets Algorithm: Optimizing over Pure Nash Equilibria via Integer Programming,” *arXiv*, vol. abs/2111.06382, 2021. [Online]. Available: <https://arxiv.org/abs/2111.06382>
- [58] G. Dragotto, S. Sankaranarayanan, M. Carvalho, and A. Lodi, “ZERO: Playing Mathematical Programming Games,” *arXiv*, vol. abs/2111.07932, 2021. [Online]. Available: <http://getzero.one>
- [59] B. C. Eaves, “The Linear Complementarity Problem,” *Management Science*, vol. 17, no. 9, pp. 612–634, May 1971. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/mnsc.17.9.612>
- [60] C. E. Eggermont and G. J. Woeginger, “Motion planning with pulley, rope, and baskets,” *Theory of Computing Systems*, vol. 53, no. 4, pp. 569–582, 2013.

- [61] R. Egging, S. A. Gabriel, F. Holz, and J. Zhuang, “A complementarity model for the European natural gas market,” *Energy policy*, vol. 36, no. 7, pp. 2385–2414, 2008.
- [62] R. Egging, F. Holz, and S. A. Gabriel, “The World Gas Model: A multi-period mixed complementarity model for the global natural gas market,” *Energy*, vol. 35, no. 10, pp. 4016–4029, 2010.
- [63] F. Facchinei and J.-S. Pang, *Finite-dimensional variational inequalities and complementarity problems*, ser. Springer series in operations research. New York: Springer, 2003.
- [64] F. Facchinei, A. Fischer, and V. Piccialli, “On generalized Nash games and variational inequalities,” *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, Mar. 2007. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0167637706000484>
- [65] K. Fan, “Fixed-point and Minimax Theorems in Locally Convex Topological Linear Spaces,” *Proceedings of the National Academy of Sciences*, vol. 38, no. 2, pp. 121–126, Feb. 1952. [Online]. Available: <http://www.pnas.org/cgi/doi/10.1073/pnas.38.2.121>
- [66] F. Feijoo and T. K. Das, “Design of Pareto optimal CO₂ cap-and-trade policies for deregulated electricity networks,” *Applied energy*, vol. 119, pp. 371–383, 2014.
- [67] —, “Emissions control via carbon policies and microgrid generation: A bilevel model and Pareto analysis,” *Energy*, vol. 90, pp. 1545–1555, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S036054421500866X>
- [68] F. Feijoo, D. Huppmann, L. Sakiyama, and S. Siddiqui, “North American natural gas model: Impact of cross-border trade with Mexico,” *Energy*, vol. 112, pp. 1084–1095, 2016.
- [69] F. Feijoo, G. C. Iyer, C. Avraam, S. A. Siddiqui, L. E. Clarke, S. Sankaranarayanan, M. T. Binsted, P. L. Patel, N. C. Prates, E. Torres-Alfaro *et al.*, “The future of natural gas infrastructure development in the United states,” *Applied energy*, vol. 228, pp. 149–166, 2018.
- [70] M. C. Ferris and T. S. Munson, “Interfaces to PATH 3.0: Design, Implementation and Usage,” *Computational Optimization and Applications*, vol. 12, no. 1/3, pp. 207–227, 1999. [Online]. Available: <http://link.springer.com/10.1023/A:1008636318275>
- [71] M. Fischetti, A. Lodi, and A. Tramontani, “On the separation of disjunctive cuts,” *Mathematical Programming*, vol. 128, no. 1-2, pp. 205–230, Jun. 2011. [Online]. Available: <http://link.springer.com/10.1007/s10107-009-0300-y>

- [72] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl, “A New General-Purpose Algorithm for Mixed-Integer Bilevel Linear Programs,” *Operations Research*, vol. 65, no. 6, pp. 1615–1637, Dec. 2017. [Online]. Available: <http://pubsonline.informs.org/doi/10.1287/opre.2017.1650>
- [73] S. Fleming, J. Brunsden, and Miles, “Italy blocks shipment of Oxford/AstraZeneca vaccine to Australia,” *The Financial Times*, Mar 2021, accessed on March the 4th 2021. [Online]. Available: <https://www.ft.com/content/bed655ac-9285-486a-b5ad-b015284798c8>
- [74] S. A. Gabriel and F. U. Leuthold, “Solving discretely-constrained MPEC problems with applications in electric power markets,” *Energy Economics*, vol. 32, no. 1, pp. 3–14, 2010.
- [75] S. A. Gabriel, A. J. Conejo, J. D. Fuller, B. F. Hobbs, and C. Ruiz, *Complementarity Modeling in Energy Markets*. Springer-Verlag, 2012.
- [76] I. Gilboa and E. Zemel, “Nash and correlated equilibria: Some complexity considerations,” *Games and Economic Behavior*, vol. 1, no. 1, pp. 80–93, 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0899825689900067>
- [77] A. Gleixner, M. Bastubbe, L. Eifler, T. Gally, G. Gamrath, R. L. Gottwald, G. Hendel, C. Hojny, T. Koch, M. E. Lübbecke, S. J. Maher, M. Miltenberger, B. Müller, M. E. Pfetsch, C. Puchert, D. Rehfeldt, F. Schlösser, C. Schubert, F. Serrano, Y. Shinano, J. M. Viernickel, M. Walter, F. Wegscheider, J. T. Witt, and J. Witzig, “The SCIP Optimization Suite 6.0,” Optimization Online, Technical Report, July 2018.
- [78] I. L. Glicksberg, “A Further Generalization of the Kakutani Fixed Point Theorem, with Application to Nash Equilibrium Points,” *Proceedings of the American Mathematical Society*, vol. 3, no. 1, p. 170, Feb. 1952. [Online]. Available: <https://www.jstor.org/stable/2032478?origin=crossref>
- [79] R. E. Gomory, “Outline of an algorithm for integer solutions to linear programs,” *Bulletin of the American Mathematical Society*, vol. 64, no. 5, pp. 275–279, Sep. 1958. [Online]. Available: <http://www.ams.org/journal-getitem?pii=S0002-9904-1958-10224-4>
- [80] L. A. Gordon and M. P. Loeb, “The economics of information security investment,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 5, no. 4, pp. 438–457, 2002.

- [81] M. Grötschel, L. Lovász, and A. Schrijver, “The ellipsoid method and its consequences in combinatorial optimization,” *Combinatorica*, vol. 1, no. 2, pp. 169–197, Jun. 1981. [Online]. Available: <http://link.springer.com/10.1007/BF02579273>
- [82] C. Guo, M. Bodur, and J. A. Taylor, “Copositive Duality for Discrete Markets and Games,” *arXiv:2101.05379 [econ, math]*, Jan. 2021, arXiv: 2101.05379. [Online]. Available: <http://arxiv.org/abs/2101.05379>
- [83] Gurobi Optimization, LLC, “Gurobi Optimizer Reference Manual,” 2021. [Online]. Available: <https://www.gurobi.com>
- [84] J. C. Harsanyi and R. Selten, *A general theory of equilibrium selection in games*, 2nd ed., ser. MIT Press Classics. Cambridge, Mass.: MIT Pr, 1992.
- [85] R. J. Hillestad and S. E. Jacobsen, “Linear programs with an additional reverse convex constraint,” *Applied Mathematics & Optimization*, vol. 6, no. 1, pp. 257–269, Mar. 1980. [Online]. Available: <http://link.springer.com/10.1007/BF01442898>
- [86] —, “Reverse convex programming,” *Applied Mathematics & Optimization*, vol. 6, no. 1, pp. 63–78, Mar. 1980. [Online]. Available: <http://link.springer.com/10.1007/BF01442883>
- [87] B. F. Hobbs, C. B. Metzler, and J.-S. Pang, “Strategic gaming analysis for electric power systems: An MPEC approach,” *IEEE transactions on power systems*, vol. 15, no. 2, pp. 638–645, 2000.
- [88] F. Holz, C. Von Hirschhausen, and C. Kemfert, “A strategic model of European gas supply (GASMOD),” *Energy Economics*, vol. 30, no. 3, pp. 766–788, 2008.
- [89] X. Hu and D. Ralph, “Using EPECs to Model Bilevel Games in Restructured Electricity Markets with Locational Prices,” *Operations Research*, vol. 55, no. 5, pp. 809–827, Oct. 2007. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.1070.0431>
- [90] R. Jeroslow, “The polynomial hierarchy and a simple model for competitive analysis,” *Mathematical Programming*, vol. 32, no. 2, pp. 146–164, 1985.
- [91] R. Karp and C. Papadimitriou, “On Linear Characterizations of Combinatorial Optimization Problems,” *SIAM Journal on Computing*, vol. 11, no. 4, pp. 620–632, 1982.
- [92] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-24777-7>

- [93] T. Kleinert and M. Schmitt, “Why there is no need to use a big-M in linear bilevel optimization: A computational study of two ready-to-use approaches,” *Optimization Online Preprint*, 2020. [Online]. Available: <http://www.optimization-online.org/DB%5FFILE/2020/10/8065.pdf>
- [94] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt, “A Survey on Mixed-Integer Programming Techniques in Bilevel Optimization,” *EURO Journal on Computational Optimization*, vol. 9, p. 100007, 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2192440621001349>
- [95] M. Köppe, C. T. Ryan, and M. Queyranne, “Rational Generating Functions and Integer Programming Games,” *Operations Research*, vol. 59, no. 6, pp. 1445–1460, Dec. 2011. [Online]. Available: <http://pubsonline.informs.org/doi/abs/10.1287/opre.1110.0964>
- [96] E. Koutsoupias and C. Papadimitriou, “Worst-Case Equilibria,” in *STACS 99*, G. Goos, J. Hartmanis, J. van Leeuwen, C. Meinel, and S. Tison, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, vol. 1563, pp. 404–413, series Title: Lecture Notes in Computer Science.
- [97] A. A. Kulkarni and U. V. Shanbhag, “A shared-constraint approach to multi-leader multi-follower games,” *Set-valued and variational analysis*, vol. 22, no. 4, pp. 691–720, 2014.
- [98] —, “An existence result for hierarchical Stackelberg v/s Stackelberg games,” *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3379–3384, 2015.
- [99] M. Labbé and A. Violin, “Bilevel programming and price setting problems,” *4OR*, vol. 11, no. 1, pp. 1–30, Mar 2013.
- [100] M. Labbé, P. Marcotte, and G. Savard, “A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing,” *Management Science*, vol. 44, no. 12-part-1, pp. 1608–1622, 1998. [Online]. Available: <https://pubsonline.informs.org/doi/abs/10.1287/mnsc.44.12.1608>
- [101] A. H. Land and A. G. Doig, “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, vol. 28, no. 3, p. 497, Jul. 1960. [Online]. Available: <https://www.jstor.org/stable/1910129?origin=crossref>
- [102] C. E. Lemke and J. T. Howson, Jr., “Equilibrium Points of Bimatrix Games,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 413–423, Jun. 1964. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0112033>

- [103] S. Leyffer and T. Munson, “Solving multi-leader–common-follower games,” *Optimisation Methods & Software*, vol. 25, no. 4, pp. 601–623, 2010.
- [104] H. Li and J. Meissner, “Competition under capacitated dynamic lot-sizing with capacity acquisition,” *International Journal of Production Economics*, vol. 131, no. 2, pp. 535–544, 2011.
- [105] Y. Liu and Y. Lu, “The economic impact of different carbon tax revenue recycling schemes in China: A model-based scenario analysis,” *Applied Energy*, vol. 141, pp. 96–105, 2015.
- [106] A. Lodi, “Mixed Integer Programming Computation,” in *50 Years of Integer Programming 1958-2008*, M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 619–645. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-68279-0%5F16>
- [107] —, “The Heuristic (Dark) Side of MIP Solvers,” in *Hybrid Metaheuristics*, E.-G. Talbi, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, vol. 434, pp. 273–284. [Online]. Available: <http://link.springer.com/10.1007/978-3-642-30671-6%5F10>
- [108] A. Lodi, T. K. Ralphs, and G. J. Woeginger, “Bilevel programming and the separation problem,” *Mathematical Programming*, 2014.
- [109] R. Lougee-Heimer, “The common optimization interface for operations research: Promoting open-source software in the operations research community,” *IBM Journal of Research and Development*, vol. 47, no. 1, pp. 57–66, 2003.
- [110] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints*. Cambridge ; New York: Cambridge University Press, 1996.
- [111] E. Martelli, M. Freschini, and M. Zatti, “Optimization of renewable energy subsidy and carbon tax for multi energy systems using bilevel programming,” *Applied Energy*, vol. 267, p. 115089, 2020.
- [112] R. McKelvey and A. McLennan, *Computation of Equilibria in Finite Games. In the Handbook of Computation Economics, Vol. I*. Elsevier, Eds. Amman, H., Kendrick, DA, Rust, J, 1996.
- [113] R. D. McKelvey, A. M. McLennan, and T. L. Turocy, “Gambit: Software Tools for Game Theory,” Version 0.2006.01.20, Tech. Rep., 2006. [Online]. Available: <http://econweb.tamu.edu/gambit/>

- [114] R. R. Meyer, “On the existence of optimal solutions to integer and mixed-integer programming problems,” *Mathematical Programming*, vol. 7, no. 1, pp. 223–235, Dec. 1974. [Online]. Available: <http://link.springer.com/10.1007/BF01585518>
- [115] O. Morgenstern and J. Von Neumann, *Theory of games and economic behavior*. Princeton University Press, 1953.
- [116] J. F. Nash, “Equilibrium Points in N-Person Games.” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, no. 1, pp. 48–9, Jan. 1950. [Online]. Available: <http://www.jstor.org/stable/88031>
- [117] J. Nash, “Non-cooperative games,” *Annals of mathematics*, vol. 54, no. 2, pp. 286–295, Sep. 1951. [Online]. Available: <https://www.jstor.org/stable/1969529>
- [118] N. Nisan, Ed., *Algorithmic Game Theory*. Cambridge: Cambridge Univ. Press, 2008, oCLC: 315869343.
- [119] D. J. Olsen, Y. Dvorkin, R. Fernández-Blanco, and M. A. Ortega-Vazquez, “Optimal carbon taxes for emissions targets in the electricity sector,” *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 5892–5901, 2018.
- [120] M. Padberg and G. Rinaldi, “A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems,” *SIAM Review*, vol. 33, no. 1, pp. 60–100, 1991. [Online]. Available: <http://www.jstor.org/stable/2030652>
- [121] D. G. Pearce, “Rationalizable Strategic Behavior and the Problem of Perfection,” *Econometrica*, vol. 52, no. 4, p. 1029, Jul. 1984. [Online]. Available: <https://www.jstor.org/stable/1911197?origin=crossref>
- [122] M. Perregaard and E. Balas, “Generating Cuts from Multiple-Term Disjunctions,” in *Integer Programming and Combinatorial Optimization*, G. Goos, J. Hartmanis, J. van Leeuwen, K. Aardal, and B. Gerards, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, vol. 2081, pp. 348–360, series Title: Lecture Notes in Computer Science. [Online]. Available: <http://link.springer.com/10.1007/3-540-45535-3%5F27>
- [123] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, “Foundations of JSON schema,” in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 263–273.
- [124] R. Porter, E. Nudelman, and Y. Shoham, “Simple search methods for finding a Nash equilibrium,” *Games and Economic Behavior*, vol. 63, no. 2, pp. 642–662, Jul. 2008. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0899825606000935>

