Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
COPPE
UFRJ

ON A GEOMETRIC GRAPH-COVERING PROBLEM RELATED TO OPTIMAL SAFETY-LANDING-SITE LOCATION

Felipe de Campos Pinto Sinnecker

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Marcia Helena Costa Fampa D.Sc.

Rio de Janeiro
Junho de 2025

ON A GEOMETRIC GRAPH-COVERING PROBLEM RELATED TO OPTIMAL SAFETY-LANDING-SITE LOCATION

Felipe de Campos Pinto Sinnecker

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Marcia Helena Costa Fampa D.Sc.

Aprovada por: Prof. Marcia Helena Costa Fampa
Prof. Claudia D'Ambrosio
Prof. Jon Lee
Prof. Luidi Gelabert Simonetti

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2025

de Campos Pinto Sinnecker, Felipe

On a geometric graph-covering problem related to optimal safety-landing-site location/Felipe de Campos Pinto Sinnecker. – Rio de Janeiro: UFRJ/COPPE, 2025.

X, 51 p.: il.; 29, 7cm.

Orientador: Marcia Helena Costa Fampa D.Sc.

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2025.

Referências Bibliográficas: p. 49 – 51.

1. urban air mobility. 2. set covering. 3. 0/1 linear programming. 4. mixed-integer second-order cone program. 5. variable fixing. I. D.Sc., Marcia Helena Costa Fampa. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Gostaria de expressar minha profunda gratidão à minha orientadora, professora Márcia Fampa, por todo o apoio, paciência e dedicação ao longo de cada etapa deste mestrado. Foi ela quem me apresentou ao tema desta dissertação e esteve presente em todo o processo, mesmo diante das minhas dificuldades, especialmente no último ano, quando seu suporte foi ainda mais fundamental.

Agradeço com carinho aos meus pais, João e Elis, pelo amor incondicional, apoio constante e por sempre acreditarem em mim, mesmo nos momentos mais difíceis.

Aos professores Jon Lee e Claudia D'Ambrosio, sou imensamente grato pela oportunidade de participar da pesquisa que resultou não apenas na minha primeira publicação, como também culminou nesta dissertação.

Ao professor Bernardo Costa, meu orientador de graduação, agradeço por ter me introduzido ao fascinante mundo da otimização, o que certamente foi um dos marcos mais importantes da minha formação acadêmica.

À minha psicóloga, Patricia Kauffmann, deixo um agradecimento especial. Sua escuta, apoio e acolhimento foram fundamentais para que eu pudesse seguir em frente. Sem ela, provavelmente eu não teria conseguido concluir o mestrado.

Agradeço também ao Centro Brasileiro de Pesquisas Físicas (CBPF/MCTI) pela disponibilização de uma maquina computacional de alto desempenho, utilizada na realização dos testes experimentais deste trabalho.

Por fim, agradeço ao CNPq pelo apoio financeiro por meio da bolsa de mestrado, que viabilizou minha dedicação integral à pesquisa.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

ON A GEOMETRIC GRAPH-COVERING PROBLEM RELATED TO OPTIMAL SAFETY-LANDING-SITE LOCATION

Felipe de Campos Pinto Sinnecker

Junho/2025

Orientador: Marcia Helena Costa Fampa D.Sc.

Programa: Engenharia de Sistemas e Computação

O desenvolvimento de sistemas de transporte aéreo urbano tem atraído o interesse de grandes empresas nos últimos anos. Um problema que surge do seu desenvolvimento é a necessidade de locais para pouso de emergência, que devem cumprir certas exigências de segurança. Neste contexto, propomos formulações de programação inteira para o problema de localização ótima de Pontos de Pouso de Emergência (Safety Landing Sites – SLSs). Foram desenvolvidos dois modelos para o problema. O primeiro modelo, baseado em recobrimento de conjuntos, considera que os locais candidatos são finitos e representados por pontos discretos, e é formulado como um problema de programação linear binária. O segundo modelo, trata do caso mais geral, no qual os SLS devem estar contidos em regiões convexas, sendo formulado como um problema de programação inteira mista com restrições cônicas de segunda ordem. Foi desenvolvido um algoritmo gerador de instâncias, as quais foram utilizadas nos experimentos numéricos que validam a aplicabilidade dos modelos. Por fim, introduzimos o *strong fixing* para os dois modelos, uma técnica que permite fixar o valor de variáveis, aprimorando a técnica clássica conhecida como *reduced-cost fixing*. O *strong fixing* demonstrou ser altamente eficaz na redução do tamanho de problemas inteiros nos nossos experimentos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ON A GEOMETRIC GRAPH-COVERING PROBLEM RELATED TO OPTIMAL SAFETY-LANDING-SITE LOCATION

Felipe de Campos Pinto Sinnecker

June/2025

Advisor: Marcia Helena Costa Fampa D.Sc.

Department: Systems Engineering and Computer Science

The development of urban air transportation systems has attracted the interest of major companies in recent years. One problem that arises from this initiative is the need for emergency landing sites, which must meet certain safety requirements. In this context, we propose integer-programming formulations for the problem of optimal location of Safety-Landing Sites (SLSs). Two models were developed for the problem. The first model, based on set covering, considers that candidate sites are finite and represented by discrete points, and is formulated as a binary linear programming problem. The second model considers a more general case, in which the SLS must be contained in convex regions, and is formulated as a mixed-integer second-order cone programming problem. An instance generator algorithm was developed and used in the numerical experiments to validate the applicability of the models. Finally, we introduce "strong fixing" for both models, a technique that allows fixing the value of variables, enhancing the classical technique of reduced-cost fixing. The strong-fixing procedure has proven to be highly effective in reducing the size of the resulting integer problems in our experiments.

Contents

List of Figures

List of Tables

List of Symbols

Chapter 1

Introduction

In recent years, the development of Urban Air Mobility (UAM) has emerged as a field of intense interest for various companies. According to a report by the market research and consulting firm Exactitude Consultancy [7], the market for air mobility is expected to grow at a Compound Annual Growth Rate [1] of 34.3% by 2029. With a growing focus on integrating aerial vehicles into existing transportation systems, both for passenger and cargo transport, the urban mobility landscape is undergoing a remarkable transformation. Renowned companies such as Amazon, iFood, UPS, and FedEx are already adopting drones for delivery services, demonstrating the feasibility and potential of this technology. Additionally, there is increasing corporate interest in establishing the first air transport system for passengers, driven by the demand for more efficient and faster transportation solutions in congested urban areas.

While drones are already being successfully used for cargo deliveries, passenger transport via urban aircraft requires careful consideration of several factors, with safety being the most crucial aspect. The successful implementation of an urban air transport system requires not only the appropriate technology but also the necessary infrastructure to ensure the safety of passengers and urban communities. Air-traffic management (ATM) provides and adapts flight planning to guarantee a proper separation of the trajectories of the flights; see e.g., [2, 13, 15]. In the case of UAM, some infrastructure has to be built to provide safe landing locations in case of failure or damage of drones. These locations are called "Safety Landing Sites" (SLSs), strategically designed locations that allow aircraft to make emergency landings in case of failures or other incidents. These SLSs must be distributed in a way that ensures an aircraft can reach at least one of them at any point along its route, providing an additional layer of safety and reliability to the urban air transport system.

In this work, a study was conducted with the objective of optimizing the construction of SLSs in an aerial transport network. The focus was on installing the minimum-cost set of SLSs, such that all the drones trajectories are covered. Let us consider Figure 1.1 as an example. It represents an aerial 2D view of a part of a city, where the rectangles are the roofs of existing buildings. The black crosses represent potential sites for SLSs. The two red segments are the trajectory of the flights in this portion of the space. The trajectory is fully covered thanks to the installation of 3 SLS over 5 potential sites, namely the ones corresponding to the center of the green circles. The latter represents the points in space that are at a distance that is smaller than the safety distance for an emergency landing. Note that every point along the trajectory is inside at least one circle, thus the trajectory is fully covered.

Thus far, the problem of determining the optimal locations for constructing SLSs has not been widely explored in the literature. A relevant study is Xu's master's thesis [20], in which he investigates the SLS positioning problem combined with drone routing for each origin-destination pair. In that approach, it is assumed that the possible locations for the SLSs completely cover the route, not allowing for partial coverage. In another study conducted by Pelegín and Xu [14], a variant of the problem was considered, formulated as a continuous covering problem. They interpreted this variant as a set covering and location problem, with continuous location candidates and demand across the network.

In the present work, we present two variants of the problem where the demand points are continuous on the network and the candidate-location set is not restricted to be in the network. For the first variant, the location candidates are a finite set composed of points that are not necessarily on the network, as illustrated in the example in Figure 1.1. The second variant considers the locations of SLSs restricted to convex sets only, from which we can choose any point as a location to build an SLS. Within the air transportation context, the second variant provides increased modeling flexibility. It enables the consideration of broader regions, rather than limiting the analysis to a predetermined location.

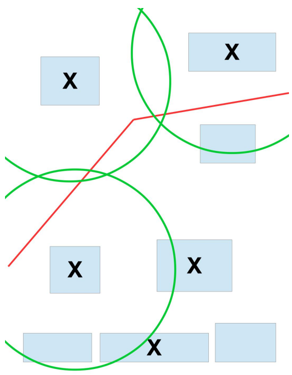

Figure 1.1: Example of full covering provided by installing 3 SLSs

More attention has been accorded to the optimal placement of vertiports; see e.g., [17, 18]. However, their location depends on the estimated service demand, and based on the decisions made on the vertiport location, the UAM network is identified. In contrast, we suppose that these decisions were already made. In fact, despite the scarcity of literature on the topic, the main actors in the UAM field assert that pre-identified emergency landing sites are necessary to guarantee safety in UAM; see e.g., [8, 12].

Organization. In Chapter 2 of this dissertation, we describe our first new mathematical model for the optimal safety-landing-site (SLS) location problem, a generally NP-hard minimum-weight set-covering problem. In Chapter 3, toward practical optimal solution of instances, we introduce "strong fixing", to enhance the classical technique of reduced-cost fixing. In Chapter 4, we present methods applied to reduce the size of the instances of our set-covering problem. In Chapter 5, we explain how we generate test instances for our numerical experiments. In Chapter 6, we present results of computational experiments, demonstrating the value of strong fixing for reducing model size and as a useful tool for solving difficult instances to optimality. In Chapter 7, we propose a mixed-integer second-order program (MISOCP) to model the more difficult version of the problem where the locations of the potential SLSs are not prescribed, they are only restricted to belong to given convex sets. In Chapter 8, we discuss extensions of strong fixing to our MISOCP. In Chapter 9, we present results of computational experiments, demonstrating that the MISOCPs, although more difficult to solve than the set-covering problems, can lead to significant savings on the overall cost of the SLS placements. The results demonstrate once more the effectiveness of the strong-fixing procedure in reducing model size. In Chapter 10 we propose a heuristic to generate upper bounds for CP2 and show that with their application on strong fixing, we can effectively fix variables on our test instances. In Chapter 11, we identify some potential next steps.

This work was presented at the 'International Symposium on Combinatorial Optimization - ISCO 2024, University of La Laguna, Tenerife, Canary Islands, Spain, May 2024', and a short version of it was published in [4].