- [125] D. Ralph and Y. Smeers, “EPECs as models for electricity markets,” in *2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 74—80. [Online]. Available: <http://www3.eng.cam.ac.uk/~dr241/Papers/Ralph-Smeers-EPEC-electricity.pdf>
- [126] G. D. Rosenberg, “Enumeration of All Extreme Equilibria of Bimatrix Games with Integer Pivoting and Improved Degeneracy Check,” *CDAM Research Report LSE-CDAM-2004-18*, p. 68, 2005. [Online]. Available: <http://www.cdam.lse.ac.uk/Reports/Files/cdam-2005-18.pdf>
- [127] J. Rosenmüller, “On a Generalization of the Lemke–Howson Algorithm to Noncooperative N -Person Games,” *SIAM Journal on Applied Mathematics*, vol. 21, no. 1, pp. 73–79, Jul. 1971. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0121010>
- [128] R. W. Rosenthal, “A class of games possessing pure-strategy Nash equilibria,” *International Journal of Game Theory*, vol. 2, pp. 65–67, 1973.
- [129] T. Roughgarden and E. Tardos, “Bounding the inefficiency of equilibria in nonatomic congestion games,” *Games and Economic Behavior*, vol. 47, no. 2, pp. 389–403, May 2004. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S089982560300188X>
- [130] S. Sagratella, “Computing All Solutions of Nash Equilibrium Problems with Discrete Strategy Sets,” *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2190–2218, Jan. 2016. [Online]. Available: <http://epubs.siam.org/doi/10.1137/15M1052445>
- [131] S. Sagratella, M. Schmidt, and N. Sudermann-Merx, “The noncooperative fixed charge transportation problem,” *European Journal of Operational Research*, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377221719310483>
- [132] C. Sanderson and R. Curtin, “Armadillo: a template-based C++ library for linear algebra,” *Journal of Open Source Software*, vol. 1, no. 2, p. 26, 2016.
- [133] T. Sandholm, A. Gilpin, and V. Conitzer, “Mixed-integer programming methods for finding Nash equilibria,” in *AAAI*, 2005, pp. 495–501.
- [134] S. Sankaranarayanan, F. Feijoo, and S. Siddiqui, “Sensitivity and covariance in stochastic complementarity problems with an application to North American natural gas markets,” *European Journal of Operational Research*, vol. 268, no. 1, pp. 25–36, Jul. 2018.
- [135] H. Scarf, “The Approximation of Fixed Points of a Continuous Mapping,” *SIAM Journal on Applied Mathematics*, vol. 15, no. 5, pp. 1328–1343, Sep. 1967. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0115116>

- [136] H. D. Sherali, “A multiple leader Stackelberg model and analysis,” *Operations Research*, vol. 32, no. 2, pp. 390–404, 1984.
- [137] H. D. Sherali and W. P. Adams, *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, ser. Nonconvex Optimization and Its Applications. Boston, MA: Springer US, 1999, vol. 31. [Online]. Available: <http://link.springer.com/10.1007/978-1-4757-4388-3>
- [138] Y. Shoham, “Computer science and game theory,” *Communications of the ACM*, vol. 51, no. 8, pp. 74–79, Aug. 2008. [Online]. Available: <https://dl.acm.org/doi/10.1145/1378704.1378721>
- [139] H. Stackelberg, “Marktform und Gleichgewicht,” *Vienna: Julius Springer*, 1934.
- [140] N. D. Stein, A. Ozdaglar, and P. A. Parrilo, “Separable and low-rank continuous games,” *International Journal of Game Theory*, vol. 37, no. 4, pp. 475–504, Dec. 2008. [Online]. Available: <http://link.springer.com/10.1007/s00182-008-0129-2>
- [141] O. Stein and N. Sudermann-Merx, “The noncooperative transportation problem and linear generalized Nash games,” *European Journal of Operational Research*, vol. 266, no. 2, pp. 543–553, 2018.
- [142] S. Tadelis, *Game theory: an introduction*. Princeton ; Oxford: Princeton University Press, 2013.
- [143] J. P. Vielma, “Mixed Integer Linear Programming Formulation Techniques,” *SIAM Review*, vol. 57, no. 1, pp. 3–57, Jan. 2015. [Online]. Available: <http://epubs.siam.org/doi/10.1137/130915303>
- [144] B. Von Stengel, “Computing equilibria for two-person games,” in *Handbook of Game Theory with Economic Applications*. Elsevier, 2002, vol. 3 (45), pp. 1723–1759. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1574000502030084>
- [145] R. Weintraub, A. Bitton, and M. Rosenberg, “The Danger of Vaccine Nationalism,” *Harvard Business Review*, May 2020, accessed on March the 4th 2021. [Online]. Available: <https://hbr.org/2020/05/the-danger-of-vaccine-nationalism>
- [146] R. Wilson, “Computing Equilibria of N -Person Games,” *SIAM Journal on Applied Mathematics*, vol. 21, no. 1, pp. 80–87, Jul. 1971. [Online]. Available: <http://epubs.siam.org/doi/10.1137/0121011>

- [147] B. Yu, J. E. Mitchell, and J.-S. Pang, “Solving linear programs with complementarity constraints using branch-and-cut,” *Mathematical Programming Computation*, vol. 11, no. 2, pp. 267–310, Jun. 2019. [Online]. Available: <http://link.springer.com/10.1007/s12532-018-0149-2>

APPENDIX A THE CUT AND PLAY ALGORITHM

A.1 Proof of Theorem 4

Proof. We follow the same structure of the proof for Theorem 7 from Carvalho et al. [31]. First, since G and $\tilde{G}$ are separable games, the payoff any given player is as in (5.2). Also, from the result of Stein et al. [140], both G and $\tilde{G}$ are separable games, all the *MNEs* (thus any *PNE*) have finite supports or finite supports equivalents. For each player i , its payoff in $\tilde{G}$ and G is linear in its variables x^i .

PNE in $\tilde{G}$ is an MNE in G . Given the *PNE* $\tilde{\sigma}$ for $\tilde{G}$, we first show $\tilde{\sigma}$ is an *MNE* for G by contradiction, thus assuming $\tilde{\sigma}$ is not an *MNE* in G . Assume i has a unilateral profitable deviation from $\tilde{\sigma}^i$ to $\bar{\sigma}^i$ in G , namely it can increase its payoff by playing the mixed-strategy $\bar{\sigma}^i$. For construction, $\text{supp}(\bar{\sigma}^i)$ should contain only pure strategies that are feasible for G , i.e., for any i and $\bar{x}^i \in \text{supp}(\bar{\sigma}^i)$ it follows that $\bar{x}^i \in \mathcal{X}^i$. Furthermore, since $\bar{\sigma}^i$ is a profitable deviation for i , then $f^i(\bar{\sigma}^i, \tilde{\sigma}^{-i}) \leq f^i(\tilde{\sigma}^i, \tilde{\sigma}^{-i})$. This clearly contradicts the assumption $\tilde{\sigma}$ is a *PNE* for $\tilde{G}$. Thus, no deviation exists for either G or $\tilde{G}$.

MNE in G is a PNE in $\tilde{G}$. Let $\hat{\sigma}$ be now an *MNE* for G . We show this is a *PNE* for $\tilde{G}$ by explicitly considering the sets $\text{cl conv}(\mathcal{X}^i)$. First, $\hat{\sigma}$ is also a feasible pure strategy profile in $\tilde{G}$ since $\mathcal{X}^i \subseteq \text{cl conv}(\mathcal{X}^i)$ for any i . The following equilibrium constraint is valid in G for every player i , since we have an *MNE* $\hat{\sigma}$:

$$f^i(\bar{x}^i, \hat{\sigma}^{-i}) \geq f^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \quad \forall \bar{x}^i \in \mathcal{X}^i. \quad (\text{A.1})$$

We show (A.1) holds for any $\bar{x}^i \in \text{cl conv}(\mathcal{X}^i)$. We multiply (A.1) on both sides by an appropriately sized non-negative vector $\lambda \geq 0$, with $\sum_{j=1}^{|\lambda|} \lambda_j = 1$. Clearly, all the resulting equations hold because the payoff functions are linear in each player's variable, and the multipliers non-negative. Thus, (A.1) holds for any $\bar{x}^i \in \text{conv}(\mathcal{X}^i)$. In order to prove (A.1) holds also for the closure, we simply consider a convergent sequence of deviations $\bar{x}_1^i, \bar{x}_2^i, \dots$ where $\bar{x}_j^i \in \text{cl conv}(\mathcal{X}^i)$ for any j , and $\lim_{j \rightarrow \infty} \bar{x}_j^i = \bar{x}^i$. Then:

$$f^i(\bar{x}_j^i, \hat{\sigma}^{-i}) \geq f^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \quad \forall j. \quad (\text{A.2})$$

We apply a $\lim_{j \rightarrow \infty}$ operator to both sides, and obtain:

$$f^i(\bar{x}^i, \hat{\sigma}^{-i}) \geq f^i(\hat{\sigma}^i, \hat{\sigma}^{-i}) \quad \forall j. \quad (\text{A.3})$$

Hence, we can simply compute a *PNE* on $\tilde{G}$ and arbitrarily map it to an *MNE* of G . Furthermore, it follows that if G has no *MNE*, then $\tilde{G}$ has no *PNE*. $\square$

A.2 Proof of Proposition 3

Proof. Consider a generic player i . Note that we assume the infimum z^i is finite. From the definition of infimum, it follows that player i cannot achieve a payoff strictly less than z^i given the other players' strategies $\tilde{\sigma}^{-i}$. Consider now the optimization program of i where its feasible set is replaced with $\text{cl conv}(\mathcal{X}^i)$: player i attains a payoff of z^i at least in one point $v^i \in \text{cl conv}(\mathcal{X}^i)$. Namely, z^i is a minimum for the optimization problem

$$z^i = \min_{x^i} \{f^i(x^i, \tilde{\sigma}^{-i}) : x^i \in \text{cl conv}(\mathcal{X}^i)\}. \quad (\text{A.4})$$

Let us define $F^i = \{x^i : x^i \in \text{cl conv}(\mathcal{X}^i), f^i(x^i, \tilde{\sigma}^{-i}) = z^i\}$, which is also a face of $\text{cl conv}(\mathcal{X}^i)$ with $v^i \in F^i$. Assume there exist a best-response $\bar{v}^i \in \text{cl conv}(\mathcal{X}^i)$ so that i can improve its payoff by deviating to that strategy. Then, v^i would not be a best-response to $\tilde{\sigma}^{-i}$. Since all the solutions of (A.4) are necessarily infima in $\mathcal{X}^i$ with respect to $f^i(x^i, \tilde{\sigma}^{-i})$, this would mean that the optimal solution of (A.4) is $\bar{v}^i$, thus resulting in an absurdity. $\square$

APPENDIX B WHEN NASH MEETS STACKELBERG

In this electronic companion, we complement the proofs of Section 6.3 in appendix B.1, the pseudo-code for the *PNEs* algorithm in appendix B.3, and an overview of computational instances in appendix B.4.

B.1 Extensions to proofs of hardness

Proof. Proof of Claim 1. All the constraints are linear, and by fixing the variables of the other players, the objectives are also linear. Also, the follower is simply parameterized in their leader's variables. There are precisely two leaders, and their interaction follows the definition of a simple Nash game. Hence – by definition – the game in (6.2) is a trivial *NASP*. $\square$

Proof. *Proof of Claim 2.* Notice that the constraints in (6.2h) enforce $y_i \geq \max(-x_i, x_i - 1)$, and since y_i is minimized, it has necessarily to be equal to $\max(x_i - 1, -x_i)$. However, if this quantity should be non-negative – as enforced in (6.2c) – then either $x_i \leq 0$ or $1 - x_i \leq 0$ should hold. The claim follows. $\square$

Proof. Proof of Theorem 12. The following bilevel problem gives the necessary extended formulation. Variables $z_1, z_2, \dots$ are the variables in the lifted space, which can be projected out.

$$x \geq 0 \tag{B.1a}$$

$$y \geq 0 \tag{B.1b}$$

$$h \geq 0 \tag{B.1c}$$

$$y \leq 1 \tag{B.1d}$$

$$h \leq x \tag{B.1e}$$

$$z_1, \dots, z_6 \geq 0 \tag{B.1f}$$

$$(z_1, \dots, z_6) \in \arg \min_z \left\{ \sum_{i=1}^6 z_i : \begin{array}{l} z_1 \geq h - x \ ; \ z_1 \geq -h \\ z_2 \geq 1 - y \ ; \ z_2 \geq -h \\ z_3 \geq y - 1 \ ; \ z_3 \geq -h \\ z_4 \geq x - h \ ; \ z_4 \geq -y \\ z_5 \geq h - x \ ; \ z_5 \geq -y \\ z_6 \geq y - 1 \ ; \ z_6 \geq -y \end{array} \right\} \tag{B.1g}$$

□

Proof. Proof of Theorem 13. If S has an extended formulation given by $\{(x, y) : A_S x + B_S y \leq b_S; y \in \arg \min\{f_S^T y : C_S x + D_S y \leq g_S\}\}$, and if T has an extended formulation given by $\{(x, y) : A_T x + B_T y \leq b_T; y \in \arg \min\{f_T^T y : C_T x + D_T y \leq g_T\}\}$, then the following is an extended formulation of $S \times T$:

$$\begin{aligned} & \{(x, y, u, v) : A_S x + B_S y \leq b_S; A_T u + B_T v \leq b_T; \\ & (y, v) \in \arg \min\{f_S^T y + f_T^T v : \begin{array}{l} C_S x + D_S y \leq g_S \\ C_T u + D_T v \leq g_T \end{array}\}\} \end{aligned}$$

□

Proof. Proof of Claim 6. All constraints are linear, and the objectives are linear given the other players' decisions as parameters. The constraints (6.5h) are valid due to Theorem 12. Also, for Theorem 13, we can have multiple bilevel constraints in (6.5h) and (6.5i). Each follower is simply parameterized in their leader's variables. There are precisely two leaders, and their interaction follows the definition of a simple Nash game. □

B.2 NASP with only MNEs

Example 6. *Considering the following Latin-Greek trivial NASP.*

Latin Player

$$\max_{x, y} : x_1 \xi_1 + x_2 \xi_2 \tag{B.2a}$$

$$x, y \geq 0 \tag{B.2b}$$

$$x \leq 1 \tag{B.2c}$$

$$x_1 + x_2 = 1 \tag{B.2d}$$

$$y \in \arg \min_y \left\{ y_1 + y_2 : \begin{array}{l} y_i \geq -x_i \\ y_i \geq x_i - 1 \end{array} \text{ for } i = 1, 2 \right\} \tag{B.2e}$$

Greek Player

$$\max_{\xi, \chi} : x_2 \xi_1 + x_1 \xi_2 \quad (\text{B.2f})$$

$$\xi, \chi \geq 0 \quad (\text{B.2g})$$

$$\xi \leq 1 \quad (\text{B.2h})$$

$$\xi_1 + \xi_2 = 1 \quad (\text{B.2i})$$

$$\chi \in \arg \min_{\chi} \left\{ \chi_1 + \chi_2 : \begin{array}{l} \chi_i \geq -\xi_i \\ \chi_i \geq \xi_i - 1 \end{array} \text{ for } i = 1, 2 \right\} \quad (\text{B.2j})$$

The only feasible decisions for both the Latin and the Greek player in (B.2) are $\{(1, 0, 0, 0), (0, 1, 0, 0)\}$. So the game can be written as a normal form game. We can compute the payoffs for these finitely many strategies so that if the Latin and the Greek player choose the same strategy, then the Latin player gets a payoff of 1 and the Greek player gets a payoff of 0. If they choose different strategies, the Latin player gets a payoff of 0, and the Greek player gets a payoff of 1. One can easily check that this game's unique Nash equilibrium is an MNE and that no PNE exists.

B.3 Enumeration algorithm for PNEs

Algorithm 7 reports the pseudo-code for the algorithm described in Section 6.5.2.

B.4 Computations

Governments act as Stackelberg leaders by trading energy, intending to minimize emissions, and eventually maximize tax incomes. Energy producers act as Stackelberg followers within each country and play a Nash game between themselves, aiming to maximize their profits. Each country is interested in imposing a tax that prevents profitable domestic production, as it is constrained to keep the domestic energy price less than a predetermined threshold. We present the optimization problems of the players formally below. For ease of understanding, the quantities in **red** are parameters, i.e., inputs to the model. Furthermore, the quantities in **blue** are decision variables, decided by the country or of the energy producers in the same country. Quantities in **green** are variables in a problem but not decided by the country in

Algorithm 7: Enumeration algorithm to obtain a *PNE* for a *NASP*

Data: A description of *NASP* $N = (P^1, \dots, P^n)$

Result: For each $i = 1, \dots, n$, a pure-strategy $\hat{x}^i$, such that the strategy profile is a *PNE* or a proof that no *PNE* exists

```

1 for  $i = 1, \dots, n$  do
2   Enumerate the polyhedra whose union defines the feasible set  $\mathcal{F}_i$  of  $P^i$ ;
3    $\tilde{\mathcal{F}}_i \leftarrow \text{cl conv } \mathcal{F}_i$  by applying Theorem 8;
4    $\tilde{P}^i \leftarrow$  objective function of  $P^i$  and a feasible set of  $\tilde{\mathcal{F}}_i$ ;
5 Let  $\tilde{N} = (\tilde{P}^1, \dots, \tilde{P}^n)$  be the facile Nash game ;
6 Enforce  $\delta_j^i$  for  $i = 1, \dots, n, j = 1, \dots, g^i$  in  $\tilde{N}$  to be binary and solve  $\tilde{N}$ ;
7 if  $\tilde{N}$  is infeasible then
8   There is no PNE for  $N$ ; return failure
9 else return Project the solution of  $\tilde{N}$  to the space of the original variables of  $N$  ;

```

consideration. Each country C solves the following problem.

$$\min_{\substack{\mathbf{q}^p, \mathbf{t}^p, \\ \mathbf{q}_{\text{imp}}^{C' \rightarrow C}, \mathbf{q}_{\text{exp}}^C}} : \left(\sum_{p \in \mathcal{P}} \mathbf{C}_{\text{emmission}}^p \mathbf{q}^p - b \mathbf{t}^p \mathbf{q}^p \right) + \sum_{C' \in \mathcal{C} \setminus C} \pi^C \mathbf{q}_{\text{imp}}^{C' \rightarrow C} - \pi^C \mathbf{q}_{\text{exp}}^C \quad (\text{B.3a})$$

$$\text{subject to} \quad \mathbf{t}^p \leq \bar{\mathbf{t}}^p \quad (\text{B.3b})$$

$$\alpha^C - \beta^C \left(\sum_{p \in \mathcal{P}} \mathbf{q}^p + \mathbf{q}_{\text{imp}}^C - \mathbf{q}_{\text{exp}}^C \right) \geq \underline{\pi}^C \quad (\text{B.3c})$$

$$\sum_{C' \in \mathcal{C}} \mathbf{q}_{\text{imp}}^{C' \rightarrow C} = \mathbf{q}_{\text{imp}}^C \quad (\text{B.3d})$$

$$\mathbf{q}^p \in \text{SOL}(\text{Lower Level Nash Game}) \quad (\text{B.3e})$$

$\mathbf{C}_{\text{emmission}}^p$ is the dollar value of the emission caused by producer p while producing a unit quantity of energy. This number is the product of cost incurred due to the emission of one unit of greenhouse gases (*GHG*), sometimes referred to as the *social cost of carbon* and the quantity of GHG emitted for each unit of energy produced by the producer p , called as the *emission factor*. b dictates whether the objective should include the tax revenue earned by the government or not. $\mathbf{q}^p$ is the quantity of energy produced by the producer $p \in \mathcal{P}$, $\mathbf{q}_{\text{imp}}^C, \mathbf{q}_{\text{exp}}^C$ are respectively import and export quantities, and α^C, β^C are the intercept and the slope of the demand curve. $\alpha^C - \beta^C Q$ is the domestic price for each country, where Q is the quantity of energy available domestically. Finally, π^C is the price at which the country can import energy from other countries, hence the variable linking the optimization problems of different

countries. Thus, π^C can be interpreted as the shadow price to the *market-clearing constraint*

$$\sum_{C' \in \mathcal{C}} \mathbf{q}_{\text{imp}}^{C \rightarrow C'} = \sum_{C \in \mathcal{C}} \mathbf{q}_{\text{exp}}^C. \quad (\text{B.3f})$$

We note that including (B.3f) does not make the game into a generalized Nash game. This is because we can rewrite (B.3f) as the KKT conditions of a fictitious optimization problem, generally referred to as the *invisible hand* in the market. An alternative manner of looking at this is as if there is perfect competition in the international energy markets and the most efficient allocation of resources happens. This is again a standard simplifying assumption considered, for example, in Egging et al. [61, 62], Feijoo et al. [69], Gabriel and Leuthold [74], Sankaranarayanan et al. [134].

Optionally for some countries, as a domestic policy, we introduce a *carbon tax* paradigm, where the tax imposed on the followers is proportional to the emissions they cause. i.e., there is a constraint $\mathbf{t}^p = \mathbf{C}_{\text{emission}}^p \mathbf{t}^{\text{GHG}}$, where the government decides the tax payable per unit emission. Furthermore, note that if $\mathbf{b}$ is non-zero, the objective is no longer linear. In such a case, we replace the product term with a McCormick relaxation. Finally, $\overline{\mathbf{t}}^p$, and $\underline{\pi}^C$ are the tax and price caps, respectively. The lower level problem that each producer p solves is as follow:

$$\min_{\mathbf{q}^p} : \quad \mathbf{C}_p \mathbf{q}^p + \frac{1}{2} \mathbf{D}_p \mathbf{q}^{p^2} + \mathbf{t}^p \mathbf{q}^p - \left(\alpha^C - \beta^C \left(\sum_{p' \in \mathcal{P}} \mathbf{q}^{p'} + \mathbf{q}_{\text{imp}}^C - \mathbf{q}_{\text{exp}}^C \right) \right) \mathbf{q}^p \quad (\text{B.4a})$$

$$\text{subject to} \quad \mathbf{q}^p \geq 0 \quad (\text{B.4b})$$

$$\mathbf{q}^p \leq \overline{\mathbf{q}}^p \quad (\text{B.4c})$$

The first two terms in the objective correspond to the energy producer's cost, while the third term is the tax expense. The parenthesis results in the revenue of p , which is the product of domestic price and the quantity produced. Further, the producer has a constraint on its capacity limit. Note that the product of variables ($\mathbf{t}^p \mathbf{q}^p$) in the objective does not pose any additional difficulty to the problem. This is because the follower's problem is still convex quadratic for a *fixed* value of $\mathbf{t}^p$, and the *KKT* conditions give complementarity constraints with only linear terms.

Further, we also note that the previously-mentioned assumption of *optimistic* equilibrium selection by the leaders and the limitations imposed, as a result, are irrelevant here. This is because, for $\mathbf{D}_p > 0$, which is always the case, the test examples have a *unique* lower-level equilibrium. Thus any technique for equilibrium selection is not warranted.

B.4.1 Instance sets

We generated three instances sets for our tests. (i) *InstanceSetA* contains 149 instances where there are 3 to 5 countries (ii) *InstanceSetB* contains 50 instances with strictly 7 countries. These instances were selected if Algorithm 5 was not able to solve them within 10 second on a single core machine. (iii) *InstanceSetInsights* contains 50 instances with 2 countries with 3 followers each. Such instances are useful to derive managerial insights from our model. The specific parameters for all these instances are described in Table B.1 and are available in our open-source GitHub repository. All our tests run on a 8-cores Intel(R) Xeon Gold 6142, with 32GB of RAM and *Gurobi* 9.0.

Table B.1 Description of the parameters for *EPEC* instances.

Parameter	Distribution	Notes
Capacities	50, 100, 130, 170, 200, 1000, 1050, 20000	Each follower's capacity is randomly drawn from these values.
Emission Costs	25, 50, 100, 200, 300, 500, 550, 600	The first two values are reserved for green producers. The following two for averagely-polluting producers, while the remaining three for highly-polluting ones.
Linear Costs	150, 200, 220, 250, 275, 290, 300	Linear costs are generally inversely proportional to the emission cost. For instance, a producer with a 50 emission cost will generally have a linear cost around 290.
Quadratic Costs	0, 0.1, 0.2, 0.3, 0.5, 0.55, 0.6	Quadratic costs are generally inversely proportional to the emission cost. Same rationale as linear costs.
Tax Caps	0, 50, 100, 150, 200, 250, 275, 300	Tax caps are assigned following the same rational of emission costs. The lower the emission cost of a given producer, the lower the maximum tax applicable to it.
Demand Alpha	275, 300, 325, 350, 375, 450	Each country alpha is randomly drawn from this set.
Demand Beta	0.5, 0.6, 0.7, 0.75, 0.8, 0.9	Each country beta is randomly drawn from this set.
Price Cap	0.8, 0.85, 0.90, 0.95	Each country price-limit is randomly drawn from this set. The final price-limit is made of the product of this value and the country's demand alpha parameter.
Tax Paradigm	Standard, Single, Carbon	A country tax scheme can be: (i.) Standard, where followers are taxed at different levels per unit-energy, or (ii.) Single, where all the followers are taxed with the same level per unit-energy (iii.) Carbon, where all the followers are taxed with the same level per unit-emission

B.4.2 Results tables

Tables B.2 and B.3 contains the full results for *InstanceSetA* and *InstanceSetB*, respectively. The first three columns are the instance number, the number of leaders, and – for each leader – their respective number of followers in squared parenthesis. The *MNE* column is the status of the instance, namely if it has an equilibrium (*YES*), if it does not (*NO*), or if the time limit was triggered (*TL*) for all the methods. In the remaining column, we report the clock time and the status for each algorithmic configuration. In particular, we have Algorithm 5 (*FE*),

and the inner approximations. We report three extension strategies, namely the sequential (*seq*), the reverse sequential (*rseq*), and the random one (*rand*). They are followed by their respective parameter k , as reported in Section 6.6. The last two columns are related to *PNEs*.

Table B.4 reports the results for *InstanceSet Insights*. The first column reports each instance's number. The second and third column are boolean values reporting whether the tax (T^a) and the trade (T^r) are allowed (value of 1) or not. The following 16 columns are results for the first country (*Country One*), while the remaining 16 are for the second country (*Country Two*). Following the column order, for each country we report: the unit-energy production level *Prod*, the domestic price per unit-energy $\$(E)$, the import *Imp* and export *Exp* unit-energies, the export price $\$(E)$, and the tax per unit-emission *Tax*. Furthermore, for each of the 3 followers of each country, we have the type *Ty* (*C* for coal, *G* for gas, or *S* for solar), the associated emission cost per unit-energy *E*, and its production *Prod*.

Table B.2: *MNE* and *PNE* results for *InstanceSetA*. Columns: # - Instance Number. L - Number of leaders in the instance. F - Number of followers each leader has. *FE* - Time taken for full enumeration algorithm. *seq1* to *rand5* - Time taken for inner approximation with different extension strategies. *MNE*- existence (or time limit reached TL). *FE-P* - Time for full enumeration to find a *PNE*. *PNE*- existence (or time limit reached *TL*).

#	L	F	FE	seq1	seq3	seq5	rseq1	rseq3	rseq5	rand1	rand3	rand5	MNE	FE-P	PNE
1	3	[1 2 2]	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	NO	0.04	NO
2	3	[2 2 3]	0.07	0.15	0.08	0.08	0.16	0.08	0.08	0.16	0.08	0.08	NO	0.07	YES
3	3	[2 2 2]	0.05	0.13	0.06	0.06	0.13	0.06	0.06	0.13	0.06	0.06	NO	0.05	NO
4	3	[1 2 1]	0.04	0.10	0.05	0.05	0.10	0.05	0.05	0.10	0.05	0.05	NO	0.04	TL
5	3	[2 1 3]	0.33	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	YES	0.09	YES
6	3	[2 2 1]	0.06	0.18	0.11	0.07	0.18	0.11	0.07	0.18	0.11	0.07	NO	0.06	YES
7	3	[2 2 2]	0.06	0.14	0.07	0.07	0.14	0.07	0.07	0.14	0.07	0.07	NO	0.06	YES
8	3	[1 2 1]	0.10	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	YES	0.08	NO
9	3	[1 2 2]	0.05	0.10	0.06	0.06	0.10	0.06	0.06	0.10	0.06	0.06	NO	0.04	NO
10	3	[2 2 1]	0.16	0.24	0.17	0.17	0.13	0.19	0.20	0.24	0.20	0.20	YES	0.16	NO
11	3	[3 2 2]	0.08	0.22	0.15	0.09	0.23	0.14	0.09	0.22	0.14	0.09	NO	0.08	NO
12	3	[2 2 2]	0.76	1.50	1.41	1.07	0.38	0.28	0.79	1.72	1.77	0.55	YES	2.07	YES
13	3	[2 1 2]	0.04	0.07	0.05	0.05	0.07	0.05	0.05	0.07	0.05	0.05	NO	0.04	NO
14	3	[2 1 3]	0.06	0.13	0.07	0.07	0.13	0.07	0.07	0.13	0.07	0.07	NO	0.06	NO
15	3	[2 1 2]	0.39	0.08	0.08	0.08	0.09	0.08	0.08	0.08	0.08	0.08	YES	0.18	NO
16	3	[2 3 2]	5.73	5.07	7.68	4.83	0.37	1.48	12.19	13.49	1370.28	1.30	YES	TL	YES
17	3	[2 2 2]	0.06	0.13	0.07	0.07	0.13	0.07	0.07	0.13	0.07	0.07	NO	0.06	YES
18	3	[2 2 2]	0.79	0.15	0.16	0.15	0.15	0.15	0.15	0.15	0.15	0.15	YES	3.17	NO
19	3	[2 2 2]	0.07	0.16	0.08	0.08	0.16	0.08	0.08	0.16	0.08	0.08	NO	0.07	NO
20	3	[1 3 2]	0.05	0.09	0.06	0.06	0.09	0.06	0.06	0.09	0.06	0.06	NO	0.05	NO
21	3	[1 2 2]	0.04	0.11	0.06	0.06	0.12	0.06	0.06	0.11	0.06	0.06	NO	0.05	YES
22	3	[2 1 1]	0.04	0.10	0.05	0.05	0.10	0.05	0.05	0.10	0.05	0.05	NO	0.04	NO
23	3	[1 2 1]	0.09	0.13	0.11	0.11	0.10	0.11	0.11	0.13	0.11	0.11	YES	0.07	YES
24	3	[2 2 1]	0.05	0.14	0.09	0.06	0.14	0.09	0.06	0.14	0.09	0.06	NO	0.05	YES
25	3	[2 2 2]	0.14	0.24	0.17	0.17	0.14	0.16	0.17	0.12	0.17	0.17	YES	0.12	NO
26	3	[2 2 2]	0.06	0.15	0.07	0.07	0.16	0.07	0.07	0.15	0.08	0.07	NO	0.06	NO
27	3	[1 1 2]	0.10	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	YES	0.12	NO
28	3	[3 1 3]	0.26	1.10	0.55	0.96	0.10	0.33	0.53	0.18	0.35	0.28	YES	TL	NO
29	3	[1 1 2]	0.05	0.15	0.09	0.06	0.15	0.09	0.06	0.15	0.09	0.06	NO	0.05	YES
30	3	[1 1 1]	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	NO	0.03	YES
31	3	[2 2 1]	0.53	7.04	0.58	0.55	0.58	0.37	0.57	0.58	0.48	0.60	YES	TL	YES
32	3	[1 1 3]	0.11	0.19	0.14	0.15	0.10	0.14	0.14	0.19	0.15	0.15	YES	0.11	YES
33	3	[3 2 3]	0.07	0.15	0.08	0.08	0.15	0.08	0.08	0.15	0.08	0.08	NO	0.07	TL
34	3	[1 2 2]	0.42	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	YES	1.03	TL
35	3	[2 2 2]	0.06	0.14	0.07	0.07	0.14	0.07	0.07	0.14	0.07	0.07	NO	0.06	NO
36	3	[1 1 3]	0.57	2.69	2.64	6.37	0.67	19.02	3.33	1.68	316.94	0.68	YES	TL	TL
37	3	[2 1 3]	0.25	0.20	0.25	0.25	0.34	0.20	0.20	0.35	0.19	0.26	YES	0.15	NO