Chapter 2

Selecting the center of the covering balls from a finite set of points: the SCP problem

We begin with a formally defined geometric optimization problem. Let G be a straight-line embedding of a graph in $\mathbb{R}^d$, $d \geq 1$ (although our main interest is $d = 2$, with $d = 3$ possibly also having some applied interest), where we denote the vertex set of G by $\mathcal{V}(G)$, and the edge set of G by $\mathcal{I}(G)$, which is a finite set of intervals, which we regard as *closed*, thus containing its end vertices. Note that an interval can be a single point (even though this might not be useful for our motivating application). We are further given a finite set N of n points in $\mathbb{R}^d$, a weight function $w : N \to \mathbb{R}_{++}$, and covering radii $r : N \to \mathbb{R}_{++}$ (we emphasize that points in N may have differing covering radii). A point $x \in N$ $r(x)$-covers all points in the $r(x)$-ball $B(x, r(x)) := \{y \in \mathbb{R}^d \ : \ \|x - y\|_2 \leq r(x)\}$. A subset $S \subset N$ r-covers G if every point y in every edge $I \in \mathcal{I}(G)$ is $r(x)$-covered by some point $x \in S$. We may as well assume, for feasibility, that N r-covers G. Our goal is to find a minimum w-weight r-covering of G.

Connecting this geometric problem with our motivating application, we observe that any realistic road network can be approximated to arbitrary precision by a straight-line embedded graph, using extra vertices, in addition to road junctions; this is just the standard technique of piecewise-linear approximation of curves. The point set N corresponds to the set of potential SLSs. In our application, a constant radius for each SLS is rather natural, but our methodology does not require this. We also allow for cost to depend on SLSs, which can be natural if sites are rented, for example.

We note that the intersection of $B(x, r(x))$ and an edge $I \in \mathcal{I}(G)$ is a closed subinterval (possibly empty) of I which we denote by $I(x, r(x))$. Considering the nonempty $I(x, r(x))$ as $x \in N$ varies, we get a finite collection $C(I)$ of nonempty closed sub-intervals of I. The collection $C(I)$ induces a finite collection $\mathcal{C}(I)$ of maximal closed subintervals such that every point y in every subinterval in $\mathcal{C}(I)$ is r-covered by precisely the same subset of N.

With all of this notation, we can re-cast the problem of finding a minimum w-weight r-covering of G as the 0/1-linear optimization problem

$$\begin{aligned} \min \; & \sum_{x \in N} w(x) z(x) \\ \text{s.t.} \; & \sum_{\substack{x \in N \,: \\ J \subset I(x, r(x))}} z(x) \geq 1, \; \forall \; J \in \mathcal{C}(I), \; I \in \mathcal{I}(G); \\ & z(x) \in \{0,1\}, \; \forall \; x \in N. \end{aligned} \tag{CP}$$

Each ball intersects each edge at most twice, possibly generating up to three different subintervals. We have $|\mathcal{C}(I)| \leq 1 + 2n$, for each edge $I \in \mathcal{I}(G)$. Therefore, the number of covering constraints, which we will denote by m, is at most $(1+2n)|\mathcal{I}(G)|$. Of course, we can view this formulation in matrix format as

$$\min\{w^\top z \; : \; Az \geq \mathbf{e}, \; z \in \{0,1\}^n\} \tag{SCP}$$

for an appropriate 0/1-valued $m \times n$ matrix A, and it is this view that we mainly work with in what follows; Figure 2.1 illustrates an example.

	A	B	C	D	E
1	1	0	0	0	0
2	1	1	0	0	0
3	1	1	1	0	0
4	0	1	1	0	0
5	0	1	1	1	0
6	0	0	1	1	0
7	0	0	0	1	0
8	0	0	0	1	1
9	0	0	0	0	1

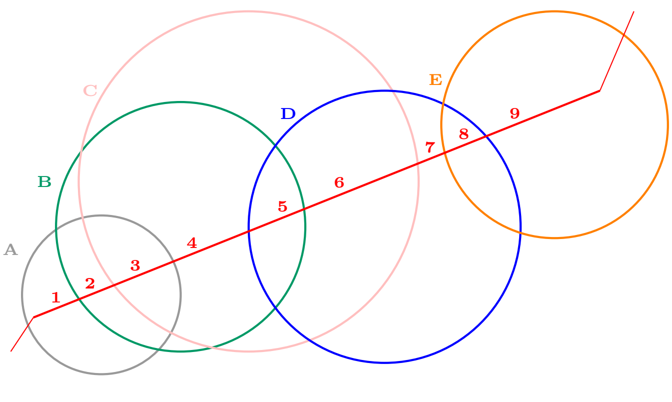

Figure 2.1: Example of the covering constraints

The set covering problem is a classical problem. The NP-hardness of the set covering problem was established by Karp in 1972 [11].

Chapter 3

Strong fixing for SCP

We consider the linear relaxation of SCP, that is $\min\{w^\top z \ : \ Az \geq \mathbf{e}, \ z \geq 0\}$, and the associated dual problem

$$\max\{u^\top \mathbf{e} \ : \ u^\top A \leq w^\top, \ u \geq 0\}. \tag{D}$$

An optimal solution of D is commonly used in the application of *reduced-cost fixing*, see, e.g. [3], a classical technique in integer programming that uses upper bounds on the optimal solution values of minimization problems for inferring variables whose values can be fixed while preserving the optimal solutions. The well-known technique is based on Theorem 1.

Theorem 1. *Let UB be the objective-function value of a feasible solution for* SCP*, and let $\hat{u}$ be a feasible solution for* D*. Then, for every optimal solution $z^\star$ for* SCP*, we have:*

$$z_j^\star \leq \left\lfloor \frac{UB - \hat{u}^\top \mathbf{e}}{w_j - \hat{u}^\top A_{\cdot j}} \right\rfloor, \qquad \forall\, j \in \{1, \ldots, n\} \text{ such that } w_j - \hat{u}^\top A_{\cdot j} > 0. \tag{3.1}$$

Proof. Let $\hat{u}$ be a feasible solution of D, we have $w^\top - \hat{u}^\top A \geq 0$, and also $z^\star \geq 0$:

$$(w^\top - \hat{u}^\top A) z^\star = w^\top z^\star - \hat{u}^\top A z^\star$$

From the constraint of SCP $Az \geq \mathbf{e}$ and $u \geq 0$, we have:

$$u^\top A z \geq u^\top \mathbf{e} \quad \Rightarrow \quad -u^\top \mathbf{e} \geq -u^\top A z$$

Adding $w^\top z^\star$:

$$w^\top z^\star - \hat{u}^\top A z^\star \leq w^\top z^\star - \hat{u}^\top \mathbf{e}$$

For every optimal solution $z^\star$ for SCP and any feasible solution z, $UB = w^\top z \geq$

$w^\top z^\star$. Then,

$$w^\top z^\star - \hat{u}^\top \mathbf{e} \leq UB - \hat{u}^\top \mathbf{e} \quad \Rightarrow \quad (w^\top - \hat{u}^\top A) z^\star \leq UB - \hat{u}^\top \mathbf{e}$$

As $z^\star \geq 0$ and $w^\top - \hat{u}^\top A \geq 0$, for each $j \in \{1, \ldots, n\}$;

$$(w_j - \hat{u}^\top A_j) z_j^\star \leq UB - \hat{u}^\star \mathbf{e}$$

If $(w_j - \hat{u}^\top A_j) > 0$ we have:

$$z_j^\star \leq \frac{UB - \hat{u}^\star \mathbf{e}}{w_j - \hat{u}^\top A_j} \tag{3.2}$$

Since $z^\star \in \{0, 1\}$ we can rewrite 3.2:

$$z_j^\star \leq \left\lfloor \frac{UB - \hat{u}^\top \mathbf{e}}{w_j - \hat{u}^\top A_{\cdot j}} \right\rfloor, \qquad \forall\, j \in \{1, \ldots, n\} \text{ such that } w_j - \hat{u}^\top A_{\cdot j} > 0.$$

□

For a given $j \in \{1, \ldots, n\}$, we should have that the right-hand side in (3.1) equal to 0 to be able to fix the variable z_j at 0 in SCP. Equivalently, we should have $w_j + \hat{u}^\top(\mathbf{e} - A_{\cdot j}) > UB$.

We observe that any feasible solution $\hat{u}$ can be used in (3.1). Then, for all $j \in \{1, \ldots, n\}$, we propose the solution of

$$\mathfrak{z}_j^{\text{SCP}(0)} := w_j + \max\{u^\top(\mathbf{e} - A_{\cdot j}) \ : \ u^\top A \leq w^\top, \ u \geq 0\}. \tag{$\text{F}_j^{\text{SCP}(0)}$}$$

Note that, for each $j \in \{1, \ldots, n\}$, if there is a feasible solution $\hat{u}$ to D that can be used in (3.1) to fix z_j at 0, then the optimal solution of $\text{F}_j^{\text{SCP}(0)}$ has objective value $\mathfrak{z}_j^{\text{SCP}(0)^*}$ greater than UB and can be used as well.

Now, we note that adding the redundant constraint $z \leq \mathbf{e}$ to the linear relaxation of SCP would lead to the modified dual problem

$$\max\{u^\top \mathbf{e} - v^\top \mathbf{e} \ : \ u^\top A - v^\top \leq w^\top, \ u, v \geq 0\}, \tag{D^+}$$

where $v \in \mathbb{R}^n$ is the dual variable corresponding to the redundant constraint. We notice that the vectors $\mathbf{e}$ in the objective are both vectors of ones, but with different dimensions according to the dimensions of u and v. Analogously to Theorem 1, we can establish the following result.

Theorem 2. *Let UB be the objective-function value of a feasible solution for* SCP*, and let* $(\hat{u}, \hat{v})$ *be a feasible solution for* D^+*. Then, for every optimal solution* $z^\star$ *for* SCP*, we have:*

$$z_j^\star = 1, \qquad \forall\, j \in \{1,\dots,n\} \text{ such that } \hat{u}^\top \mathbf{e} - \hat{v}^\top \mathbf{e} + \hat{v}_j > UB. \tag{3.3}$$

Proof. We consider a modified version of SCP where we add to it the constraint $z_{\hat{\jmath}} = 0$, for some $\hat{\jmath} \in \{1,\dots,n\}$. In this case, the only difference in the dual of the continuous relaxation of the modified problem with respect to D^+ is the addition of the new dual variable $\sigma \in \mathbb{R}$ corresponding to this added constraint to the dual constraint corresponding to the variable $z_{\hat{\jmath}}$, which becomes

$$A_{\cdot j}^\top u - v_j + \sigma \leq w_j\,.$$

We consider that $(\hat{u}, \hat{v})$ is feasible to D^+ with objective value $\hat{u}^\top \mathbf{e} - \hat{v}^\top \mathbf{e}$, and we define $\tilde{v}_j := \hat{v}_j + \sigma$, if $j = \hat{\jmath}$, and $\tilde{v}_j := \hat{v}_j$, otherwise. Then $(\hat{u}, \tilde{v}, \sigma)$ is a feasible solution to the modified dual problem with objective value $\hat{u}^\top \mathbf{e} - \hat{v}^\top \mathbf{e} - \sigma$, if $\hat{v}_{\hat{\jmath}} + \sigma \geq 0$. To maximize the objective of the modified dual, we take $\sigma = -\hat{v}_{\hat{\jmath}}$, which gives a lower bound for the optimal value of the modified SCP equal to $\hat{u}^\top \mathbf{e} - \hat{v}^\top \mathbf{e} + \hat{v}_{\hat{\jmath}}$. If this lower bound is strictly greater than a given upper bound UB for the objective value of SCP, we conclude that no optimal solution to SCP can have $z_{\hat{\jmath}} = 0$. □

The redundant constraint $z \leq \mathbf{e}$ can be useful if we search for the *best feasible solution* to D^+ that can be used in (3.3) to fix z_j at 1. With this purpose, we propose the solution of

$$\mathfrak{z}_j^{\text{SCP(1)}} := \max\{u^\top \mathbf{e} - v^\top \mathbf{e} + v_j \ : \ u^\top A - v^\top \leq w^\top,\ u, v \geq 0\}, \tag{$\mathrm{F}_j^{\text{SCP(1)}}$}$$

for all $j \in \{1,\dots,n\}$. If the value of the optimal solution of $\mathrm{F}_j^{\text{SCP(1)}}$ is greater than UB, we can fix z_j at 1.