Table B.2: *Continued. MNE and PNE results for InstanceSetA.*

38	3	[2 2 2]	0.06	0.20	0.10	0.07	0.20	0.10	0.07	0.20	0.10	0.07	NO	0.06	NO
39	3	[3 3 1]	0.07	0.16	0.08	0.08	0.16	0.08	0.08	0.16	0.08	0.08	NO	0.07	TL
40	3	[2 1 2]	0.13	0.24	0.18	0.18	0.15	0.17	0.17	0.19	0.17	0.18	YES	0.14	YES
41	3	[2 2 3]	0.08	0.21	0.14	0.08	0.22	0.14	0.09	0.22	0.14	0.09	NO	0.07	NO
42	3	[1 1 1]	0.03	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	NO	0.03	NO
43	3	[1 3 2]	1.02	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	0.10	YES	TL	YES
44	3	[3 2 2]	0.10	0.42	0.18	0.20	0.42	0.18	0.20	0.42	0.18	0.19	NO	0.10	NO
45	3	[2 2 1]	0.05	0.13	0.06	0.06	0.13	0.06	0.06	0.13	0.06	0.06	NO	0.05	TL
46	3	[1 2 2]	0.05	0.12	0.06	0.06	0.12	0.06	0.06	0.12	0.06	0.06	NO	0.05	NO
47	3	[1 3 2]	0.40	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	0.03	YES	0.58	NO
48	3	[2 2 1]	0.04	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	NO	0.04	YES
49	3	[2 1 1]	0.04	0.11	0.07	0.05	0.11	0.07	0.05	0.11	0.07	0.05	NO	0.04	NO
50	4	[1 2 1 2]	0.18	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	YES	0.17	NO
51	4	[1 1 1 1]	0.06	0.11	0.07	0.07	0.11	0.07	0.07	0.11	0.07	0.07	NO	0.06	NO
52	4	[3 1 3 1]	0.10	0.17	0.11	0.11	0.17	0.11	0.11	0.18	0.11	0.11	NO	0.10	NO
53	4	[3 1 2 1]	775.55	0.36	0.36	0.36	0.36	0.36	0.35	0.36	0.36	0.37	YES	TL	TL
54	4	[1 1 2 3]	0.09	0.22	0.10	0.10	0.22	0.10	0.10	0.22	0.10	0.10	NO	0.09	NO
55	4	[1 2 1 2]	0.14	0.43	0.36	0.36	0.16	0.18	0.18	0.16	0.18	0.36	YES	0.17	NO
56	4	[2 2 1 2]	1.69	0.20	0.20	0.20	0.20	0.21	0.20	0.20	0.20	0.20	YES	TL	NO
57	4	[1 2 2 2]	0.29	0.42	0.28	0.28	0.21	0.31	0.31	0.48	0.45	0.31	YES	0.64	YES
58	4	[2 2 2 1]	0.09	0.24	0.11	0.11	0.24	0.11	0.11	0.24	0.11	0.11	NO	0.09	YES
59	4	[1 2 2 1]	0.09	0.22	0.11	0.11	0.23	0.11	0.11	0.22	0.11	0.11	NO	0.09	NO
60	4	[1 3 1 3]	0.09	0.16	0.11	0.11	0.16	0.11	0.11	0.16	0.11	0.11	NO	0.10	YES
61	4	[3 1 3 2]	38.83	TL	TL	17.70	0.31	2.64	54.99	94.38	1.36	TL	YES	152.48	NO
62	4	[1 1 3 2]	24.30	0.80	0.79	0.79	0.79	0.80	0.80	0.79	0.80	0.80	YES	TL	NO
63	4	[2 3 2 3]	0.32	0.70	0.41	0.41	0.24	0.36	0.36	0.47	0.41	0.42	YES	0.25	NO
64	4	[2 2 3 1]	0.18	0.76	0.32	0.34	0.77	0.32	0.34	0.76	0.32	0.34	NO	0.18	NO
65	4	[2 1 3 2]	1.32	2.63	2.25	1.89	0.42	1.26	1.27	3.06	1.80	1.64	YES	1.61	NO
66	4	[3 3 3 3]	0.58	1.34	0.77	0.82	0.37	0.88	0.60	0.66	0.57	0.50	YES	0.44	YES
67	4	[3 2 2 1]	0.12	0.44	0.22	0.14	0.44	0.22	0.13	0.44	0.22	0.14	NO	0.12	NO
68	4	[3 2 2 2]	1.90	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	YES	TL	YES
69	4	[2 1 3 3]	0.12	0.28	0.13	0.13	0.28	0.13	0.13	0.28	0.13	0.13	NO	0.12	NO
70	4	[2 2 2 1]	0.08	0.16	0.10	0.10	0.16	0.10	0.10	0.16	0.10	0.10	NO	0.09	YES
71	4	[1 2 2 2]	0.08	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	0.09	NO	0.08	YES
72	4	[1 2 1 3]	0.10	0.32	0.20	0.12	0.33	0.20	0.12	0.33	0.20	0.12	NO	0.10	NO
73	4	[2 2 2 2]	160.82	12.66	12.80	12.71	12.65	12.81	12.64	12.75	12.83	12.66	YES	TL	NO
74	4	[2 1 1 3]	0.16	0.85	0.37	0.28	0.85	0.37	0.28	0.85	0.39	0.28	NO	0.16	NO
75	4	[1 2 1 3]	0.10	0.24	0.11	0.11	0.24	0.11	0.11	0.24	0.11	0.11	NO	0.10	YES
76	4	[1 1 1 2]	0.06	0.11	0.07	0.07	0.11	0.07	0.07	0.11	0.07	0.07	NO	0.06	YES
77	4	[3 1 2 2]	0.14	0.52	0.26	0.16	0.52	0.26	0.16	0.52	0.26	0.16	NO	0.14	NO
78	4	[2 1 1 2]	0.24	0.34	0.28	0.28	0.21	0.28	0.28	0.21	0.28	0.28	YES	0.25	YES
79	4	[3 2 1 3]	54.99	297.65	297.34	296.37	4.68	415.64	422.42	35.04	57.32	61.47	YES	TL	NO
80	4	[2 1 1 2]	0.07	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	NO	0.07	YES
81	4	[3 2 2 2]	0.19	0.80	0.33	0.36	0.81	0.33	0.37	0.81	0.33	0.37	NO	0.19	YES
82	4	[2 3 1 1]	0.23	0.64	0.32	0.26	0.17	0.22	0.26	0.23	0.19	0.26	YES	0.40	NO
83	4	[2 1 2 2]	TL	1.50	1.36	TL	8.73	51.28	TL	8.74	76.26	1209.76	YES	0.88	NO
84	4	[3 3 1 2]	0.26	1.89	0.79	0.66	1.90	0.80	0.66	1.91	0.79	0.66	NO	0.26	NO
85	4	[1 1 1 2]	0.07	0.12	0.08	0.08	0.12	0.08	0.08	0.12	0.08	0.08	NO	0.07	YES
86	4	[1 1 1 3]	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.14	0.15	YES	0.21	NO
87	4	[2 3 1 2]	0.42	34.44	34.31	0.46	0.74	0.42	0.44	0.58	TL	0.49	YES	TL	NO
88	4	[1 2 2 1]	0.18	0.33	0.22	0.22	0.18	0.21	0.22	0.25	0.21	0.22	YES	0.18	YES
89	4	[1 1 2 1]	0.05	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06	NO	0.05	NO
90	4	[3 3 2 1]	0.47	0.51	0.33	0.33	0.24	0.32	0.31	0.52	0.34	0.32	YES	0.71	NO
91	4	[3 1 3 2]	TL	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	0.38	YES	TL	NO
92	4	[3 3 2 3]	0.46	0.92	0.73	0.86	0.23	0.57	0.48	0.46	0.36	0.40	YES	1.00	YES
93	4	[2 2 2 2]	0.41	0.66	0.51	0.40	0.60	0.44	0.46	0.70	0.85	0.43	YES	0.32	NO
94	4	[2 2 3 2]	1.00	3.28	2.03	2.86	0.44	1.74	1.23	3.31	57.72	2.65	YES	15.24	NO
95	4	[1 1 2 2]	0.08	0.20	0.09	0.09	0.20	0.09	0.09	0.20	0.09	0.09	NO	0.08	YES
96	4	[1 2 1 1]	0.06	0.11	0.07	0.07	0.11	0.07	0.07	0.11	0.07	0.07	NO	0.06	YES
97	4	[2 2 2 2]	0.22	0.32	0.26	0.26	0.22	0.25	0.25	0.32	0.25	0.26	YES	0.22	NO
98	4	[1 1 2 1]	0.27	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	0.18	YES	98.14	NO
99	4	[2 2 2 2]	0.17	0.63	0.32	0.19	0.64	0.32	0.19	0.64	0.32	0.19	NO	0.17	YES
100	5	[2 2 2 1 1]	0.15	0.38	0.17	0.16	0.39	0.16	0.17	0.38	0.17	0.17	NO	0.15	NO
101	5	[2 3 3 2 2]	TL	TL	TL	TL	1.82	TL	TL	TL	TL	TL	YES	TL	YES
102	5	[2 2 2 3 3]	4.45	2.36	1.14	1.36	4.71	2.84	41.97	51.47	1.35	3.76	YES	TL	YES
103	5	[1 2 3 1 2]	1.29	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	YES	TL	YES
104	5	[1 3 1 1 1]	0.37	0.31	0.33	0.33	0.42	0.31	0.31	0.60	0.31	0.33	YES	0.23	NO
105	5	[2 3 2 2 1]	0.32	1.76	0.77	0.58	1.78	0.77	0.58	1.78	0.77	0.58	NO	0.32	NO
106	5	[2 2 1 2 2]	0.19	0.48	0.21	0.21	0.48	0.20	0.20	0.48	0.20	0.20	NO	0.19	NO
107	5	[1 2 3 2 1]	0.21	0.81	0.38	0.23	0.83	0.38	0.23	0.83	0.38	0.23	NO	0.21	NO
108	5	[3 2 2 1 1]	0.14	0.35	0.17	0.16	0.35	0.17	0.16	0.36	0.16	0.17	NO	0.15	YES
109	5	[2 2 2 3 1]	582.94	2.12	2.09	2.11	2.11	2.12	2.11	2.10	2.11	2.10	YES	TL	YES

Table B.2: *Continued. MNE and PNE results for InstanceSetA.*

110	5	[3 2 2 3 3]	0.29	0.67	0.33	0.32	0.69	0.32	0.32	0.67	0.32	0.32	NO	0.30	YES
111	5	[2 1 3 1 3]	0.21	0.66	0.41	0.23	0.66	0.41	0.23	0.66	0.41	0.23	NO	0.21	NO
112	5	[1 2 1 3 2]	0.49	0.74	0.46	0.50	0.32	0.43	0.51	0.75	0.43	0.52	YES	1.11	YES
113	5	[1 1 2 1 1]	0.34	0.76	0.38	0.38	0.25	0.38	0.38	0.51	0.38	0.38	YES	0.53	YES
114	5	[1 3 3 1 1]	0.21	0.53	0.23	0.23	0.53	0.23	0.23	0.53	0.23	0.23	NO	0.22	NO
115	5	[3 1 2 3 2]	0.24	0.59	0.26	0.26	0.59	0.26	0.26	0.59	0.26	0.27	NO	0.24	YES
116	5	[1 2 2 3 2]	0.16	0.30	0.18	0.18	0.30	0.18	0.18	0.30	0.18	0.18	NO	0.16	NO
117	5	[2 2 1 2 2]	6.07	20.36	5.18	5.01	1.09	1.79	2.81	0.34	27.81	3.09	YES	4.20	NO
118	5	[2 2 2 2 2]	0.20	0.47	0.22	0.22	0.48	0.22	0.22	0.48	0.22	0.22	NO	0.20	NO
119	5	[3 3 2 1 1]	1.61	2.03	25.49	6.88	0.68	1.86	1.34	12.00	3.35	47.53	YES	TL	NO
120	5	[3 2 1 2 2]	0.19	0.50	0.22	0.21	0.50	0.22	0.22	0.49	0.22	0.22	NO	0.20	NO
121	5	[1 2 1 2 1]	0.14	0.35	0.16	0.15	0.36	0.15	0.16	0.36	0.16	0.15	NO	0.13	YES
122	5	[1 2 2 2 3]	0.20	0.51	0.23	0.23	0.51	0.23	0.23	0.52	0.23	0.23	NO	0.21	NO
123	5	[2 2 2 3 2]	0.21	0.53	0.23	0.23	0.53	0.23	0.23	0.53	0.23	0.24	NO	0.21	NO
124	5	[1 1 2 1 3]	2.10	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	YES	TL	NO
125	5	[2 3 2 1 2]	2.12	0.50	0.51	0.50	0.49	0.50	0.50	0.50	0.51	0.51	YES	TL	NO
126	5	[2 2 2 2 2]	0.95	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.48	0.48	YES	TL	YES
127	5	[2 1 2 1 2]	0.30	0.52	0.33	0.33	0.22	0.30	0.30	0.51	0.29	0.33	YES	0.23	YES
128	5	[2 1 2 2 3]	TL	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	0.48	YES	TL	YES
129	5	[2 2 3 3 3]	TL	TL	TL	TL	204.95	TL	TL	TL	TL	83.46	YES	TL	NO
130	5	[3 2 1 2 2]	TL	TL	TL	TL	65.01	TL	TL	74.42	TL	TL	YES	TL	NO
131	5	[2 1 2 2 1]	0.21	0.68	0.40	0.23	0.68	0.41	0.23	0.68	0.41	0.23	NO	0.21	YES
132	5	[2 2 1 1 2]	0.19	0.48	0.21	0.21	0.48	0.21	0.21	0.48	0.21	0.21	NO	0.19	TL
133	5	[2 2 2 2 3]	TL	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	YES	TL	NO
134	5	[2 2 1 2 2]	5.16	1.04	0.75	1.12	0.72	170.73	TL	0.72	TL	TL	YES	0.87	NO
135	5	[2 2 1 2 1]	0.14	0.36	0.16	0.16	0.36	0.16	0.16	0.36	0.16	0.16	NO	0.14	YES
136	5	[2 2 1 3 2]	0.24	0.94	0.45	0.26	0.94	0.45	0.26	0.94	0.45	0.26	NO	0.24	YES
137	5	[2 2 2 2 1]	0.40	0.88	0.61	0.61	0.35	0.48	0.47	0.65	0.45	0.45	YES	0.29	YES
138	5	[2 2 2 1 2]	0.26	1.37	0.69	0.49	1.38	0.69	0.49	1.37	0.69	0.49	NO	0.27	NO
139	5	[3 2 1 2 3]	TL	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	0.58	YES	TL	NO
140	5	[2 2 1 1 2]	0.11	0.21	0.13	0.13	0.21	0.13	0.13	0.22	0.13	0.13	NO	0.11	YES
141	5	[1 2 1 2 1]	0.62	0.94	1.13	0.67	0.33	0.58	0.71	0.72	0.95	0.99	YES	1.15	TL
142	5	[2 1 2 2 1]	0.21	0.83	0.39	0.23	0.84	0.39	0.22	0.83	0.39	0.22	NO	0.21	YES
143	5	[1 2 1 2 2]	0.19	0.46	0.20	0.21	0.46	0.20	0.20	0.46	0.20	0.20	NO	0.19	YES
144	5	[1 3 1 2 2]	0.17	0.31	0.19	0.19	0.31	0.19	0.19	0.31	0.19	0.19	NO	0.17	YES
145	5	[1 2 2 2 1]	0.16	0.42	0.18	0.18	0.42	0.18	0.18	0.42	0.18	0.18	NO	0.17	NO
146	5	[3 2 2 1 2]	TL	3.17	688.67	20.60	203.07	TL	TL	2.27	TL	21.66	YES	TL	NO
147	5	[1 1 2 2 3]	0.16	0.29	0.18	0.18	0.30	0.18	0.18	0.30	0.18	0.18	NO	0.16	YES
148	5	[2 2 2 2 3]	0.25	0.62	0.28	0.27	0.62	0.27	0.28	0.62	0.27	0.27	NO	0.25	YES
149	5	[2 1 1 2 2]	0.13	0.35	0.15	0.15	0.35	0.15	0.15	0.35	0.15	0.15	NO	0.13	NO