We call *strong fixing*, the procedure that fixes all possible variables in SCP at 0 and 1, in the context of Theorem 1 and Theorem 2, by using a given upper bound UB on the optimal solution value of SCP, and solving all problems $\mathrm{F}_j^{\text{SCP(0)}}$ and $\mathrm{F}_j^{\text{SCP(1)}}$, for $j \in \{1,\dots,n\}$.

Chapter 4

Reduction for SCP

While solving the instances of SCP, we noted that pre-processing the data could decrease the size of the problem, and make it faster to solve. To test the limits of the reduction we developed a methodology to reduce the problem, by applying the following procedures in the given order.

(a) Reduce the number of constraints in SCP by eliminating dominant rows of the associated constraint matrix A.

(b) Fix variables in the reduced SCP, applying *reduced-cost fixing*, i.e., using Theorem 1, taking $\hat{u}$ as an optimal solution of D. If it was possible to fix variables, reapply (a) to reduce the number of constraints further.

(c) Apply *strong fixing* (see Chapter 3). In case it was possible to fix variables, reapply procedure (a) to reduce the number of constraints in the remaining problem.

4.1 By dominant row elimination

We have that A is a binary matrix. We say that row $A_{j\cdot}$ dominates row $A_{i\cdot}$ if $A_{i\cdot} \subseteq A_{j\cdot}$, meaning that subinterval m_j is covered by all the circles that covers subinterval m_i. See the example in Figure 4.1.

In other words, if a subinterval is covered by multiple circles, selecting any one of them is sufficient. However, if a subinterval is covered by only a single circle, that circle must necessarily be included in the solution. Thus, if a row dominates another, it is redundant and can be removed without affecting the problem's constraints.

Alg. 1 explains the step (a) of the reduction procedure for SCP. We begin by sorting the rows of the matrix in ascending order based on the sum of their elements. This ensures that the first row does not dominate any other, as it contains the fewest number of ones, with at least one at minimum. Then, starting from the first row, we

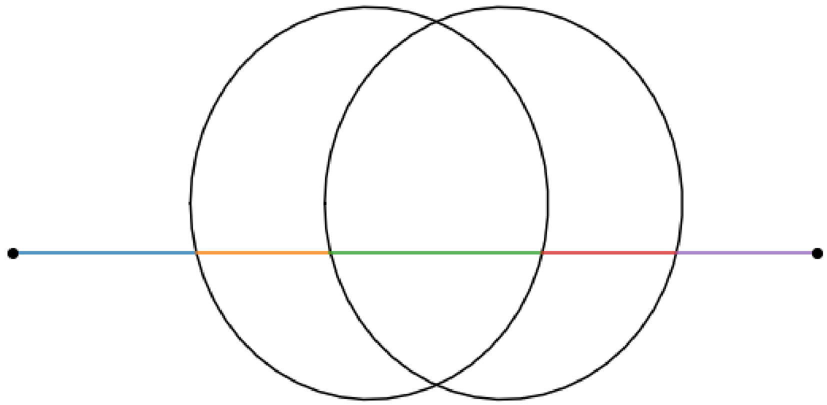

Figure 4.1: The green subinterval dominates the red and yellow subintervals

iteratively append rows to a new matrix. Each new row is compared with the rows already present in the new matrix. To verify dominance, we subtract the row that is already in the matrix from the row we are comparing. Since the new row always contains the same or a greater number of ones, if the smallest resulting element is -1, there is at least one difference, meaning no dominance occurs. However, if the smallest element is 0, it indicates that all elements of the new row cover those of the previous one, confirming dominance. In the case of dominance, we skip the row and move to the next one. Otherwise, we append it to the new matrix.

Algorithm 1: Row reduction

Require: A
Ensure: New matrix without dominant rows.
1: Sort the rows of matrix A, $A_{1\cdot}, \cdots, A_{m\cdot}$ according to the sum of elements
2: Create a new matrix A', append $A_{1\cdot}$ to the matrix A'
3: **for** $A_{i\cdot} \in \{A_{2\cdot}, \cdots, A_{m\cdot}\}$ **do**
4:    **for** $A_{j\cdot} \in A'$ **do**
5:       **if** $A_{j\cdot} \subseteq A_{i\cdot}$ **then**
6:          Skip to step 3 for $A_{i+1,\cdot}$
7:       **end if**
8:    **end for**
9:    Append $A_{i\cdot}$ to A'
10: **end for**
11: **Return** A'

4.2 By strong fixing

After step (b), we aply the *strong fixing* procedure. We adopt the following methodology to do that. We begin with an initial variable $j_1 \in \{1, \cdots n\}$ from SCP, formulating and solving the associated problem $\mathrm{F}_j^{\mathrm{SCP}(0)}$ for $j := j_1$. Using Theorem

1, we fix as many variables as possible. We then select a new variable $j_2 \in \{1, \cdots n\}$ (not yet fixed) based on some criteria used to determine that $A_{\cdot j_2}$ is the column of A most 'similar' to $A_{\cdot j_1}$, such that variable j_2 is not yet fixed. The three criteria considered are explained below. We continue this process until all variables have been processed. Note that for every $j \in \{1, \cdots n\}$, the constraint set of $\mathrm{F}_j^{\mathrm{SCP}(0)}$ remains unchanged, with only the objective function being modified.

The criterion to select the new variable could be arbitrary, however, as the only change in problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$, for all j, is the objective function, we select the new variable in such a way that the new objective function deviates minimally from the previous one. By doing this, and depending on the solution method employed to solve $\mathrm{F}_j^{\mathrm{SCP}(0)}$, we can leverage information from the previous solution to warm start the solver effectively.

For $\mathrm{F}_j^{\mathrm{SCP}(0)}$ the only change introduced when selecting a new variable corresponds to the column of matrix A involved in the objective function $u^\top(\mathbf{e} - A_{\cdot j})$. Therefore, it is natural to define a selection criterion based on the similarity between binary vectors. For this work, the following criteria were used.

Euclidean Distance

For two vectors $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the **Euclidean distance** is given by:

$$d_E(a, b) := \sqrt{\sum (a_i - b_i)^2} \ .$$

This is the most commonly used distance measure in continuous spaces and reflects the geometric distance between points in $\mathbb{R}^n$.

Hamming Distance

Given two vectors $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$, the **Hamming distance**[9] between them is defined as:

$$d_H(a, b) := \sum |a_i - b_i| \ ,$$

This distance metric is particularly well-suited for binary vectors, as it simply counts the number of positions in which the corresponding entries differ. In the context of combinatorial optimization or binary classification, it provides a straightforward measure of dissimilarity.

Jaccard Distance

Let $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$ be two binary vectors. The **Jaccard distance**[10] between them is defined as:

$$d_J(a, b) := 1 - J(a, b),$$

where

$$J(a, b) := \frac{\sum a_i b_i}{\sum a_i + b_i - a_i b_i} \,. \qquad \text{(Jaccard Index)}$$

The Jaccard distance $d_J(a, b)$ is derived from the Jaccard Index, a widely used similarity coefficient that measures the proportion of shared positive entries between two binary vectors. It is particularly relevant in applications involving sparse or set-based binary representations.

Algorithm 2: Strong Fixing

1: Let $N' := \emptyset$ be the set of all variables already fixed by Thm. 1.
2: Select $j_1 \in N := \{1, \ldots, n\}$.
3: **while** $(N \setminus N') \neq \emptyset$ **do**
4: Solve $\mathrm{F}_j^{\mathrm{SCP}(0)}$ for $j := j_1$.
5: Let $N := N \setminus \{j_1\}$.
6: Update $\bar{N}$ adding to it the the set of the new variables fixed by Thm. 1.
7: Select the next variable j_2 using one of the criteria presented above: $j_2 := \mathrm{argmin}_{j \in N \setminus N'} \{d(A_{\cdot j_1}, A_{\cdot j})\}$.
8: Let $j_1 := j_2$.
9: **end while**

Chapter 5

Generating Instances

We have implemented a framework to generate random instances of CP and formulate them as the set-covering problem SCP, which is described in Algorithm 3. The first step of the algorithm was to generate a spatial graph G. For a given number $\nu > 0$, we began by sampling the set of ν vertices $\mathcal{V}(G)$ from the unit square $[0, 1] \times [0, 1]$. For the edges $\mathcal{I}(G)$, we started considering the edges of the Minimum Spanning Tree (MST), ensuring the connectivity of the graph G.

Initially, we also explored the Maximum Spanning Tree; however, as illustrated in Figure 5.1, the Maximum Spanning Tree resulted in a pathological instance, with one vertex having a significantly higher degree than all the others, as well as overlapping edges.

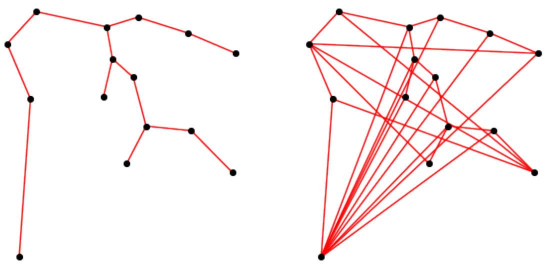

Figure 5.1: MST of V (left) and resulting graph G (right)

Then, we adopted a second approach: computing the Delaunay triangulation of the ν points in $\mathcal{V}(G)$ [5], adding to the set $\mathcal{I}(G)$ all the edges that are not on the convex hull of the points. Using a triangulation, we ensure that the edges do not cross each other. In particular, the Delaunay triangulation has two relevant properties for our application. First, it tends to avoid the formation of 'very thin' triangles by maximizing the smallest angle present in the triangulation [6]. Second, in the plane, the shortest path between two vertices along the edges of the Delaunay triangulation is known to be no longer than 1.998 times the Euclidean distance between them [19], so we always have an upper bound for the maximum distance between the vertices. We can see an example in Figure 5.2.

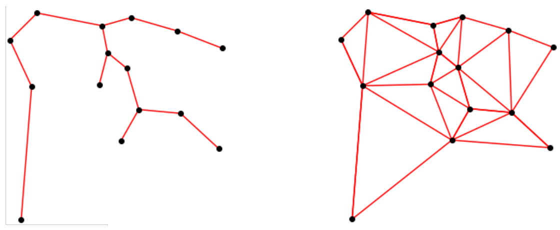

Figure 5.2: Constructing an instance of CP - MST of V (left) and graph G (right)

For the set N of circles, we start by sampling n points c_k in the unit square $[0, 1] \times [0, 1]$, assigning to each, a radius r_k within the interval $[R_{\min}, R_{\max}]$ and a weight w_k within a given interval $[0.5r_k^2, 1.5r_k^2]$. Once both sets N and $\mathcal{I}(G)$ are defined, we compute, for each circle, the intersections with each edge of the problem, generating subintervals of the edges as exemplified in Figure 5.3.

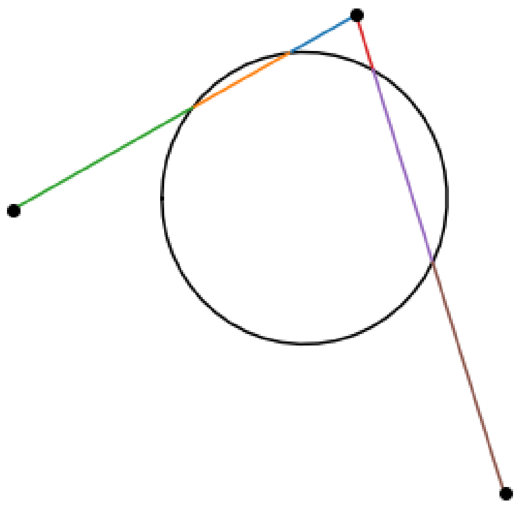

Figure 5.3: Example with 2 edges and 6 subintervals with different colors

After computing all the intersections, we generate the binary matrix A that represents the circles that cover each subinterval of the problem. Note that the instance constructed may be infeasible if any part of an edge of G is not covered by any circle. In this case, we iteratively increase all radii r_k , for $k = 1, \ldots, n$, by 10%, until the instance is feasible. Once all lines of A are not null, the instance is feasible. For feasible instances, we check if there are circles that do not intersect any edges of the graph. If so, we iteratively increase the radius of each of those circles by 10%, until they intercept an edge. By this last procedure, we avoid zero columns in the constraint matrices A associated with our instances of SCP.

In Figure 5.4, we show an instance of CP and its corresponding optimal solution. In the optimal solution, 9 points/circles are selected.

Algorithm 3: Instance Generator

Require: n, ν, $R_{\min}$, $R_{\max}$

Ensure: Instance of CP/SCP.

1: Sample $\nu \in Q := \{[0,1] \times [0,1]\}$ and form the set V.
2: Compute the MST of V, and form the set of edges E_{ST}.
3: Compute the Delaunay Triangulation of V, and form the set of edges E_{DT} excluding the edges that belong to the convex hull.
4: Define the graph $G = (V, E)$, with the set of edges $E := E_{\text{ST}} \cup E_{\text{DT}}$.
5: Sample n points $x_k \in Q := \{[0,1] \times [0,1]\}$, $r_k \in [R_{\min}, R_{\max}]$ and $w_k \in [0.5r_k^2, 1.5r_k^2]$, define c_k as the circle with center x_k, radii r_k and weight w_k for $k = 1, \ldots, n$.
6: **for** each edge $e \in E$ **do**
7:     Compute the intersections between c_k and e
8: **end for**
9: Compute all subintervals and create the intersection matrix A
10: **while** A has a zero line **do**
11:     Raise all radii r_k by 10%.
12:     Do 6-9 again
13: **end while**
14: **while** A has a zero column **do**
15:     Raise the corresponding r_k by 10%
16:     Compute the new matrix A
17: **end while**
18: **Return** $A, \{c\}, \{e\}$

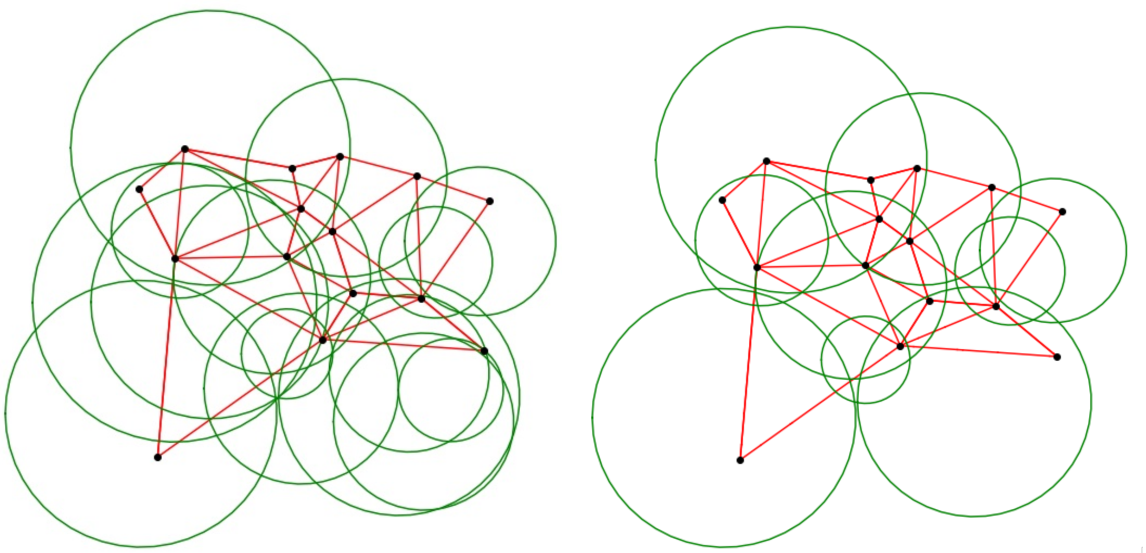

Figure 5.4: Instance data (left) and its optimal solution (right)

Chapter 6

Computational experiments for SCP

The implementation was made in `Python`, using `Gurobi` v. 10.0.2. We ran `Gurobi` with one thread per core, default parameter settings (with the presolve option on). The computer was an 8-core machine (under Ubuntu): Intel i9-9900K CPU running at 3.60GHz, with 32 GB of memory.

For the strong fixing implementation, all the criteria $d(A_{\cdot j}, A)$ were tested, and the Jaccard 4.2 distance outperformed the others. In Table 6.1, the comparison was made using 50 instances, all with the same size (2000 circles and a graph with 60 nodes). The table reports the average of the pairwise time ratios of the strong fixing procedure across all 50 instances for each criteria implementation.

	Euclidean	Hamming	Jaccard
Euclidean	1.00	0.98	0.67
Hamming	1.02	1.00	0.69
Jaccard	1.49	1.46	1.00

Table 6.1: Time ratios comparison for the different distances

In Table 6.2 we present detailed numerical results for five instances of each problem size considered. We aim at observing the impact of strong fixing in reducing the size of SCP, after having already applied standard reduced-cost fixing and eliminating redundant constraints. We display the number of rows (m) and columns (n) of the constraint matrix A after applying the procedures described in steps (a-c) of Chapter 4 to reduce SCP, and the elapsed times (seconds) of the procedures. We have under '`Gurobi`' the size of the original A and the time to solve it with `Gurobi`, under '`Gurobi` presolve', the size of the matrix A after `Gurobi`'s presolve is applied and the time to apply it, and under '`Gurobi` reduced time' the final time to solve the reduced problem. For all instances we set $\nu = 0.03n$. We see that our own presolve, corresponding to procedures (a-c), is effective in reducing the size of the problem, and does not lead in general to problems bigger than the ones obtained with `Gurobi`'s presolve (even though from the increase in the number of variables when compared to the original problem, we see that `Gurobi`'s presolve implements different procedures, such as sparsification on the equation system after adding slack variables). We also see that *strong fixing* is very effective in reducing the size of the problem. Compared to the problems to which it is applied, we have an average decrease of 47% in m and 42% in n. Of course, our presolve is very time-consuming compared to `Gurobi`'s. Nevertheless, for all instances where *strong fixing* can further reduce a significant number of variables, there is an improvement in the final time to solve the problem, and we note that all the steps of reduction and fixing can still be further improved. We also observe that for all reduced instances Gurobi reaches an optimal solution faster than Gurobi's original time; however, in some cases, proving optimality turns out to be more difficult, resulting in a longer runtime than the original, as seen in instances 19 and 25. This shows that *strong fixing* is a promising tool to be adopted in the solution of difficult problems, as is the case of the well-known *strong-branching* procedure.

	`Gurobi`			`Gurobi` presolve			matrix reduction			reduced-cost fixing			strong fixing			`Gurobi` reduced
#	m	n	time	m	n	time	m	n	time	m	n	time	m	n	time	time
1	7164	500	0.28	372	257	0.17	531	500	24.49	214	167	0.50	12	13	5.67	0.00
2	4078	500	0.37	427	327	0.10	574	500	12.08	344	267	0.94	49	52	19.18	0.00
3	3865	500	0.34	399	331	0.10	537	500	10.22	342	291	0.83	46	58	21.39	0.00
4	4399	500	0.27	412	300	0.10	559	500	13.15	310	236	0.79	6	10	11.49	0.00
5	4098	500	0.22	446	359	0.13	546	500	11.92	146	132	0.35	0	0	2.57	0.00
6	13254	1000	2.90	1279	1119	0.73	1728	1000	136.00	1331	705	15.92	244	202	133.39	0.03
7	9709	1000	3.91	1263	1125	0.77	1581	1000	81.86	1443	837	13.33	909	562	484.77	2.29
8	10847	1000	2.53	1307	1140	0.80	1672	1000	94.13	1351	712	15.22	263	196	131.05	0.06
9	9720	1000	2.81	1335	1149	0.69	1700	1000	84.27	1247	667	12.14	209	192	120.86	0.00
10	9377	1000	3.20	1300	1193	0.67	1621	1000	75.76	1329	764	12.17	322	246	161.34	0.08
11	17970	1500	29.46	2720	2222	2.75	3175	1500	278.33	2862	1304	62.73	1061	609	507.11	2.31
12	22022	1500	65.65	2838	2208	3.01	3316	1500	425.03	3115	1336	68.69	1725	844	651.32	9.24
13	18716	1500	42.05	2944	2219	2.86	3466	1500	315.96	3311	1379	82.08	2184	983	646.89	24.44
14	18464	1500	14.30	2545	1988	2.44	3114	1500	306.49	2856	1270	59.32	1548	762	457.58	6.17
15	19053	1500	33.95	2908	2207	2.82	3376	1500	327.13	3162	1361	72.56	1745	851	614.52	7.57
16	29227	2000	170.78	4194	3037	6.59	4988	2000	832.81	4691	1793	177.51	2959	1221	1677.15	83.46
17	29206	2000	212.07	4365	3186	7.04	4993	2000	823.83	4793	1828	188.71	3182	1318	1789.23	102.57
18	28927	2000	762.26	4120	2971	7.22	5016	2000	799.22	4889	1862	192.64	4060	1557	1961.38	497.09
19	30746	2000	6970.90	4943	3645	7.22	5243	2000	842.61	5243	1998	229.23	5109	1944	3602.62	7421.27
20	31606	2000	1914.49	5158	3711	8.20	5378	2000	885.00	5374	1997	241.15	5088	1897	3665.37	1034.66
21	42137	2500	30111.51	6928	4656	14.99	7606	2500	1785.99	7580	2470	517.05	7377	2382	8332.57	26554.56
22	41229	2500	594.63	6511	4454	15.07	7454	2500	1689.45	7127	2344	432.95	3979	1450	4672.70	142.18
23	43100	2500	614.62	6236	4209	15.16	7381	2500	1856.51	7139	2313	462.52	5477	1828	4528.96	494.37
24	42809	2500	27394.59	7210	4836	18.89	7851	2500	964.75	7810	2456	179.03	7557	2362	8474.90	21279.99
25	42641	2500	3594.00	6639	4723	16.43	7038	2500	1615.20	7032	2497	412.83	6604	2360	7047.04	6540.81