Table B.3: *MNE and PNE results for InstanceSetB. Same notation as Table B.2.*

#	L	F	FE	seq	seq3	seq5	rseq1	rseq3	rseq5	rand1	rand3	rand5	MNE	FE-P	PNE
0	7	[1 1 2 2 2 2 2]	TL	TL	TL	TL	TL	8.86	TL	240.99	TL	TL	YES	TL	TL
1	7	[1 1 1 2 1 2 1]	62.12	2.82	1.93	102.97	1.64	6.91	279.06	1.87	5.98	83.84	YES	TL	TL
2	7	[2 3 2 1 3 2 2]	TL	6.28	6.29	6.32	6.43	6.42	6.35	6.28	6.35	6.32	YES	TL	TL
3	7	[1 2 2 1 1 3 1]	572.69	0.51	0.50	0.50	0.51	0.50	0.50	0.50	0.50	0.50	YES	TL	TL
4	7	[1 3 2 2 1 2 1]	TL	1.34	1.33	1.33	1.33	1.35	1.32	1.33	1.35	1.32	YES	TL	TL
5	7	[2 1 2 2 1 2 2]	192.37	0.28	0.27	0.28	0.28	0.28	0.27	0.27	0.27	0.27	YES	2.40	YES
6	7	[2 2 2 3 3 2 1]	TL	TL	TL	TL	20.02	TL	TL	TL	TL	TL	YES	TL	TL
7	7	[2 2 1 1 3 2 3]	TL	TL	TL	TL	464.21	TL	TL	TL	TL	TL	YES	TL	TL
8	7	[3 3 1 1 3 3 3]	TL	0.38	0.38	0.39	0.38	0.38	0.39	0.38	0.39	0.39	YES	TL	TL
9	7	[2 1 2 2 2 1 2]	TL	2.45	2.46	2.45	2.43	2.47	2.46	2.47	2.46	2.46	YES	TL	TL
10	7	[2 3 2 2 1 2 2]	TL	TL	TL	TL	1704.42	TL	TL	TL	TL	TL	YES	TL	TL
11	7	[1 2 3 1 3 2 1]	TL	TL	418.96	TL	2.81	8.91	TL	4.17	TL	TL	YES	TL	TL
12	7	[3 2 2 1 2 1 1]	9.29	3.67	2.70	8.17	1.38	4.59	TL	4.86	9.14	3.90	YES	TL	TL
13	7	[2 3 3 1 1 1 2]	TL	251.78	250.48	64.79	687.63	TL	22.45	TL	21.43	31.46	YES	15.66	YES
14	7	[3 2 2 2 2 2 2]	TL	TL	TL	TL	234.33	TL	TL	480.56	TL	TL	YES	TL	TL
15	7	[2 2 3 2 2 2 1]	10.65	2.09	1.12	16.33	0.69	19.91	4.13	5.11	5.25	18.68	YES	2.24	YES
16	7	[2 2 2 2 1 2 2]	TL	162.89	162.19	TL	14.14	51.75	TL	82.21	975.49	TL	YES	TL	TL
17	7	[3 3 1 3 2 2 1]	635.27	TL	TL	TL	1.81	TL	TL	6.88	92.29	TL	YES	TL	TL
18	7	[1 1 2 2 2 2 2]	TL	TL	TL	TL	TL	8.79	TL	TL	10.92	269.57	YES	TL	TL
19	7	[2 1 1 3 2 3 2]	0.97	8.99	3.57	2.28	9.00	3.57	2.30	8.96	3.59	2.30	NO	0.97	NO
20	7	[2 2 3 2 2 2 1]	10.77	2.12	1.12	16.32	0.70	20.15	4.08	2.56	4.72	5.82	YES	2.24	YES
21	7	[3 2 2 2 2 2 2]	TL	TL	TL	TL	231.01	TL	TL	TL	TL	TL	YES	TL	TL
22	7	[2 1 1 2 3 1 3]	1.27	10.29	4.19	3.39	10.33	4.16	3.40	10.35	4.16	3.41	NO	1.26	NO

Table B.3: *Continued. MNE and PNE results for InstanceSetB.*

23	7	[2 3 3 1 1 1 2]	TL	247.80	248.01	64.28	674.37	TL	22.26	TL	56.89	TL	YES	15.73	YES
24	7	[1 1 2 2 2 2 2]	TL	TL	TL	TL	TL	8.85	TL	40.14	TL	72.52	YES	TL	TL
25	7	[3 3 1 3 2 2 1]	634.30	TL	TL	TL	1.82	TL	TL	21.92	47.97	TL	YES	TL	TL
26	7	[2 2 2 2 1 2 2]	TL	163.13	162.77	TL	14.13	51.84	TL	4.82	49.23	TL	YES	TL	TL
27	7	[2 2 3 2 2 2 1]	10.80	2.10	1.12	16.50	0.70	20.20	4.11	1.32	3.96	29.07	YES	2.25	YES
28	7	[3 2 2 2 2 2 2]	TL	TL	TL	TL	232.56	TL	TL	TL	TL	TL	YES	TL	TL
29	7	[2 3 3 1 1 1 2]	TL	251.41	250.92	64.24	687.38	TL	22.52	TL	26.55	2.11	YES	15.83	YES
30	7	[3 2 2 1 2 1 1]	9.32	3.63	2.71	8.17	1.37	4.62	TL	3.86	TL	2.77	YES	TL	TL
31	7	[1 2 3 1 3 2 1]	TL	TL	419.48	TL	2.80	8.88	TL	3.55	1403.19	TL	YES	TL	TL
32	7	[2 3 2 2 1 2 2]	TL	TL	TL	TL	1687.53	TL	TL	TL	TL	TL	YES	TL	TL
33	7	[2 1 2 2 2 1 2]	TL	2.46	2.46	2.45	2.46	2.46	2.43	2.44	2.47	2.43	YES	TL	TL
34	7	[3 3 1 1 3 3 3]	TL	0.38	0.39	0.38	0.39	0.38	0.38	0.38	0.38	0.38	YES	TL	TL
35	7	[2 2 1 1 3 2 3]	TL	TL	TL	TL	458.72	TL	TL	TL	TL	TL	YES	TL	TL
36	7	[2 2 2 3 3 2 1]	TL	TL	TL	TL	20.15	TL	TL	TL	TL	TL	YES	TL	TL
37	7	[2 1 2 2 1 2 2]	194.28	0.27	0.28	0.28	0.29	0.28	0.27	0.28	0.28	0.28	YES	2.49	YES
38	7	[1 3 2 2 1 2 1]	TL	1.32	1.32	1.33	1.32	1.33	1.33	1.32	1.33	1.33	YES	TL	TL
39	7	[1 2 2 1 1 3 1]	572.42	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	YES	TL	TL
40	7	[2 3 2 1 3 2 2]	TL	6.31	6.36	6.31	6.34	6.32	6.30	6.36	6.32	6.31	YES	TL	TL
41	7	[1 1 1 2 1 2 1]	62.55	2.79	1.92	103.28	1.61	6.88	281.18	8.29	23.78	TL	YES	TL	TL
42	7	[1 2 3 3 2 1 3]	TL	4.96	4.99	4.94	4.98	5.00	4.95	4.95	4.94	4.97	YES	TL	TL
43	7	[2 2 2 2 2 2 1]	TL	9.47	9.58	9.50	9.48	9.59	9.51	9.53	9.55	9.56	YES	TL	TL
44	7	[3 1 1 2 2 2 2]	328.72	23.73	22.80	151.03	36.63	36.91	1190.55	11.59	168.20	4.98	YES	TL	TL
45	7	[2 2 2 3 1 2 2]	62.39	0.12	0.13	0.12	0.12	0.12	0.12	0.13	0.13	0.12	YES	TL	TL
46	7	[2 1 2 2 2 3 2]	TL	TL	TL	TL	357.43	TL	TL	TL	TL	TL	YES	TL	TL
47	7	[2 2 2 1 1 3 2]	1131.25	1.67	1.69	1.66	1.67	1.66	1.66	1.69	1.66	1.66	YES	TL	TL
48	7	[1 2 2 2 2 3 2]	72.68	19.09	18.06	18.08	130.47	136.21	136.37	47.97	173.22	113.87	YES	TL	TL
49	7	[2 2 2 1 2 3 3]	113.30	TL	TL	TL	4.85	TL	115.96	413.01	1065.67	689.10	YES	TL	TL

Table B.4: Instances' solutions for *InstanceSetInsights*. The columns are, in order of appearance: the instance's number, the boolean tax switch (T^a) and the trade switch (T^r). Then, the set of results associated with each of the two countries (*Country One*, and *Country Two*). In particular: the unit-energy production level *Prod*, the domestic price per unit-energy $\$(E)$, the import *Imp* and export *Exp* unit-energies, the export price $\$(E)$, and the tax per unit-emission *Tax*. Furthermore, for each of the the 3 followers of each country, we have the type *Ty* (*C* for coal, *G* for gas, or *S* for solar), the associated emission cost per unit-energy *E*, and its production *Prod*.

			Country One												Country Two															
						Follower 1			Follower 2			Follower 3						Follower 1			Follower 2			Follower 3						
#	T ^a	T ^r	Prod \$ (D)	Imp	Exp	\$ (E)	Tax	Ty	E	Prod	T	E	Prod	Ty	E	Prod	Prod \$ (D)	Imp	Exp	\$ (E)	Tax	Ty	E	Prod	T	E	Prod	Ty	E	Prod
I_0	0	0	83,33 300,00	0,00	0,00	-	80,00	C	500	0,00	G	200	41,67	S	25	41,67	60,00 255,00	0,00	0,00	-	0,30	C	500	27,76	G	100	27,76	S	50	4,48
I_0	0	1	98,81 300,00	0,56	16,03	279,43	80,00	C	500	0,00	G	200	57,14	S	25	41,67	44,52 255,00	16,03	0,56	278,43	9,35	C	500	20,52	G	100	20,52	S	50	3,49
I_0	1	0	83,33 300,00	0,00	0,00	-	0,14	C	500	7,35	G	200	37,22	S	25	38,76	60,00 255,00	0,00	0,00	-	0,00	C	500	27,42	G	100	27,88	S	50	4,69
I_0	1	1	119,32 300,00	0,00	35,99	103255,86	0,07	C	500	32,02	G	200	47,10	S	25	40,20	24,01 255,00	35,99	0,00	103256,86	0,07	C	500	0,14	G	100	22,43	S	50	1,45
I_1	0	0	37,50 270,00	0,00	0,00	-	11,41	C	300	29,69	G	100	7,81	S	50	0,00	50,00 315,00	0,00	0,00	-	15,77	G	100	26,92	S	50	11,54	S	25	11,54
I_1	0	1	0,00 270,00	46,59	9,09	63,54	50,00	C	300	0,00	G	100	0,00	S	50	0,00	87,50 315,00	9,09	46,59	62,54	7,27	G	100	36,36	S	50	25,57	S	25	25,57
I_1	1	0	37,50 270,00	0,00	0,00	-	0,06	C	300	24,73	G	100	12,77	S	50	0,00	50,00 315,00	0,00	0,00	-	0,28	G	100	13,56	S	50	13,88	S	25	22,56
I_1	1	1	3,83 270,00	33,67	0,00	5498,03	0,16	C	300	0,57	G	100	3,26	S	50	0,00	83,67 315,00	0,00	33,67	5497,03	0,11	G	100	31,82	S	50	24,15	S	25	27,70
I_2	0	0	30,56 247,50	0,00	0,00	-	15,56	C	300	22,03	G	100	8,53	S	25	0,00	97,50 276,25	0,00	0,00	-	20,04	C	500	53,53	G	200	36,21	S	50	7,76
I_2	0	1	52,40 247,50	30,03	51,88	304,10	0,00	C	300	32,76	G	100	19,64	S	25	0,00	75,65 276,25	51,88	30,03	303,10	40,57	C	500	33,98	G	200	26,32	S	50	15,35
I_2	1	0	30,56 247,50	0,00	0,00	-	0,08	C	300	16,52	G	100	14,04	S	25	0,00	97,50 276,25	0,00	0,00	-	0,09	C	500	31,24	G	200	38,87	S	50	27,38
I_2	1	1	52,40 247,50	0,00	21,85	161540,06	0,00	C	300	32,76	G	100	19,64	S	25	0,00	75,65 276,25	21,85	0,00	161541,06	0,12	C	500	17,16	G	200	32,96	S	50	25,53
I_3	0	0	84,37 382,50	0,00	0,00	-	116,88	C	500	35,09	G	200	35,09	S	25	14,20	30,56 247,50	0,00	0,00	-	32,73	S	25	10,19	S	50	10,19	S	50	10,19
I_3	0	1	16,65 382,50	79,62	11,90	178,85	153,34	C	500	2,82	G	200	7,05	S	25	6,79	98,28 247,50	11,90	79,62	179,85	0,00	S	25	32,76	S	50	32,76	S	50	32,76
I_3	1	0	134,71 342,23	0,00	0,00	-	0,24	C	500	0,00	G	200	56,42	S	25	78,29	54,22 226,20	0,00	0,00	-	0,00	S	25	18,07	S	50	18,07	S	50	18,07
I_3	1	1	178,28 374,04	14,96	98,30	64,87	0,31	C	500	0,00	G	200	71,10	S	25	107,18	0,00 200,00	98,30	14,96	65,87	0,00	S	25	0,00	S	50	0,00	S	50	0,00
I_4	0	0	108,43 277,41	0,00	0,00	-	25,00	S	25	36,14	S	50	36,14	S	50	36,14	93,75 318,75	0,00	0,00	-	0,29	G	200	11,68	S	50	36,49	S	25	45,59
I_4	0	1	151,64 298,29	0,00	66,41	27,11	25,00	S	25	50,55	S	50	50,55	S	50	50,55	27,34 318,75	66,41	0,00	28,11	0,88	G	200	0,00	S	50	0,00	S	25	27,34
I_4	1	0	93,55 290,80	0,00	0,00	-	1,09	S	25	43,76	S	50	24,90	S	50	24,90	93,75 318,75	0,00	0,00	-	0,29	G	200	11,68	S	50	36,49	S	25	45,59
I_4	1	1	101,42 298,44	0,00	16,36	159,09	1,19	S	25	47,44	S	50	26,99	S	50	26,99	77,39 318,75	16,36	0,00	160,09	0,34	G	200	0,17	S	50	33,25	S	25	43,97
I_5	0	0	112,50 318,75	0,00	0,00	-	110,96	C	300	52,54	G	200	52,54	S	25	7,42	80,00 240,00	0,00	0,00	-	300,00	C	500	0,00	G	200	49,23	G	200	30,77
I_5	0	1	192,50 318,75	60,93	140,93	188,86	91,39	C	300	70,33	G	200	70,33	S	25	51,85	0,00 240,00	140,93	60,93	189,86	90,00	C	500	0,00	G	200	0,00	G	200	0,00
I_5	1	0	136,64 306,68	0,00	0,00	-	0,52	C	300	0,00	G	200	47,48	S	25	89,16	80,00 240,00	0,00	0,00	-	0,13	C	500	20,34	G	200	48,14	G	200	11,53
I_5	1	1	150,84 318,75	0,00	38,34	116,22	0,56	C	300	0,00	G	200	51,14	S	25	99,70	41,66 240,00	38,34	0,00	117,22	0,18	C	500	0,00	G	200	39,31	G	200	2,36
I_6	0	0	56,25 255,00	0,00	0,00	-	0,00	C	500	25,30	G	100	26,60	S	50	4,35	81,25 276,25	0,00	0,00	-	9,01	G	200	42,94	S	50	19,15	S	50	19,15
I_6	0	1	28,03 255,00	28,22	0,00	398,97	0,06	C	500	3,89	G	100	22,32	S	50	1,82	109,47 276,25	0,00	28,22	397,97	0,00	G	200	51,14	S	50	29,17	S	50	29,17
I_6	1	0	56,25 255,00	0,00	0,00	-	0,00	C	500	25,30	G	100	26,60	S	50	4,35	81,25 276,25	0,00	0,00	-	0,10	G	200	33,62	S	50	23,81	S	50	23,81
I_6	1	1	28,03 255,00	28,22	0,00	190054,14	0,06	C	500	3,89	G	100	22,32	S	50	1,82	109,47 276,25	0,00	28,22	190053,14	0,00	G	200	51,14	S	50	29,17	S	50	29,17
I_7	0	0	58,33 297,50	0,00	0,00	-	103,75	C	300	29,17	G	100	29,17	S	25	0,00	56,52 399,13	0,00	0,00	-	199,13	C	500	0,00	G	100	0,00	G	200	56,52
I_7	0	1	0,00 297,50	58,33	0,00	99,00	147,50	C	300	0,00	G	100	0,00	S	25	0,00	108,33 405,00	0,00	58,33	100,00	205,00	C	500	0,00	G	100	47,62	G	200	60,71
I_7	1	0	91,94 267,25	0,00	0,00	-	0,39	C	300	0,13	G	100	52,16	S	25	39,65	122,61 339,65	0,00	0,00	-	0,28	C	500	0,00	G	100	77,05	G	200	45,56
I_7	1	1	124,56 297,50	0,00	66,22	114,18	0,49	C	300	0,17	G	100	65,61	S	25	58,78	91,57 307,98	66,22	0,00	115,18	0,22	C	500	0,00	G	100	59,58	G	200	31,99
I_8	0	0	150,00 360,00	0,00	0,00	-	2,46	S	50	32,17	S	25	85,65	S	50	32,17	60,94 276,25	0,00	0,00	-	58,91	C	500	48,10	G	200	12,84	S	50	0,00
I_8	0	1	210,94 360,00	0,00	60,94	45,00	1,90	S	50	56,55	S	25	97,84	S	50	56,55	0,00 276,25	60,94	0,00	46,00	126,25	C	500	0,00	G	200	0,00	S	50	0,00
I_8	1	0	150,00 360,00	0,00	0,00	-	2,46	S	50	32,17	S	25	85,65	S	50	32,17	60,94 276,25	0,00	0,00	-	0,24	C	500	5,46	G	200	21,34	S	50	34,15
I_8	1	1	157,50 360,00	0,00	7,50	1045,00	2,39	S	50	35,18	S	25	87,15	S	50	35,18	53,43 276,25	7,50	0,00	1046,00	0,25	C	500	0,53	G	200	19,29	S	50	33,61
I_9	0	0	150,00 360,00	0,00	0,00	-	2,30	C	300	0,00	S	25	75,00	S	25	75,00	93,33 280,00	0,00	0,00	-	0,47	G	100	25,47	S	25	38,62	S	50	29,24
I_9	0	1	219,33 360,00	0,00	69,33	25,00	0,77	C	300	0,00	S	25	109,67	S	25	109,67	24,00 280,00	69,33	0,00	26,00	1,20	G	100	0,00	S	25	24,00	S	50	0,00
I_9	1	0	190,15 335,91	0,00	0,00	-	0,45	C	300	0,00	S	25	95,08	S	25	95,08	100,45 274,66	0,00	0,00	-	0,32	G	100	32,46	S	25	37,24	S	50	30,75

Table B.4: *Continued.* Instances' solutions for *InstanceSetInsights*

I_9	1	1	230,47	360,00	9,15	89,62	3239,81	0,53	C 300	0,14	S 25	115,16	S 25	115,16	60,66	244,15	89,62	9,15	3238,81	0,19	G 100	19,20	S 25	22,65	S 50	18,81
I_10	0	0	69,57	283,26	0,00	0,00	-	83,26	G 100	0,00	S 25	34,78	S 50	34,78	37,50	270,00	0,00	0,00	-	11,41	C 300	29,69	G 100	7,81	S 50	0,00
I_10	1	0	91,67	292,50	0,00	37,50	25,00	92,50	G 100	0,00	S 25	48,48	S 50	43,18	-0,00	270,00	37,50	0,00	26,00	50,00	C 300	-0,00	G 100	0,00	S 50	0,00
I_10	1	0	67,09	284,75	0,00	0,00	-	0,80	G 100	4,01	S 25	40,65	S 50	22,43	37,50	270,00	0,00	0,00	-	0,06	C 300	24,73	G 100	12,77	S 50	0,00
I_10	1	1	87,84	292,50	0,00	33,67	11284,55	0,80	G 100	10,70	S 25	47,68	S 50	29,46	3,83	270,00	33,67	0,00	11285,55	0,16	C 300	0,57	G 100	3,26	S 50	0,00
I_11	0	0	125,76	286,97	0,00	0,00	-	25,00	S 50	34,97	S 50	34,97	S 25	55,81	75,00	337,50	0,00	0,00	-	0,58	C 500	0,00	G 200	1,79	S 50	73,21
I_11	0	1	173,48	306,06	0,00	75,00	26,52	25,00	S 50	50,88	S 50	50,88	S 25	71,72	-0,00	337,50	75,00	0,00	27,52	1,75	C 500	0,00	G 200	0,00	S 50	-0,00
I_11	1	0	138,74	277,88	0,00	0,00	-	0,06	S 50	45,85	S 50	45,85	S 25	47,04	123,42	313,29	0,00	0,00	-	0,19	C 500	0,00	G 200	55,97	S 50	67,45
I_11	1	1	82,23	254,31	90,19	0,00	284,84	0,03	S 50	27,17	S 50	27,17	S 25	27,88	165,19	337,50	0,00	90,19	283,84	0,24	C 500	0,00	G 200	70,50	S 50	94,69
I_12	0	0	97,50	276,25	0,00	0,00	-	0,06	C 500	28,12	G 100	50,63	G 200	18,75	150,00	300,00	0,00	0,00	-	8,57	C 300	51,79	G 100	62,50	S 25	35,71
I_12	0	1	86,79	276,25	10,71	0,00	364,71	0,07	C 500	21,82	G 100	49,36	G 200	15,60	160,71	300,00	0,00	10,71	363,71	0,00	C 300	62,50	G 100	62,50	S 25	35,71
I_12	1	0	97,50	276,25	0,00	0,00	-	0,06	C 500	28,12	G 100	50,63	G 200	18,75	150,00	300,00	0,00	0,00	-	0,02	C 300	55,00	G 100	60,00	S 25	35,00
I_12	1	1	86,79	276,25	10,71	0,00	88598,00	0,07	C 500	21,82	G 100	49,36	G 200	15,60	160,71	300,00	0,00	10,71	88599,00	0,00	C 300	62,50	G 100	62,50	S 25	35,71
I_13	0	0	135,00	382,50	0,00	0,00	-	0,72	C 300	16,22	G 200	81,27	G 200	37,52	38,89	315,00	0,00	0,00	-	135,83	C 500	19,44	G 200	19,44	G 100	0,00
I_13	0	1	173,89	382,50	31,08	69,97	231,74	0,63	C 300	24,66	G 200	95,03	G 200	54,20	0,00	315,00	69,97	31,08	232,74	165,00	C 500	0,00	G 200	0,00	G 100	0,00
I_13	1	0	135,00	382,50	0,00	0,00	-	0,72	C 300	16,22	G 200	81,27	G 200	37,52	83,95	274,44	0,00	0,00	-	0,25	C 500	0,00	G 200	49,78	G 100	34,18
I_13	1	1	96,65	382,50	71,27	32,92	169,07	0,78	C 300	0,00	G 200	70,45	G 200	26,19	104,40	290,56	32,92	71,27	168,07	0,28	C 500	0,00	G 200	56,22	G 100	48,18
I_14	0	0	90,00	382,50	0,00	0,00	-	2,87	S 50	30,00	S 50	30,00	S 50	30,00	60,00	255,00	0,00	0,00	-	6,57	C 500	37,25	G 100	22,75	S 50	0,00
I_14	0	1	150,00	382,50	0,00	60,00	50,00	2,35	S 50	50,00	S 50	50,00	S 50	50,00	0,00	255,00	60,00	0,00	51,00	55,00	C 500	0,00	G 100	0,00	S 50	0,00
I_14	1	0	207,61	294,29	0,00	0,00	-	0,09	S 50	69,20	S 50	69,20	S 50	69,20	60,00	255,00	0,00	0,00	-	0,03	C 500	30,99	G 100	25,65	S 50	3,36
I_14	1	1	233,02	305,83	0,00	40,80	290,66	0,10	S 50	77,67	S 50	77,67	S 50	77,67	19,20	255,00	40,80	0,00	291,66	0,11	C 500	0,00	G 100	19,20	S 50	0,00
I_15	0	0	60,00	405,00	0,00	0,00	-	118,16	C 300	35,08	S 50	12,46	S 50	12,46	40,00	268,00	0,00	0,00	-	50,00	S 50	13,33	S 50	13,33	S 50	13,33
I_15	0	1	0,00	405,00	60,00	0,00	51,00	155,00	C 300	0,00	S 50	0,00	S 50	0,00	97,50	270,00	0,00	60,00	50,00	26,13	S 50	32,50	S 50	32,50	S 50	32,50
I_15	1	0	128,79	353,41	0,00	0,00	-	0,34	C 300	0,00	S 50	64,39	S 50	64,39	77,42	238,06	0,00	0,00	-	0,06	S 50	25,81	S 50	25,81	S 50	25,81
I_15	1	1	201,64	392,52	21,47	146,47	60,88	0,48	C 300	0,00	S 50	100,82	S 50	100,82	0,00	200,00	146,47	21,47	61,88	0,00	S 50	0,00	S 50	0,00	S 50	0,00
I_16	0	0	60,00	270,00	0,00	0,00	-	0,10	S 50	19,00	S 50	19,00	S 25	22,00	128,57	360,00	0,00	0,00	-	110,00	C 500	0,00	G 100	34,13	S 50	94,44
I_16	0	1	75,00	270,00	0,00	15,00	99,00	0,00	S 50	25,00	S 50	25,00	S 25	25,00	113,57	360,00	15,00	0,00	98,00	110,00	C 500	0,00	G 100	19,13	S 50	94,44
I_16	1	0	60,00	270,00	0,00	0,00	-	0,10	S 50	19,00	S 50	19,00	S 25	22,00	147,24	346,93	0,00	0,00	-	0,19	C 500	0,41	G 100	77,63	S 50	69,20
I_16	1	1	17,75	270,00	42,25	0,00	543,00	0,37	S 50	2,10	S 50	2,10	S 25	13,55	170,83	360,00	0,00	42,25	544,00	0,22	C 500	0,46	G 100	88,09	S 50	82,27
I_17	0	0	112,50	318,75	0,00	0,00	-	9,63	C 500	48,75	S 25	31,88	S 25	31,88	43,75	315,00	0,00	0,00	-	0,38	C 300	0,53	G 200	28,75	G 200	14,47
I_17	0	1	113,75	318,75	31,68	32,93	211,08	12,32	C 500	44,90	S 25	34,43	S 25	34,43	42,50	315,00	32,93	31,68	212,08	0,38	C 300	0,00	G 200	28,40	G 200	14,10
I_17	1	0	112,50	318,75	0,00	0,00	-	0,06	C 500	21,46	S 25	45,52	S 25	45,52	43,75	315,00	0,00	0,00	-	0,38	C 300	0,53	G 200	28,75	G 200	14,47
I_17	1	1	88,87	318,75	23,63	0,00	38393,57	0,09	C 500	0,29	S 25	44,29	S 25	44,29	67,38	315,00	0,00	23,63	38394,57	0,34	C 300	10,55	G 200	35,43	G 200	21,41
I_18	0	0	50,00	337,50	0,00	0,00	-	1,03	G 100	0,00	S 50	11,40	S 25	38,60	150,00	360,00	0,00	0,00	-	85,00	C 300	0,00	G 200	106,25	G 200	43,75
I_18	0	1	200,00	337,50	0,00	150,00	73,56	0,09	G 100	75,18	S 50	61,28	S 25	63,54	0,00	360,00	150,00	0,00	72,56	85,00	C 300	0,00	G 200	0,00	G 200	0,00
I_18	1	0	84,10	311,93	0,00	0,00	-	0,30	G 100	30,21	S 50	22,97	S 25	30,92	150,00	360,00	0,00	0,00	-	0,18	C 300	39,34	G 200	61,64	G 200	49,02
I_18	1	1	140,56	337,50	0,00	90,56	6225,32	0,43	G 100	42,68	S 50	43,32	S 25	54,56	79,91	347,71	90,56	0,00	6226,32	0,21	C 300	13,72	G 200	39,44	G 200	26,75
I_19	0	0	64,29	405,00	0,00	0,00	-	2,21	S 50	21,43	S 50	21,43	S 50	21,43	60,00	270,00	0,00	0,00	-	0,26	C 500	0,00	G 200	16,57	S 25	43,43
I_19	0	1	83,04	405,00	0,00	18,75	50,00	2,10	S 50	27,68	S 50	27,68	S 50	27,68	41,25	270,00	18,75	0,00	51,00	0,35	C 500	0,00	G 200	0,00	S 25	41,25
I_19	1	0	64,29	405,00	0,00	0,00	-	2,21	S 50	21,43	S 50	21,43	S 50	21,43	78,74	260,63	0,00	0,00	-	0,11	C 500	4,47	G 200	36,43	S 25	37,83
I_19	1	1	150,00	405,00	16,61	102,32	5570,48	1,70	S 50	50,00	S 50	50,00	S 50	50,00	45,91	234,19	102,32	16,61	5569,48	0,06	C 500	2,52	G 200	20,54	S 25	22,84
I_20	0	0	48,65	331,22	0,00	0,00	-	25,00	S 25	16,22	S 25	16,22	S 50	16,22	107,14	300,00	0,00	0,00	-	44,76	C 500	29,37	G 100	50,00	S 50	27,78
I_20	0	1	142,50	337,50	0,00	100,83	49,00	0,00	S 25	47,50	S 25	47,50	S 50	47,50	6,31	300,00	100,83	0,00	48,00	80,00	C 500	0,00	G 100	0,00	S 50	6,31
I_20	1	0	68,14	313,67	0,00	0,00	-	0,03	S 25	22,95	S 25	22,95	S 50	22,24	107,14	300,00	0,00	0,00	-	0,07	C 500	39,51	G 100	43,48	S 50	24,16
I_20	1	1	142,50	337,50	0,00	100,83	499,24	0,00	S 25	47,50	S 25	47,50	S 50	47,50	38,51	277,46	100,83	0,00	500,24	0,11	C 500	0,38	G 100	24,51	S 50	13,62
I_21	0	0	43,75	315,00	0,00	0,00	-	0,98	S 50	14,58	S 50	14,58	S 50	14,58	78,57	220,00	0,00	0,00	-	3,07	C 500	51,48	S 50	13,54	S 50	13,54
I_21	0	1	122,32	315,00	0,00	78,57	50,00	0,40	S 50	40,77	S 50	40,77	S 50	40,77	0,00	220,00	78,57	0,00	51,00	70,00	C 500	0,00	S 50	0,00	S 50	0,00
I_21	1	0	84,21	282,63	0,00	0,00	-	0,04	S 50	28,07	S 50	28,07	S 50	28,07	78,57	220,00	0,00	0,00	-	0,02	C 500	47,82	S 50	15,37	S 50	15,37
I_21	1	1	177,27	315,00	0,00	133,52	583,63	0,00	S 50	59,09	S 50	59,09	S 50	59,09	7,86	176,03	133,52	0,00	584,63	0,05	C 500	0,10	S 50	3,88	S 50	3,88
I_22	0	0	36,67	247,50	0,00	0,00	-	13,94	C 300	25,82																