Table 6.2: SCP reduction experiment (5 instances of each size)

In Table 6.3, we show the shifted geometric mean (with shift parameter 1) for the same statistics presented in Table 6.2, for 50 instances of each $n = 500, 1000, 1500, 2000$ and for 39 instances of $n = 2500$ (we generated 50 instances with $n = 2500$, but we only consider the 39 instances that could be solved without reduction within our time limit of 10 hours). The average time-reduction factor for all instances solved for $n = 500, 1000, 1500, 2000, 2500$ is respectively, 0.01, 0.13, 0.55, 0.86, and 0.76. The time-reduction factor is the ratio of the elapsed time used by `Gurobi` to solve the problem after we applied the matrix reduction and the strong fixing to the elapsed time used to solve the original problem. When the factor is less than one, our presolve leads to a problem that could be solved faster than the original problem. In Fig. 6.1, we show the number of instances for each

Gurobi			Gurobi presolve			matrix reduction			reduced-cost fixing			strong fixing			Gurobi reduced
m	n	time	m	n	time	m	n	time	m	n	time	m	n	time	time
3324.76	500	0.22	391.18	313.38	0.10	517.59	500	5.49	216.77	179.87	0.31	6.75	8.17	6.43	0.00
9819.57	1000	3.81	1379.35	1220.71	0.82	1673.98	1000	48.84	1281.17	707.19	5.58	183.99	165.37	159.11	0.50
19028.97	1500	70.78	2908.71	2270.53	3.42	3319.23	1500	181.51	3157.20	1377.20	29.35	1971.15	944.04	699.93	29.68
29924.19	2000	1350.81	4721.50	3444.67	7.05	5220.12	2000	836.22	5149.53	1949.80	236.20	4379.65	1693.80	2253.60	1062.38
42038.95	2500	10347.39	6740.81	4636.68	17.39	7442.30	2500	886.65	7372.69	2447.14	163.88	6648.62	2224.60	6868.64	7265.90

Table 6.3: SCP reduction experiment (shifted geometric mean for 50 instances of each size)

$n = 1000, 1500, 2000, 2500$, for which the time-reduction factor is not greater than $0.1, 0.2, \ldots, 1$. For $n = 2500$, for the 39 instances solved, approximately 50% of them had their times reduced to no more than 70% of the original times, and 90% of them had their times reduced. We do not have a plot for $n = 500$ because all of those instances had their times reduced to no more than 10% of the original times.

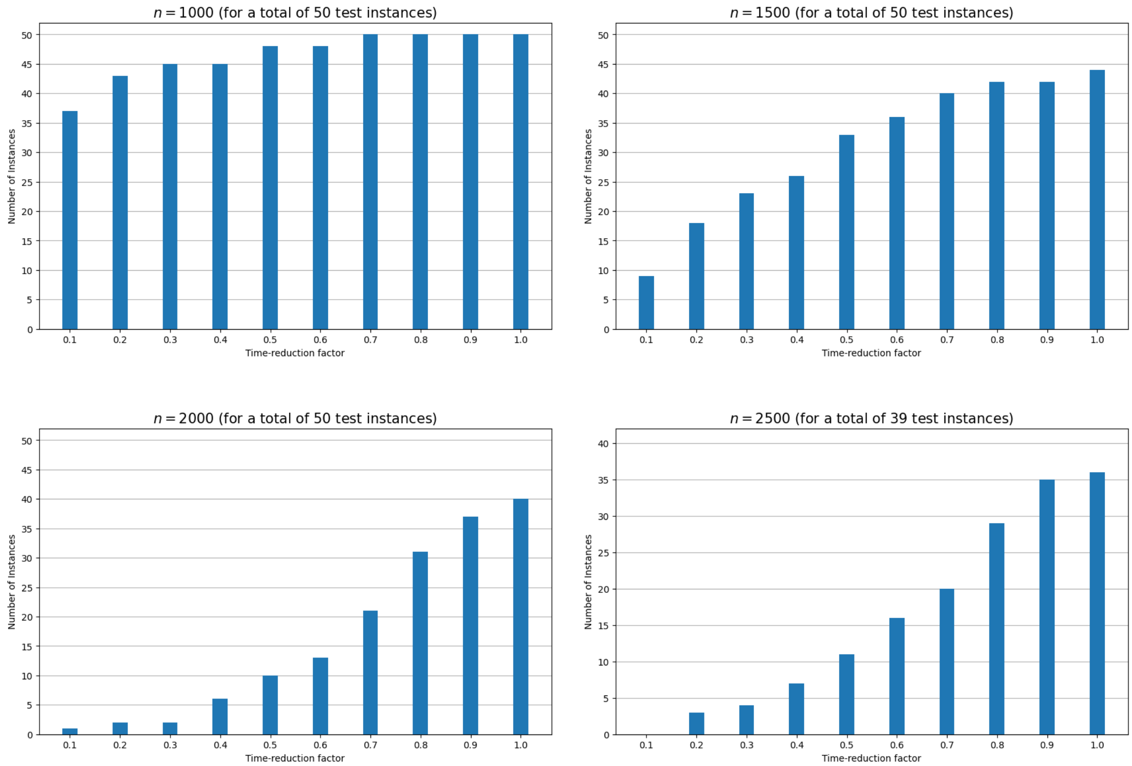

Figure 6.1: Number of SCP instances for each time-reduction factor

6.1 Limiting the number of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ solved for strong fixing

Solving problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ for all $j \in \{1,\ldots,n\}$ can take a long time and may not compensate for the effort to fix variables by the strong-fixing procedure. Here, we analyze the trade-off between the number of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ solved, the number of variables fixed at 0, and the time to solve SCP.

We consider for this analysis, 50 instances of CP generated by Alg. 3, for which the graphs G have 60 nodes and around 200 edges, and the dimension of the constraint matrix $A \in \mathbb{R}^{m\times n}$ of SCP is given by $n = 2000$ and $m \approx 30000$. In Table 6.4, we show details for 10 test instances, specifically, the number of nodes and edges of G, the dimension of the constraint matrix $A \in \mathbb{R}^{m\times n}$ of SCP, the time to solve the instances by `Gurobi`, and the number of variables that were fixed at 0 when applying reduced-cost fixing (n^0_{RC}) and strong fixing (n^0_{SF}).

#	$\lvert\mathcal{V}(G)\rvert$	$\lvert\mathcal{I}(G)\rvert$	m	n	time	n^0_{RC}	n^0_{SF}
1	60	210	29227	2000	163.56	207	779
2	60	210	29206	2000	204.67	172	682
3	60	214	28927	2000	745.25	138	443
4	60	218	30746	2000	6895.17	2	56
5	60	214	31606	2000	1721.46	3	103
6	60	208	28552	2000	346.24	83	379
7	60	214	29785	2000	273.65	73	476
8	60	216	30911	2000	4232.38	57	112
9	60	216	29569	2000	601.11	48	380
10	60	222	31244	2000	2806.55	8	152

Table 6.4: Impact of limiting the number of $\mathrm{F}_j^{\mathrm{SCP}(0)}$ solved for strong fixing

When limiting to an integer $k < n$, the number of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ that we solve, i.e., solving them only for $j \in S \subset \{1,\ldots,n\}$, with $|S| = k$, the selection of the subset S is crucial. We will compare results for two different strategies of selection, which we denote in the following by 'Jaccard' and 'Best-$\mathfrak{z}_j$'. Specifically, after solving $\mathrm{F}_j^{\mathrm{SCP}(0)}$ for $j = i$, we solve it for $j = \hat{\imath}$ such that the variable $z_{\hat{\imath}}$ is still not fixed, and satisfies one of the following criteria. We denote by $J_i \subset \{1,\ldots,n\}$ the set of indices associated to the variables that are still not fixed after solving $\mathrm{F}_i^{\mathrm{SCP}(0)}$.

- Jaccard: When using this strategy, the initial problem solved is the one corresponding to the point on the upper left corner of the rectangle where the points in N are positioned.

- Best-$\mathfrak{z}_j$: $z_{\hat{\imath}}$ is the 'first' variable not fixed, that is

$$\hat{\imath} := \mathrm{argmax}_{j\in J_i}\{w_j + (u^*(F_i))^\top(\mathbf{e} - A_{\cdot j}) - \mathrm{UB}\}.$$

When using this strategy, the initial problem solved is the one corresponding to the smallest right-hand side of (3.1) for the variables still not fixed after solving the linear relaxation of SCP and applying reduced-cost fixing.

To evaluate both these selection strategies of the set S, we also obtain the best possible selection for the set, following the two steps below.

1. We solve problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$, for all $j \in \{1, \ldots, n\}$, and save S_j, which we define as the set of indices corresponding to the variables that we can fix with the solution of $\mathrm{F}_j^{\mathrm{SCP}(0)}$.

2. Then, we solve the problem, where for a given integer k, with $1 \leq k \leq n$, we select the k problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ for which we can fix the maximum number of variables. Next, we will formulate this problem.

   We are given the sets S_j, for $j \in \{1, \ldots, n\}$, and the integer k; and we construct the $n \times n$ matrix F, such that $f_{ij} = 1$ if $i \in S_j$, and $f_{ij} = 0$ otherwise. We define the variable $x_j \in \{0, 1\}$ for each $j \in \{1, \ldots, n\}$, which is 1 if we solve $\mathrm{F}_j^{\mathrm{SCP}(0)}$, and 0 otherwise; and the variable $y_j \in \{0, 1\}$, for each $j \in \{1, \ldots, n\}$, which is 1 if the variable z_j could be fixed by our solution, and 0 otherwise. Then, we formulate the problem as

$$
\begin{array}{l}
\max \ \sum_{j \in N} y_j \\
Fx - y \geq 0, \\
\sum_{j \in N} x_j \leq k, \\
x_j, y_j \in \{0, 1\}, \ \forall \ j \in \{1, \ldots, n\}.
\end{array}
\tag{Best-k}
$$

   The optimal objective value of Best-k gives the maximum number of variables that can be fixed by solving k problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$.

In Figs. 6.2 and 6.3, we present results from our experiments. On the horizontal axis of the plots, we have k as the fraction of the total number of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ that are solved, i.e., for each $k = 0.1, 0.2, \ldots, 1$, we solve kn problems. We present plots for the two selection strategies that we propose ('Jaccard' and 'Best-$\mathfrak{z}_j$'). For comparison purposes, we also consider the best possible selection determined by the solution of Best-k ('Best-k').

In Fig. 6.2, we present average results for the 50 instances tested. In its first sub-figure, we have fractions of the total number of variables that can be fixed at 0 by solving all the n problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$. We see that 'Best-$\mathfrak{z}_j$' is more effective than 'Jaccard' in fixing variables with the solution of a few problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$, approaching the maximum number of variables that can be fixed, given by 'Best-k'. In the second sub-figure, we show the average time-reduction factor for the instances, for the two selection strategies.