Table B.4: *Continued.* Instances' solutions for *InstanceSetInsights*

I_22	1	0	36,67	247,50	0,00	0,00	-	0,06	C 300	23,62	G 200	13,04	S 50	0,00	100,46	314,73	0,00	0,00	-	0,23	C 500	0,46	S 50	50,00	S 25	50,00
I_22	1	1	0,32	246,22	38,05	0,00	777,91	0,15	C 300	0,32	G 200	0,00	S 50	0,00	100,55	337,50	0,00	38,05	776,91	0,27	C 500	0,55	S 50	50,00	S 25	50,00
I_23	0	0	100,00	300,00	0,00	0,00	-	38,33	S 25	33,33	S 25	33,33	S 25	33,33	70,00	315,00	0,00	0,00	-	19,00	G 100	10,00	S 50	30,00	S 50	30,00
I_23	0	1	170,00	300,00	0,00	70,00	25,00	9,17	S 25	56,67	S 25	56,67	S 25	56,67	0,00	315,00	70,00	0,00	24,00	25,00	G 100	0,00	S 50	0,00	S 50	0,00
I_23	1	0	100,00	300,00	0,00	0,00	-	1,53	S 25	33,33	S 25	33,33	S 25	33,33	70,00	315,00	0,00	0,00	-	0,09	G 100	27,27	S 50	21,36	S 50	21,36
I_23	1	1	158,91	300,00	0,00	58,91	441,67	0,55	S 25	52,97	S 25	52,97	S 25	52,97	11,09	315,00	58,91	0,00	442,67	0,25	G 100	0,49	S 50	5,30	S 50	5,30
I_24	0	0	120,00	360,00	0,00	0,00	-	0,73	C 500	0,00	G 100	66,67	G 100	53,33	70,00	297,50	0,00	0,00	-	38,88	C 500	30,90	G 100	30,90	G 100	8,21
I_24	0	1	184,86	360,00	2,85	67,71	227,31	0,32	C 500	0,00	G 100	98,46	G 100	86,40	5,14	297,50	67,71	2,85	226,31	74,97	C 500	1,63	G 100	2,03	G 100	1,48
I_24	1	0	153,41	334,94	0,00	0,00	-	0,27	C 500	0,00	G 100	83,04	G 100	70,36	74,91	293,82	0,00	0,00	-	0,15	C 500	0,00	G 100	47,24	G 100	27,67
I_24	1	1	148,52	331,05	10,08	0,00	85,12	0,26	C 500	0,00	G 100	80,65	G 100	67,87	80,08	297,50	0,00	10,08	86,12	0,16	C 500	0,00	G 100	49,60	G 100	30,48
I_25	0	0	70,00	315,00	0,00	0,00	-	95,00	C 500	0,00	G 200	0,00	S 50	70,00	42,86	270,00	0,00	0,00	-	39,11	G 200	24,71	S 25	9,07	S 25	9,07
I_25	0	1	81,25	315,00	21,45	32,70	90,98	95,00	C 500	0,00	G 200	0,00	S 50	81,25	31,61	270,00	32,70	21,45	89,98	53,41	G 200	11,85	S 25	9,88	S 25	9,88
I_25	1	0	70,00	315,00	0,00	0,00	-	0,40	C 500	0,00	G 200	14,05	S 50	55,95	55,67	261,03	0,00	0,00	-	0,31	G 200	0,00	S 25	27,84	S 25	27,84
I_25	1	1	49,97	312,07	25,89	0,00	57,83	0,46	C 500	0,00	G 200	0,00	S 50	49,97	68,75	270,00	0,00	25,89	58,83	0,35	G 200	0,00	S 25	34,38	S 25	34,38
I_26	0	0	135,00	382,50	0,00	0,00	-	0,74	C 500	0,00	G 200	15,36	S 50	119,64	90,00	382,50	0,00	0,00	-	70,71	C 500	38,73	S 25	25,64	S 50	25,64
I_26	0	1	114,84	382,50	73,11	52,96	124,23	0,81	C 500	0,00	G 200	0,00	S 50	114,84	110,16	382,50	52,96	73,11	123,23	63,26	C 500	20,58	S 25	44,79	S 50	44,79
I_26	1	0	192,89	353,56	0,00	0,00	-	0,27	C 500	0,00	G 200	80,13	S 50	112,75	131,28	351,54	0,00	0,00	-	0,15	C 500	0,00	S 25	67,89	S 50	63,39
I_26	1	1	150,24	320,54	175,38	66,71	102,76	0,20	C 500	0,00	G 200	60,33	S 50	89,91	198,68	382,50	66,71	175,38	103,76	0,22	C 500	0,00	S 25	102,50	S 50	96,18
I_27	0	0	115,22	305,87	0,00	0,00	-	50,00	S 50	32,61	S 50	32,61	S 25	50,00	75,00	337,50	0,00	0,00	-	82,88	C 300	34,62	G 200	34,62	G 200	5,77
I_27	0	1	142,86	334,29	8,33	83,33	22,48	2,38	S 50	47,62	S 50	47,62	S 25	47,62	0,00	337,50	83,33	8,33	23,48	117,50	C 300	0,00	G 200	0,00	G 200	0,00
I_27	1	0	62,50	337,50	0,00	0,00	-	2,21	S 50	6,25	S 50	6,25	S 25	50,00	93,12	328,44	0,00	0,00	-	0,30	C 300	19,72	G 200	49,29	G 200	24,11
I_27	1	1	150,00	337,50	3,53	91,03	5779,98	1,25	S 50	50,00	S 50	50,00	S 25	50,00	67,10	297,70	91,03	3,53	5778,98	0,21	C 300	14,13	G 200	35,32	G 200	17,66
I_28	0	0	80,00	240,00	0,00	0,00	-	14,47	C 300	55,94	G 200	19,63	S 25	4,42	150,00	360,00	0,00	0,00	-	0,47	G 200	0,00	S 25	83,33	S 50	66,67
I_28	0	1	0,00	240,00	80,00	0,00	153,50	90,00	C 300	0,00	G 200	0,00	S 25	0,00	230,00	360,00	0,00	80,00	152,50	0,21	G 200	52,88	S 25	92,38	S 50	84,75
I_28	1	0	80,00	240,00	0,00	0,00	-	0,08	C 300	47,91	G 200	17,78	S 25	14,31	152,08	358,75	0,00	0,00	-	0,42	G 200	0,35	S 25	83,31	S 50	68,41
I_28	1	1	26,77	240,00	53,23	0,00	28153,50	0,23	C 300	15,38	G 200	0,00	S 25	11,38	203,23	360,00	0,00	53,23	28152,50	0,29	G 200	34,14	S 25	89,70	S 50	79,40
I_29	0	0	75,00	382,50	0,00	0,00	-	4,40	S 25	37,50	S 50	0,00	S 25	37,50	40,62	292,50	0,00	0,00	-	0,16	C 500	0,00	G 200	31,12	G 200	9,51
I_29	0	1	115,63	382,50	0,00	40,62	25,00	3,26	S 25	57,81	S 50	0,00	S 25	57,81	0,00	292,50	40,62	0,00	26,00	0,36	C 500	0,00	G 200	0,00	G 200	0,00
I_29	1	0	75,00	382,50	0,00	0,00	-	4,40	S 25	37,50	S 50	0,00	S 25	37,50	40,62	292,50	0,00	0,00	-	0,16	C 500	0,00	G 200	31,12	G 200	9,51
I_29	1	1	69,86	382,50	33,88	28,75	425,88	4,89	S 25	28,83	S 50	12,21	S 25	28,83	45,76	292,50	28,75	33,88	426,88	0,14	C 500	0,01	G 200	33,47	G 200	12,28
I_30	0	0	112,50	360,00	0,00	0,00	-	54,14	C 300	50,78	S 50	30,86	S 50	30,86	36,11	292,50	0,00	0,00	-	0,00	S 50	15,91	S 50	4,29	S 25	15,91
I_30	0	1	100,88	360,00	70,47	58,85	141,76	68,90	C 300	37,36	S 50	31,76	S 50	31,76	47,73	292,50	58,85	70,47	142,76	0,00	S 50	15,91	S 50	15,91	S 25	15,91
I_30	1	0	121,51	352,79	0,00	0,00	-	0,34	C 300	0,14	S 50	60,69	S 50	60,69	36,11	292,50	0,00	0,00	-	0,10	S 50	11,26	S 50	11,26	S 25	13,59
I_30	1	1	133,53	360,00	0,00	21,03	227,15	0,37	C 300	0,15	S 50	66,69	S 50	66,69	20,09	287,99	21,03	0,00	226,15	0,14	S 50	5,67	S 50	5,67	S 25	8,74
I_31	0	0	46,67	315,00	0,00	0,00	-	56,62	C 300	30,70	S 25	7,98	S 25	7,98	75,00	382,50	0,00	0,00	-	162,50	C 300	0,00	G 100	0,00	S 50	75,00
I_31	0	1	-0,00	315,00	46,67	0,00	101,00	95,00	C 300	-0,00	S 25	0,00	S 25	0,00	121,67	382,50	0,00	46,67	100,00	162,50	C 300	0,00	G 100	23,94	S 50	97,73
I_31	1	0	73,59	294,81	0,00	0,00	-	0,25	C 300	0,10	S 25	36,75	S 25	36,75	104,27	356,15	0,00	0,00	-	0,45	C 300	0,30	G 100	50,76	S 50	53,21
I_31	1	1	108,88	315,00	38,70	100,92	133,07	0,32	C 300	0,13	S 25	54,38	S 25	54,38	84,09	318,33	100,92	38,70	132,07	0,33	C 300	0,22	G 100	39,58	S 50	44,28
I_32	0	0	86,67	260,00	0,00	0,00	-	0,08	C 300	27,51	S 50	28,77	S 25	30,38	65,00	276,25	0,00	0,00	-	3,50	S 25	21,67	S 50	21,67	S 50	21,67
I_32	0	1	76,67	260,00	10,00	0,00	246,69	0,12	C 300	19,58	S 50	27,39	S 25	29,70	75,00	276,25	0,00	10,00	245,69	0,00	S 25	25,00	S 50	25,00	S 50	25,00
I_32	1	0	86,67	260,00	0,00	0,00	-	0,08	C 300	27,51	S 50	28,77	S 25	30,38	68,18	273,86	0,00	0,00	-	-0,00	S 25	22,73	S 50	22,73	S 50	22,73
I_32	1	1	76,67	260,00	10,00	0,00	71673,26	0,12	C 300	19,58	S 50	27,39	S 25	29,70	75,00	276,25	0,00	10,00	71674,26	0,00	S 25	25,00	S 50	25,00	S 50	25,00
I_33	0	0	75,00	318,75	0,00	0,00	-	0,08	C 300	21,08	G 200	29,40	G 100	24,52	54,17	292,50	0,00	0,00	-	0,09	C 300	17,01	G 100	37,15	G 200	0,00
I_33	0	1	97,69	318,75	0,00	22,69	230,77	0,04	C 300	32,20	G 200	36,82	G 100	28,66	31,48	292,50	22,69	0,00	231,77	0,14	C 300	0,00	G 100	31,48	G 200	0,00
I_33	1	0	75,00	318,75	0,00	0,00	-	0,08	C 300	21,08	G 200	29,40	G 100	24,52	54,17	292,50	0,00	0,00	-	0,09	C 300	17,01	G 100	37,15	G 200	0,00
I_33	1	1	97,06	318,75	0,00	22,06	44365,38	0,04	C 300	31,90	G 200	36,62	G 100	28,55	32,10	292,50	22,06	0,00	44366,38	0,14	C 300	0,47	G 100	31,64	G 200	0,00
I_34	0	0	69,64	276,25	0,00	0,00	-	0,08	C 300	26,46	G 200	33,26	G 200	9,92	97,50	276,25	0,00	0,00	-	0,25	S 25	32,50	S 50	32,50	S 25	32,50
I_34	0	1	68,71	276,25	26,55	25,61	243,88	0,09	C 300	23,84	G 200	31,52	G 200	13,35	98,44	276,25	25,61	26,55	244,88	0,00	S 25	32,81	S 50	32,81	S 25	32,81
I_34	1	0	69,64	276,25	0,00	0,00	-	0,08	C 300	26,46	G 200	33,26	G 200	9,92	97,50	276,25	0,00	0,00	-	0,01	S 25	32,58	S 50	32,34	S 25	32,58
I_34	1	1	68,71	276,25	26,52	25,58	67586,10	0,09	C 300	23,85	G 200	31,52														