In Fig. 6.3, we show the time-reduction factor for the strategy 'Best-$\mathfrak{z}_j$', for each instance described in Table 6.4, but now we account for the time spent on the solution of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$, i.e., the time spent in our presolve is added to the time to solve SCP. We see that when using the 'Best-$\mathfrak{z}_j$' selection strategy, by solving $0.4n$ problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$, we can reduce the solution time for 5 out of 10 instances analyzed. The reduction of time considering the presolve time can be obtained due to the fact that, on average, more than 80% of the number of variables possibly fixed by strong fixing can already be fixed when solving only 40% of the problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ (see Fig. 6.2).

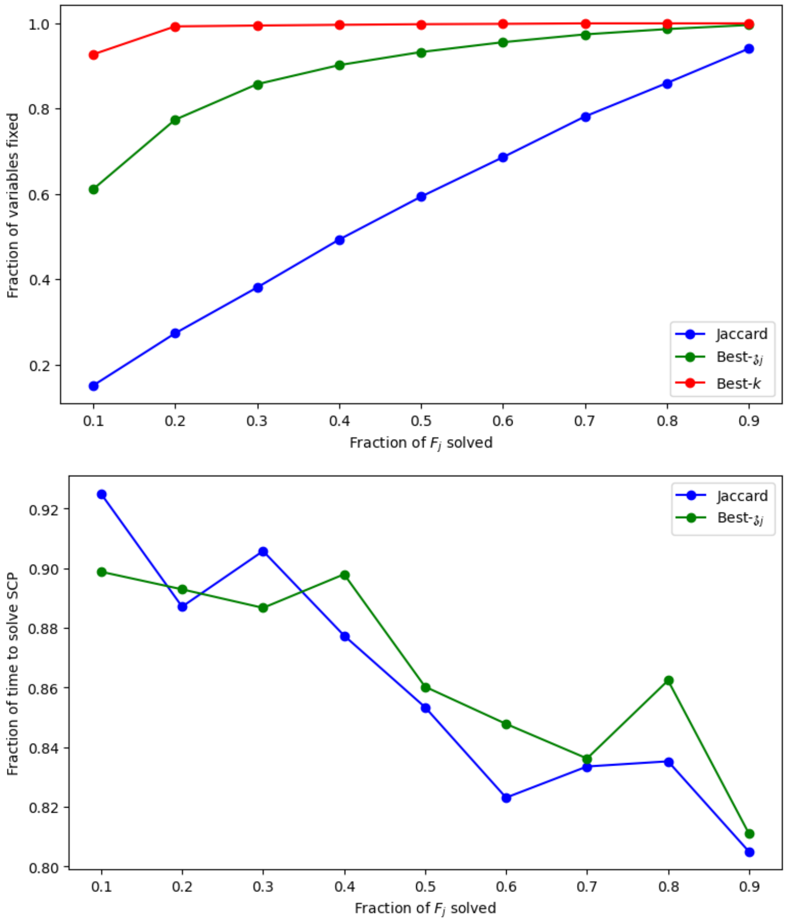

Figure 6.2: Average impact of a number of problems $\mathrm{F}_j^{\mathrm{SCP}(0)}$ solved for strong fixing

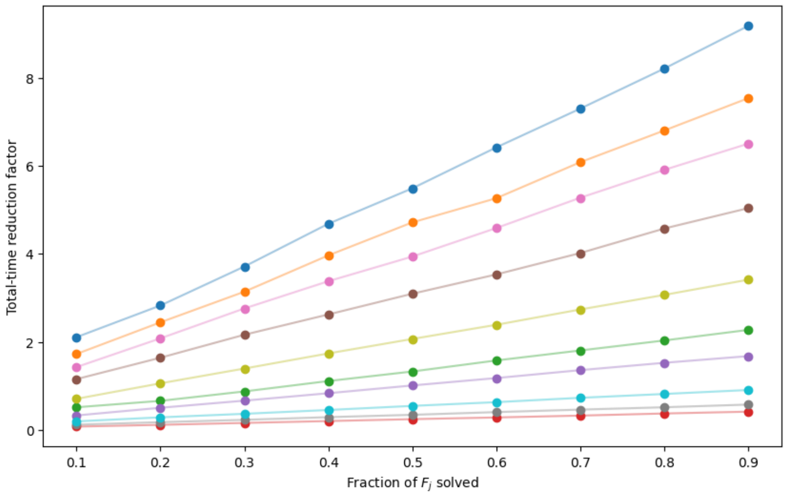

Figure 6.3: Impact of a number of problems $F_j^{SCP(0)}$ solved on the total solution time with strong fixing for the strategy 'Best-$\mathfrak{z}_j$', for instances of Table 6.4

Chapter 7

Selecting the center of the covering balls from a convex set: the CP2 problem

We now have a version of our problem where the exact position of the center of each ball is not prescribed. Rather, for each ball, we are given a bounded convex set where the center could be located. We will now assume that if a ball covers *some* point in an edge, then it covers the entire edge; that is, the edge is either fully contained within the ball or is considered to be outside the ball. This assumption is not extremely restrictive, because we can break each edge into smaller edges, limiting the modeling error, albeit at the expense of adding more nodes to the graph.

We note that in the previous version of the problem, the set where the center of each ball could be located is restricted to an enumerable set of points. Again, we can always increase the number of points, limiting the modeling error, albeit at the expense of increasing the size of the set-covering problem to be solved.

The more appropriate model will depend on which approximation is better for the case studied, the one given by the discretization of the set where the center of each ball could be located or by the partition of the edges into smaller edges such that each one should be completely covered by a unique ball.

Similar to G before, H is a straight-line embedding of a graph in $\mathbb{R}^d$, where we denote the vertex set of H by $\mathcal{V}(H)$, and the edge set of H by $\mathcal{I}(H)$, which is a finite set of closed intervals.

Before, we were given a finite set N of n points in $\mathbb{R}^d$, which were the possible locations to center a ball at. Now, instead, we define $N := \{1, 2, \ldots, n\}$ to be the index set of possible locations for locating a ball. As before, we have a weight function $w : N \to \mathbb{R}_{++}$, and covering radii $r : N \to \mathbb{R}_{++}$.

We let $Q_j \subset \mathbb{R}^d$ be the allowable locations associated with $j \in N$. We will assume that Q_j is "tractable", in the sense that we can impose membership of a point in Q_j as a constraint within a tractable convex optimization model.

We define the variables

- $x_j \in \mathbb{R}^d$, the center of a ball indexed by $j \in N$.
- For $j \in N$, the indicator variable $z_j = 1$ if our solution uses the ball indexed by j in the solution (so its cost is considered in the objective value), and $z_j = 0$, otherwise.
- The indicator variable $y_{ij} = 1$ if our solution uses the ball indexed by j to covers edge i, and $y_{ij} = 0$, otherwise.

We only need a y_{ij} variable if the following d-variable convex system has a feasible solution:

$$x \in Q_j\,,\ \|x - a_{i(1)}\| \leq r_j\,,\ \|x - a_{i(2)}\| \leq r_j\,,\ \text{for edge } i := [a_{i(1)}\,, a_{i(2)})].$$

Then, for each edge $i \in \mathcal{I}(H)$, we let $N_i \subset N$ denote the set of j for which the system above has a feasible solution, and we use N_i in the model to define the constraints.

We are now prepared to formulate our problem as the following mixed integer second order cone program (MISOCP):

$$\begin{array}{lll}
\min & \sum_{j \in N} w_j z_j & \\
\text{s.t.} & z_j \geq y_{ij}\,, & \forall i \in \mathcal{I}(H),\ j \in N_i\,; \\
& \sum_{j \in N_i} y_{ij} \geq 1, & \forall i \in \mathcal{I}(H); \\
& \|x_j - a_{i(k)}\| \leq r_j + M_{ijk}(1 - y_{ij}), & \forall i \in \mathcal{I}(H),\ j \in N_i\,,\ k \in \{1,2\}; \\
& y_{ij} \in \{0,1\}, & \forall i \in \mathcal{I}(H),\ j \in N_i\,; \\
& z_j \in \mathbb{R},\ x_j \in Q_j \subset \mathbb{R}^d, & \forall j \in N.
\end{array} \tag{CP2}$$

Our model has Big-M coefficients. For each edge $i \in \mathcal{I}(H)$, ball index $j \in N_i$, and $k = 1, 2$, we define $M_{ijk} := D_{ijk} - r_j$ where D_{ikj} is the maximum distance between any point in Q_j and the endpoint $a_{i(k)}$, with the edge $i := [a_{i(1)}\,, a_{i(2)})]$. To compute D_{ijk}, assuming that Q_j is a polytope, we compute the Euclidean distance between each vertex of Q_j and the endpoint $a_{i(k)}$ of the edge i. D_{ijk} is then the maximum distance computed (over all the vertices of Q_j). Finally, we note that although it is not explicitly imposed in the model, we have $z_j \in \{0, 1\}$, for all $j \in N$, at an optimal solution of CP2.

Chapter 8

Strong fixing for CP2

Aiming at applying the strong fixing methodology discussed in Chapter 3 to CP2, we now consider its continuous relaxation, i.e., the following second-order cone program (SOCP),

$$
\begin{array}{lll}
\min & \sum_{j\in N} w_j z_j & \\
\text{s.t.} & z_j \geq y_{ij}\,, & \forall i\in\mathcal{I}(H),\ j\in N_i\,; \\
& \sum_{j\in N_i} y_{ij} \geq 1, & \forall i\in\mathcal{I}(H); \\
& \|x_j - a_{i(k)}\| \leq r_j + M_{ijk}(1-y_{ij}), & \forall i\in\mathcal{I}(H),\ j\in N_i\,,\ k\in\{1,2\}; \\
& y_{ij} \geq 0, & \forall i\in\mathcal{I}(H),\ j\in N_i\,; \\
& z_j\in\mathbb{R},\ x_j\in Q_j\subset\mathbb{R}^d, & \forall j\in N,
\end{array}
\tag{$\overline{\text{CP2}}$}
$$

and its associated Lagrangian dual problem presented in Proposition 3.

To simplify the presentation, we will assume here that Q_j is a polytope. So, given $Q_{1j}\,, Q_{2j} \in \mathbb{R}^d$, we have $Q_j := \{x \in \mathbb{R}^d \ : \ Q_{2j} \leq x \leq Q_{1j}\}$, for all $j \in N$.

We also consider the sets $\mathcal{I}^j \subset \mathcal{I}(H)$, for all $j \in N$, defined as

$$\mathcal{I}^j := \{i \in \mathcal{I}(H) \ : \ j \in N_i\}.$$

We note that $\mathcal{I}^j$ is the set of indices of the edges that could be covered by the ball indexed by j.

Proposition 3. *The Lagrangian dual problem of $\overline{CP2}$ can then be formulated as*

$$
\begin{aligned}
\max \;\; & \sum_{j\in N}(\theta_{2j}^\top Q_{2j} - \theta_{1j}^\top Q_{1j}) + \sum_{i\in\mathcal{I}(H)} (\mu_i + \sum_{j\in N_i}\sum_{k=1}^{2}(\gamma_{ijk}^\top a_{ik} - \nu_{ijk}D_{ijk})) \\
s.t. \;\; & w_j - \sum_{i\in\mathcal{I}^j}\lambda_{ij} = 0, && \forall j\in N; \\
& \lambda_{ij} - \mu_i + \sum_{k=1}^{2}\nu_{ijk}M_{ijk} - \beta_{ij} = 0, && \forall i\in\mathcal{I}(H),\ j\in N_i; \\
& \theta_{1j} - \theta_{2j} - \sum_{i\in\mathcal{I}^j}\sum_{k=1}^{2}\gamma_{ijk} = 0, && \forall j\in N; \\
& ||\gamma_{ijk}|| \le \nu_{ijk}, && \forall i\in\mathcal{I}(H),\ j\in N_i,\ k\in\{1,2\}; \\
& \lambda_{ij},\mu_i,\nu_{ijk},\beta_{ij},\gamma_{ijk} \ge 0, && \forall i\in\mathcal{I}(H),\ j\in N_i,\ k\in\{1,2\}; \\
& \theta_{1j},\theta_{2j} \ge 0, && \forall j\in N; \\
& \lambda_{ij},\mu_i,\nu_{ijk},\beta_{ij} \in \mathbb{R}, && \forall i\in\mathcal{I}(H),\ j\in N_i,\ k\in\{1,2\}; \\
& \gamma_{ijk} \in \mathbb{R}^d, && \forall i\in\mathcal{I}(H),\ j\in N_i,\ k\in\{1,2\}; \\
& \theta_{1j},\theta_{2j} \in \mathbb{R}^d, && \forall j\in N.
\end{aligned}
\tag{D2}
$$

Proof. First, we introduce the variable $\rho_{ijk} \in \mathbb{R}^d$, defined as

$$\rho_{ijk} := x_j - a_{ik}\,, \qquad \forall\ i := [a_i(1), a_i(2)] \in \mathcal{I}(H),\ j\in N_i\,,\ k\in\{1,2\}.$$

Then, the Lagrangian function associated with $\overline{\text{CP2}}$ is:

$$
\begin{aligned}
& L(z,y,x,\rho;\lambda,\mu,\nu,\gamma,\beta,\theta) := \\
& \quad \sum_{j\in N} w_j z_j + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\lambda_{ij}(y_{ij} - z_j) + \sum_{i\in\mathcal{I}(H)}\mu_i\left(1 - \sum_{j\in N_i} y_{ij}\right) \\
& \quad + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\nu_{ijk}\left(\|\rho_{ijk}\| - r_j - M_{ijk}(1-y_{ij})\right) \\
& \quad + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\gamma_{ijk}^\top(\rho_{ijk} - x_j + a_{ik}) - \beta_{ij}y_{ij} \\
& \quad + \sum_{j\in N}\left(\theta_{1j}^\top(x_j - Q_{1j}) + \theta_{2j}^\top(Q_{2j} - x_j)\right),
\end{aligned}
$$

or equivalently,

$$
\begin{aligned}
&L(z,y,x,\rho;\lambda,\mu,\nu,\gamma,\beta,\theta) := \\
&\quad \sum_{j\in N} w_j z_j - \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\lambda_{ij}z_j \\
&\quad + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\left(\lambda_{ij}-\mu_i+\sum_{k=1}^{2}\nu_{ijk}C_{ijk}-\beta_{ij}\right)y_{ij} \\
&\quad + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\upsilon_{ijk}||\rho_{ijk}|| + \gamma_{ijk}^\top\rho_{ijk} \\
&\quad + \sum_{j\in N}(\theta_{1j}-\theta_{2j})^\top x_j - \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\gamma_{ijk}^\top x_j + \sum_{i\in\mathcal{I}(H)}\mu_i \\
&\quad - \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\nu_{ijk}D_{ijk} + \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\sum_{k=1}^{2}\gamma_{ijk}^\top a_{ik} \\
&\quad + \sum_{j\in N}\theta_{2j}^\top Q_{2j} - \theta_{1j}^\top Q_{1j}\,.
\end{aligned}
$$

We note that $\inf_{z,y,x,\rho} L = \inf_z L + \inf_y L + \inf_x L + \inf_\rho L$. We have

$$
\begin{aligned}
&\inf_z \sum_{j\in N}\left(w_j - \sum_{i\in\mathcal{I}^j}\lambda_{ij}\right)z_j \\
&\quad = \begin{cases} 0, & \text{if } w_j - \sum_{i\in\mathcal{I}^j}\lambda_{ij} = 0, \quad \forall j\in N; \\ -\infty, & \text{otherwise,}\end{cases} \\
&\inf_y \sum_{i\in\mathcal{I}(H)}\sum_{j\in N_i}\left(\lambda_{ij}-\mu_i+\sum_{k=1}^{2}\nu_{ijk}C_{ijk}-\beta_{ij}\right)y_{ij} \\
&\quad = \begin{cases} 0, & \text{if } \lambda_{ij}-\mu_i+\sum_{k=1}^{2}\nu_{ijk}C_{ijk}-\beta_{ij}=0, \quad \forall i\in\mathcal{I}(H),\ j\in N_i\,; \\ -\infty, & \text{otherwise,}\end{cases} \\
&\inf_x \sum_{j\in N}\left(\theta_{1j}-\theta_{2j}-\sum_{i\in\mathcal{I}^j}\sum_{k=1}^{2}\gamma_{ijk}\right)^\top x_j \\
&\quad = \begin{cases} 0, & \text{if } \theta_{1j}-\theta_{2j}-\left(\sum_{i\in\mathcal{I}^j}\sum_{k=1}^{2}\gamma_{ijk}\right)=0, \quad \forall j\in N; \\ -\infty, & \text{otherwise,}\end{cases}
\end{aligned}
$$

where we use the fact that a linear function is bounded below only when it is identically zero.

To obtain the infimum of the Lagrangian function with respect to ρ, we note that

$$\inf_{\rho} \sum_{i\in\mathcal{I}(H)} \sum_{j\in N_i} \sum_{k=1}^{2} \nu_{ijk}||\rho_{ijk}|| + \gamma_{ijk}^{\top}\rho_{ijk} = \begin{cases} 0, & \text{if } ||\gamma_{ijk}|| \le \nu_{ijk}\,, \quad \forall i\in\mathcal{I}(H),\ j\in N_i\,,\ k\in\{1,2\}; \\ -\infty, & \text{otherwise.} \end{cases} \tag{8.1}$$

To verify 8.1, first suppose that $||\gamma_{ijk}|| > \nu_{ijk}$, we may consider the vector $\rho_{ijk} = -k\gamma_{ijk}$ for any $k > 0$. Then we obtain:

$$\nu_{ijk}||\rho_{ijk}|| + \gamma_{ijk}^{\top}\rho_{ijk} = \nu_{ijk}k||\gamma_{ijk}|| - k||\gamma_{ijk}||^2 = k||\gamma_{ijk}||(\nu_{ijk} - ||\gamma_{ijk}||).$$

Because $(\nu_{ijk} - ||\gamma_{ijk}||) < 0$ and $k||\gamma_{ijk}|| \ge 0$, we have:

$$\lim_{k\to\infty} k||\gamma_{ijk}||(\nu_{ijk} - ||\gamma_{ijk}||) = -\infty.$$

Now, suppose $||\gamma_{ijk}|| \le \nu_{ijk}$, then:

$$(\nu_{ijk} - ||\gamma_{ijk}||)||\rho_{ijk}|| \ge 0 \ \Rightarrow\ \nu_{ijk}||\rho_{ijk}|| - ||\gamma_{ijk}||||\rho_{ijk}|| \ge 0.$$

Using the Cauchy–Schwarz inequality:

$$-||\rho_{ijk}||||\gamma_{ijk}|| \le \gamma_{ijk}^{\top}\rho_{ijk} \le ||\rho_{ijk}||||\gamma_{ijk}||.$$

Then:

$$\nu_{ijk}||\rho_{ijk}|| + \gamma_{ijk}^{\top}\rho_{ijk} \ge \nu_{ijk}||\rho_{ijk}|| - ||\gamma_{ijk}||||\rho_{ijk}|| \ge 0.$$

Thus, $\nu_{ijk}||\rho_{ijk}|| + \gamma_{ijk}^{\top}\rho_{ijk}$ is nonnegative for all ρ_{ijk}, and we conclude that the infimum in 8.1 is attained at 0 (take $\rho = 0$).

Considering the discussion above, it follows that the Lagrangian dual problem of $\overline{\text{CP2}}$ can be formulated as D2. □

Based on convex duality, in Theorem 4 we address the generalization of the technique of *reduced-cost fixing*, discussed in Chapter 3, to CP2. The result is well known and has been widely used in the literature (for example, see [16]).

Theorem 4. *Let UB be the objective-function value of a feasible solution for* CP2*, and let* $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2)$ *be a feasible solution for* D2 *with objective value* $\hat{\xi}$*. Then, for every optimal solution* $(z^\star, y^\star, x^\star)$ *for* CP2*, we have:*

$$y_{ij}^\star = 0, \quad \forall i\in\mathcal{I}(H), j\in N_i \ \textit{such that}\ \hat{\beta}_{ij} > UB - \hat{\xi}. \tag{8.2}$$

Proof. We consider a modified version of CP2 where we add to it the constraint $y_{\hat{\imath}\hat{\jmath}} = 1$, for some $\hat{\imath} \in \mathcal{I}(H)$ and $\hat{\jmath} \in N_i$. In this case, the only difference on the Lagrangian dual of the continuous relaxation of the modified problem with respect to D2 is the subtraction of the new dual variable $\upsilon \in \mathbb{R}$ corresponding to this added constraint, from the objective function, and its addition to the dual constraint corresponding to the variable $y_{\hat{\imath}\hat{\jmath}}$, which becomes

$$\lambda_{\hat{\imath}\hat{\jmath}} - \mu_{\hat{\imath}} + \sum_{k=1}^{2} \nu_{\hat{\imath}\hat{\jmath}k} M_{\hat{\imath}\hat{\jmath}k} - \beta_{\hat{\imath}\hat{\jmath}} + \upsilon = 0.$$

We consider that $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2)$ is feasible to D2 with objective value $\hat{\xi}$, and we define $\tilde{\beta}_{ij} := \hat{\beta}_{ij} + \upsilon$, if $(i, j) = (\hat{\imath}, \hat{\jmath})$, and $\tilde{\beta}_{ij} := \hat{\beta}_{ij}$, otherwise. Then $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \tilde{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \upsilon)$ is a feasible solution to the modified dual problem with objective value $\hat{\xi} - \upsilon$, if $\hat{\beta}_{\hat{\imath}\hat{\jmath}} + \upsilon \geq 0$. To maximize the objective of the modified dual, we take $\upsilon = -\hat{\beta}_{\hat{\imath}\hat{\jmath}}$, which gives a lower bound for the optimal value of the modified CP2 equal to $\hat{\xi} + \hat{\beta}_{\hat{\imath}\hat{\jmath}}$. If this lower bound is strictly greater than a given upper bound UB for the objective value of CP2, we conclude that no optimal solution to CP2 can have $y_{\hat{\imath}\hat{\jmath}} = 1$. □

We observe that any feasible solution to D2 can be used in (8.2). Then, for all (i, j), with $i \in \mathcal{I}(H)$ and $j \in N_i$, we propose the solution of

$$\mathfrak{z}_{ij}^{\text{CP2}} := \max \beta_{ij} + \sum_{j \in N} (\theta_{2j}^{\top} Q_{2j} - \theta_{1j}^{\top} Q_{1j}) + \sum_{i \in \mathcal{I}(H)} (\mu_i + \sum_{j \in N_i} \sum_{k=1}^{2} (\gamma_{ijk}^{\top} a_{ik} - \nu_{ijk} D_{ijk}))$$
$$\text{s.t. } (\lambda, \mu, \nu, \beta, \gamma, \theta_1, \theta_2) \text{ be a feasible solution to D2.} \qquad (\text{F}_{ij}^{\text{CP2}})$$

For each pair (i, j) such that $i \in \mathcal{I}(H)$, $j \in N_i$, if there is a feasible solution $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2)$ to D2 that can be used in (8.2) to fix y_{ij} at 0, then the optimal solution of $\text{F}_{ij}^{\text{CP2}}$ has objective value greater than UB and can be used as well. Thus, we can consider solving all problems $\text{F}_{ij}^{\text{CP2}}$ to fix the maximum number of variables y_{ij} in CP2 at 0. However, from the following result, we see that we can still achieve the same goal by solving a smaller number of problems.