Table B.4: *Continued.* Instances' solutions for *InstanceSetInsights*

I_35 0	1	37,50	300,00	69,64	0,00	101,00	0,13	C 500	0,00	G 100	37,50	G 200	0,00	169,64	360,00	0,00	69,64	100,00	0,00	G 100	69,64	S 25	50,00	S 25	50,00
I_35 1	0	107,14	300,00	0,00	0,00	-	0,03	C 500	37,45	G 100	47,49	G 200	22,20	119,53	342,43	0,00	0,00	-	0,17	G 100	34,95	S 25	42,29	S 25	42,29
I_35 1	1	45,57	300,00	61,57	0,00	10101,00	0,10	C 500	0,01	G 100	40,00	G 200	5,56	161,57	360,00	0,00	61,57	10100,00	0,08	G 100	61,57	S 25	50,00	S 25	50,00
I_36 0	0	75,00	337,50	0,00	0,00	-	59,32	C 500	35,23	G 100	35,23	G 100	4,55	70,00	315,00	0,00	0,00	-	0,00	S 25	41,67	S 50	0,00	S 50	28,33
I_36 0	1	20,00	337,50	69,98	14,98	233,43	81,64	C 500	6,70	G 100	7,32	G 100	5,98	125,00	315,00	14,98	69,98	234,43	0,00	S 25	41,67	S 50	41,67	S 50	41,67
I_36 1	0	112,07	318,97	0,00	0,00	-	0,14	C 500	0,00	G 100	68,97	G 100	43,10	85,71	307,14	0,00	0,00	-	0,00	S 25	28,57	S 50	28,57	S 50	28,57
I_36 1	1	151,79	337,50	0,00	76,79	277,33	0,18	C 500	0,00	G 100	87,50	G 100	64,29	9,43	306,89	76,79	0,00	278,33	0,45	S 25	9,43	S 50	0,00	S 50	0,00
I_37 0	0	68,75	220,00	0,00	0,00	-	64,49	C 300	3,94	G 100	50,00	G 200	14,81	135,00	382,50	0,00	0,00	-	98,50	G 100	50,00	S 25	42,50	S 25	42,50
I_37 0	1	0,00	220,00	131,60	62,85	35,13	70,00	C 300	0,00	G 100	0,00	G 200	0,00	203,75	382,50	62,85	131,60	34,13	75,66	G 100	26,72	S 25	88,51	S 25	88,51
I_37 1	0	68,75	220,00	0,00	0,00	-	0,11	C 300	26,56	G 100	42,19	G 200	0,00	183,87	358,06	0,00	0,00	-	1,38	G 100	0,00	S 25	91,94	S 25	91,94
I_37 1	1	16,80	185,28	95,35	0,00	85,52	0,12	C 300	0,00	G 100	16,80	G 200	0,00	230,35	382,50	0,00	95,35	84,52	1,62	G 100	0,41	S 25	114,97	S 25	114,97
I_38 0	0	50,00	337,50	0,00	0,00	-	128,92	C 300	43,40	G 100	6,60	S 50	0,00	50,00	405,00	0,00	0,00	-	99,28	C 300	27,93	G 200	15,72	S 50	6,35
I_38 0	1	100,00	337,50	34,84	84,84	164,29	130,64	C 300	39,42	G 100	32,07	S 50	28,50	0,00	405,00	84,84	34,84	165,29	130,00	C 300	0,00	G 200	0,00	S 50	0,00
I_38 1	0	90,69	306,98	0,00	0,00	-	0,52	C 300	0,00	G 100	42,04	S 50	48,65	71,79	385,38	0,00	0,00	-	0,37	C 300	0,00	G 200	21,79	S 50	50,00
I_38 1	1	126,69	337,50	0,00	76,69	88,84	0,63	C 300	0,00	G 100	57,69	S 50	69,00	42,50	342,72	76,69	0,00	89,84	0,23	C 300	0,00	G 200	7,57	S 50	34,93
I_39 0	0	50,00	337,50	0,00	0,00	-	128,92	C 300	43,40	G 100	6,60	S 50	0,00	69,64	276,25	0,00	0,00	-	47,23	S 50	23,21	S 50	23,21	S 50	23,21
I_39 0	1	0,00	337,50	50,00	0,00	51,00	187,50	C 300	0,00	G 100	0,00	S 50	0,00	119,64	276,25	0,00	50,00	50,00	26,40	S 50	39,88	S 50	39,88	S 50	39,88
I_39 1	0	90,69	306,98	0,00	0,00	-	0,52	C 300	0,00	G 100	42,04	S 50	48,65	69,64	276,25	0,00	0,00	-	0,94	S 50	23,21	S 50	23,21	S 50	23,21
I_39 1	1	120,56	328,43	14,51	72,98	87,59	0,59	C 300	0,00	G 100	54,90	S 50	65,66	11,17	276,25	72,98	14,51	88,59	1,43	S 50	3,72	S 50	3,72	S 50	3,72
I_40 0	0	72,22	260,00	0,00	0,00	-	38,13	G 200	15,08	S 25	28,57	S 50	28,57	60,00	255,00	0,00	0,00	-	0,87	S 25	25,58	S 25	25,58	S 50	8,85
I_40 0	1	28,57	260,00	43,65	0,00	36,50	60,00	G 200	0,00	S 25	28,57	S 50	-0,00	103,65	255,00	0,00	43,65	37,50	0,30	S 25	36,49	S 25	36,49	S 50	30,67
I_40 1	0	73,34	259,00	0,00	0,00	-	0,12	G 200	24,08	S 25	25,70	S 50	23,55	60,00	255,00	0,00	0,00	-	0,87	S 25	25,58	S 25	25,58	S 50	8,85
I_40 1	1	38,12	230,46	68,37	1,45	3501,39	0,06	G 200	12,43	S 25	13,40	S 50	12,29	126,92	255,00	1,45	68,37	3502,39	0,00	S 25	42,31	S 25	42,31	S 50	42,31
I_41 0	0	112,50	318,75	0,00	0,00	-	0,55	C 300	2,10	G 100	102,97	G 200	7,42	120,00	360,00	0,00	0,00	-	85,00	G 200	0,00	S 50	82,35	S 50	37,65
I_41 0	1	67,79	318,75	44,71	0,00	98,00	0,94	C 300	0,00	G 100	67,79	G 200	0,00	164,71	360,00	0,00	44,71	99,00	85,00	G 200	0,00	S 50	82,35	S 50	82,35
I_41 1	0	112,50	318,75	0,00	0,00	-	0,55	C 300	2,10	G 100	102,97	G 200	7,42	120,43	359,68	0,00	0,00	-	0,40	G 200	4,04	S 50	58,20	S 50	58,20
I_41 1	1	108,61	318,75	3,89	0,00	12977,42	0,56	C 300	0,19	G 100	102,34	G 200	6,08	123,89	360,00	0,00	3,89	12976,42	0,40	G 200	5,89	S 50	59,00	S 50	59,00
I_42 0	0	40,00	270,00	0,00	0,00	-	23,81	C 500	20,95	G 100	19,05	S 25	0,00	93,75	318,75	0,00	0,00	-	0,56	S 50	19,53	S 25	37,11	S 25	37,11
I_42 0	1	0,00	270,00	40,00	0,00	38,50	50,00	C 500	0,00	G 100	0,00	S 25	0,00	133,75	318,75	0,00	40,00	37,50	0,24	S 50	39,53	S 25	47,11	S 25	47,11
I_42 1	0	40,00	270,00	0,00	0,00	-	0,04	C 500	24,62	G 100	15,38	S 25	0,00	93,75	318,75	0,00	0,00	-	0,56	S 50	19,53	S 25	37,11	S 25	37,11
I_42 1	1	9,61	270,00	30,39	0,00	1288,50	0,10	C 500	0,07	G 100	9,54	S 25	0,00	124,14	318,75	0,00	30,39	1287,50	0,32	S 50	34,73	S 25	44,71	S 25	44,71
I_43 0	0	55,00	247,50	0,00	0,00	-	0,00	C 300	27,50	G 200	27,50	G 200	0,00	87,50	297,50	0,00	0,00	-	1,09	C 300	51,57	G 100	26,77	S 25	9,16
I_43 0	1	50,88	247,50	4,12	0,00	258,00	0,01	C 300	25,03	G 200	25,85	G 200	0,00	91,62	297,50	0,00	4,12	259,00	0,00	C 300	52,78	G 100	28,13	S 25	10,71
I_43 1	0	55,00	247,50	0,00	0,00	-	0,00	C 300	27,50	G 200	27,50	G 200	0,00	87,50	297,50	0,00	0,00	-	0,01	C 300	50,00	G 100	27,08	S 25	10,42
I_43 1	1	50,88	247,50	4,12	0,00	90258,00	0,01	C 300	25,03	G 200	25,85	G 200	0,00	91,62	297,50	0,00	4,12	90259,00	0,00	C 300	52,78	G 100	28,13	S 25	10,71
I_44 0	0	125,00	300,00	0,00	0,00	-	64,72	G 200	13,89	S 25	55,56	S 50	55,56	66,67	240,00	0,00	0,00	-	0,00	S 25	27,59	S 25	27,59	S 25	11,49
I_44 0	1	108,91	300,00	16,09	0,00	50,00	80,00	G 200	0,00	S 25	55,56	S 50	53,35	82,76	240,00	0,00	16,09	49,00	0,00	S 25	27,59	S 25	27,59	S 25	27,59
I_44 1	0	125,00	300,00	0,00	0,00	-	0,22	G 200	32,38	S 25	49,39	S 50	43,23	66,67	240,00	0,00	0,00	-	0,31	S 25	22,22	S 25	22,22	S 25	22,22
I_44 1	1	108,91	300,00	16,09	0,00	13864,52	0,28	G 200	21,35	S 25	47,71	S 50	39,86	82,76	240,00	0,00	16,09	13863,52	0,00	S 25	27,59	S 25	27,59	S 25	27,59
I_45 0	0	128,57	360,00	0,00	0,00	-	0,40	C 300	16,00	G 100	83,11	G 200	29,46	50,00	315,00	0,00	0,00	-	40,00	C 300	0,00	G 200	44,44	G 200	5,56
I_45 0	1	102,88	360,00	25,69	0,00	229,25	0,45	C 300	3,95	G 100	79,10	G 200	19,83	75,69	315,00	0,00	25,69	230,25	40,00	C 300	-0,00	G 200	44,44	G 200	31,25
I_45 1	0	128,57	360,00	0,00	0,00	-	0,40	C 300	16,00	G 100	83,11	G 200	29,46	50,00	315,00	0,00	0,00	-	0,09	C 300	15,42	G 200	25,10	G 200	9,48
I_45 1	1	96,52	360,00	44,77	12,71	38214,91	0,46	C 300	0,97	G 100	78,10	G 200	17,44	82,05	315,00	12,71	44,77	38213,91	0,05	C 300	28,38	G 200	33,73	G 200	19,95
I_46 0	0	43,33	292,50	0,00	0,00	-	0,09	C 300	15,64	S 50	13,85	S 50	13,85	68,75	233,75	0,00	0,00	-	0,92	C 500	28,54	G 200	28,54	G 100	11,66
I_46 0	1	77,32	292,50	1,19	35,18	305,18	0,00	C 300	40,48	S 50	18,42	S 50	18,42	34,77	233,75	35,18	1,19	304,18	17,27	C 500	14,33	G 200	14,33	G 100	6,10
I_46 1	0	43,33	292,50	0,00	0,00	-	0,09	C 300	15,64	S 50	13,85	S 50	13,85	68,75	233,75	0,00	0,00	-	0,00	C 500	27,83	G 200	28,74	G 100	12,18
I_46 1	1	77,32	292,50	0,00	33,98	155739,68	0,00	C 300	40,48	S 50	18,42	S 50	18,42	34,77	233,75	33,98	0,00	155740,68	0,05	C 500	6,71	G 200	20,29	G 100	7,77
I_47 0	0	87,50	297,50	0,00	0,00	-	77,50	C 500	0,00	G 200	34,72	G 200	52,78	75,00	337,50	0,00	0,00	-	0,78	S 50	14,17	S 25	46,67	S 50	14,17
I_47 0	1	-0,00	297,50	87,50	0,00	46,00	77,50	C 500	0,00	G 200	-0,00	G 200	0,00	162,50	337,50	0,00	87,50	45,00	0,36	S 50	49,17	S 25	64,17	S 50	49,17
I_47 1	0	87,50	297,50	0,00	0,00	-	0,12	C 500	14,24	G 200	47,97	G 200	25,29	89,61	330,19	0,00	0,00	-	0,53	S 50	22,45	S 25	44,72	S 50	22,45
I_47 1	1	19,28	240,93																						

Table B.4: *Continued.* Instances' solutions for *InstanceSetInsights*

I_48 0 0	54,22	276,20	0,00	0,00	-	50,00	S 25	18,07	S 25	18,07	S 50	18,07	39,29	247,50	0,00	0,00	-	24,51	C 300	18,40	G 100	18,40	S 50	2,50
I_48 0 1	93,45	276,25	0,00	39,29	25,00	43,24	S 25	22,76	S 25	52,59	S 50	18,10	0,00	247,50	39,29	0,00	26,00	47,50	C 300	0,00	G 100	0,00	S 50	0,00
I_48 1 0	54,17	276,25	0,00	0,00	-	1,50	S 25	26,69	S 25	26,69	S 50	0,79	40,64	246,55	0,00	0,00	-	0,15	C 300	0,10	G 100	24,86	S 50	15,68
I_48 1 1	51,64	276,25	31,25	28,72	434,52	1,98	S 25	18,53	S 25	18,53	S 50	14,58	41,81	247,50	28,72	31,25	435,52	0,16	C 300	0,11	G 100	25,37	S 50	16,34
I_49 0 0	69,64	276,25	0,00	0,00	-	0,08	C 300	26,46	G 200	33,26	G 200	9,92	70,31	318,75	0,00	0,00	-	0,60	C 300	0,00	S 50	35,16	S 50	35,16
I_49 0 1	15,63	276,25	54,02	0,00	230,35	0,19	C 300	0,00	G 200	15,63	G 200	0,00	124,33	318,75	0,00	54,02	229,35	0,24	C 300	20,99	S 50	51,67	S 50	51,67
I_49 1 0	69,64	276,25	0,00	0,00	-	0,08	C 300	26,46	G 200	33,26	G 200	9,92	90,48	302,61	0,00	0,00	-	0,21	C 300	14,24	S 50	38,12	S 50	38,12
I_49 1 1	15,73	276,25	53,91	0,00	64793,57	0,19	C 300	0,06	G 200	15,67	G 200	0,00	124,22	318,75	0,00	53,91	64794,57	0,24	C 300	20,91	S 50	51,66	S 50	51,66