Theorem 5. *Suppose that we are able to conclude from Theorem 4 that for every optimal solution $(z^\star, y^\star, x^\star)$ for* CP2*, we have $y^*_{\hat{\imath}\hat{\jmath}} = 0$, for a given pair $(\hat{\imath}, \hat{\jmath})$, with $\hat{\imath} \in \mathcal{I}(H)$, $\hat{\jmath} \in N_{\hat{\imath}}$. Then, we may also conclude that $z^*_{\hat{\jmath}} = 0$ and $y^*_{i\hat{\jmath}} = 0$, for all $i \in \mathcal{I}^j$, for every optimal solution $(z^\star, y^\star, x^\star)$ for* CP2.

Proof. If $y^*_{\hat{\imath}\hat{\jmath}} = 0$ at every optimal solution for CP2, we cannot have $z_{\hat{\jmath}} = 1$ at any optimal solution, otherwise, setting $y_{\hat{\imath}\hat{\jmath}} = 1$ at this optimal solution, we would not change its objective value and therefore, we would still have an optimal solution for

CP2, contradicting the hypothesis. Then, as $z^*_{\hat{j}} = 0$, by the first constraints in CP2, we see that $y^*_{i\hat{j}} = 0$ for all $i \in \mathcal{I}^{\hat{j}}$. $\square$

We note that we can view the fixing of z_j variables motivated by Theorem 5, as the same as strong fixing on z_j variables with respect to a model that includes the redundant constraint $z \geq 0$ in $\overline{\text{CP2}}$. Following the approach discussed in Chapter 3 for SCP, we can also include the redundant constraint $z \leq \mathbf{e}$ in $\overline{\text{CP2}}$, aiming at fixing variables at 1.

To further explain our strong fixing on z_j, let $\overline{\text{CP2}}^+$ be the continuous relaxation of CP2 with the additional redundant constraints $z \geq 0$ and $z \leq \mathbf{e}$, specifically given by

$$
\begin{array}{lll}
\min & \sum_{j \in N} w_j z_j & \\
\text{s.t.} & z_j \geq y_{ij}\,, & \forall i \in \mathcal{I}(H), j \in N_i\,; \\
& \sum_{j \in N_i} y_{ij} \geq 1, & \forall i \in \mathcal{I}(H); \\
& \|x_j - a_{i(k)}\| \leq r_j + M_{ijk}(1 - y_{ij}), & \forall i \in \mathcal{I}(H), j \in N_i\,, k \in \{1,2\}; \\
& y_{ij} \geq 0, & \forall i \in \mathcal{I}(H), j \in N_i\,; \\
& Q_{2j} \leq x_j \leq Q_{1j}\,, & \forall j \in N; \\
& 0 \leq z_j \leq 1,\ x_j \in \mathbb{R}^d, & \forall j \in N, \\
& y_{ij} \in \mathbb{R}, & \forall\ i \in \mathcal{I}(H), \forall j \in N_i\,.
\end{array}
\tag{$\overline{\text{CP2}}^+$}
$$

With the same analysis used in the proof of Proposition 3 and considering $\phi, \delta \in \mathbb{R}^n$ as the Lagrangian multipliers associated to the new constraints $z \geq 0$ and $z \leq \mathbf{e}$, we can formulate the Lagrangian dual problem of $\overline{\text{CP2}}^+$ as

$$
\begin{array}{lll}
\max & \sum_{j \in N} (\theta_{2j}^\top Q_{2j} - \theta_{1j}^\top Q_{1j} - \delta_j) + \sum_{i \in \mathcal{I}(H)} (\mu_i + \sum_{j \in N_i} \sum_{k=1}^{2} (\gamma_{ijk}^\top a_{ik} - \nu_{ijk} D_{ijk})) & \\
\text{s.t.} & w_j - \phi_j + \delta_j - \sum_{i \in \mathcal{I}^j} \lambda_{ij} = 0, & \forall j \in N; \\
& \lambda_{ij} - \mu_i + \sum_{k=1}^{2} \nu_{ijk} M_{ijk} - \beta_{ij} = 0, & \forall i \in \mathcal{I}(H), j \in N_i\,; \\
& \theta_{1j} - \theta_{2j} - \sum_{i \in \mathcal{I}^j} \sum_{k=1}^{2} \gamma_{ijk} = 0, & \forall j \in N; \\
& ||\gamma_{ijk}|| \leq \nu_{ijk}\,, & \forall i \in \mathcal{I}(H), j \in N_i\,, k \in \{1,2\}; \\
& \lambda_{ij}, \mu_i, \nu_{ijk}, \beta_{ij}, \gamma_{ijk} \geq 0, & \forall i \in \mathcal{I}(H), j \in N_i\,, k \in \{1,2\}; \\
& \theta_{1j}, \theta_{2j}, \phi_j, \delta_j \geq 0, & \forall j \in N; \\
& \lambda_{ij}, \mu_i, \nu_{ijk}, \beta_{ij} \in \mathbb{R}, & \forall i \in \mathcal{I}(H), j \in N_i\,, k \in \{1,2\}; \\
& \gamma_{ijk} \in \mathbb{R}^d, & \forall i \in \mathcal{I}(H), j \in N_i\,, k \in \{1,2\}; \\
& \theta_{1j}, \theta_{2j} \in \mathbb{R}^d, & \forall j \in N.
\end{array}
\tag{D2$^+$}
$$

We can now state the following result, similar to Theorem 4.

Theorem 6. *Let UB be the objective-function value of a feasible solution for* CP2*, and let* $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \hat{\phi}, \hat{\delta})$ *be a feasible solution for* $\mathrm{D2}^+$ *with objective value* $\hat{\xi}$*. Then, for every optimal solution* $(z^\star, y^\star, x^\star)$ *for* CP2*, we have:*

$$z_j^\star = 0, \quad \forall j \in N \text{ such that } \hat{\phi}_j > UB - \hat{\xi}, \tag{8.3}$$

$$z_j^\star = 1, \quad \forall j \in N \text{ such that } \hat{\delta}_j > UB - \hat{\xi}. \tag{8.4}$$

Proof. We first consider a modified version of CP2 where we add to it the constraint $z_{\hat{\jmath}} = 1$, for some $\hat{\jmath} \in N_i$. In this case, the only difference on the Lagrangian dual of the continuous relaxation of the modified problem with respect to $\mathrm{D2}^+$ is the subtraction of the new dual variable $\upsilon \in \mathbb{R}$ corresponding to this added constraint, from the objective function, and its addition to the dual constraint corresponding to the variable $z_{\hat{\jmath}}$, which becomes

$$w_{\hat{\jmath}} - \phi_{\hat{\jmath}} + \delta_{\hat{\jmath}} - \sum_{i \in \mathcal{I}^{\hat{\jmath}}} \lambda_{i\hat{\jmath}} + \upsilon = 0.$$

We consider that $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \hat{\phi}, \hat{\delta})$ is feasible to $\mathrm{D2}^+$ with objective value $\hat{\xi}$, and we define $\tilde{\phi}_j := \hat{\phi}_j + \upsilon$, if $j = \hat{\jmath}$, and $\tilde{\phi}_j := \hat{\phi}_j$, otherwise. Then $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \tilde{\phi}, \hat{\delta}, \upsilon)$ is a feasible solution to the modified dual problem with objective value $\hat{\xi} - \upsilon$, if $\hat{\phi}_{\hat{\jmath}} + \upsilon \geq 0$. To maximize the objective of the modified dual, we take $\upsilon = -\hat{\phi}_{\hat{\jmath}}$, which gives a lower bound for the optimal value of the modified CP2 equal to $\hat{\xi} + \hat{\phi}_{\hat{\jmath}}$. If this lower bound is strictly greater than a given upper bound UB for the objective value of CP2, we conclude that no optimal solution to CP2 can have $z_{\hat{\jmath}} = 1$.

Next, we consider a modified version of CP2 where we add to it the constraint $z_{\hat{\jmath}} = 0$, for some $\hat{\jmath} \in N_i$. In this case, the only difference on the Lagrangian dual of the continuous relaxation of the modified problem with respect to $\mathrm{D2}^+$ is the addition of the new dual variable $\upsilon \in \mathbb{R}$ corresponding to this added constraint to the dual constraint corresponding to the variable $z_{\hat{\jmath}}$, which becomes

$$w_{\hat{\jmath}} - \phi_{\hat{\jmath}} + \delta_{\hat{\jmath}} - \sum_{i \in \mathcal{I}^{\hat{\jmath}}} \lambda_{i\hat{\jmath}} + \upsilon = 0.$$

We consider that $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \hat{\phi}, \hat{\delta})$ is feasible to $\mathrm{D2}^+$ with objective value $\hat{\xi}$, and we define $\tilde{\delta}_j := \hat{\delta}_j - \upsilon$, if $j = \hat{\jmath}$, and $\tilde{\delta}_j := \hat{\delta}_j$, otherwise. Then $(\hat{\lambda}, \hat{\mu}, \hat{\nu}, \hat{\beta}, \hat{\gamma}, \hat{\theta}_1, \hat{\theta}_2, \hat{\phi}, \tilde{\delta}, \upsilon)$ is a feasible solution to the modified dual problem with objective value $\hat{\xi} + \upsilon$, if $\hat{\delta}_{\hat{\jmath}} - \upsilon \geq 0$. To maximize the objective of the modified dual, we take $\upsilon = \hat{\delta}_{\hat{\jmath}}$, which gives a lower bound for the optimal value of the modified CP2 equal to $\hat{\xi} + \hat{\delta}_{\hat{\jmath}}$. If this lower bound is strictly greater than a given upper bound UB for the objective value of CP2, we conclude that no optimal solution to CP2 can have $z_{\hat{\jmath}} = 0$. $\square$

Finally, based on Theorem 6, we propose the solution of the problems

$$\begin{aligned} \mathfrak{z}_j^{\text{CP2(0)}} := \max \phi_j + & \textstyle\sum_{j\in N}(\theta_{2j}^\top Q_{2j} - \theta_{1j}^\top Q_{1j} - \delta_j) \\ & + \textstyle\sum_{i\in\mathcal{I}(H)}(\mu_i + \sum_{j\in N_i}\sum_{k=1}^2(\gamma_{ijk}^\top a_{ik} - \nu_{ijk}D_{ijk})) \\ & \text{s.t. } (\lambda,\mu,\nu,\beta,\gamma,\theta_1,\theta_2,\phi,\delta) \text{ be a feasible solution to D2}^+, \end{aligned} \qquad (\text{F}_j^{\text{CP2(0)}})$$

and

$$\begin{aligned} \mathfrak{z}_j^{\text{CP2(1)}} := \max \delta_j + & \textstyle\sum_{j\in N}(\theta_{2j}^\top Q_{2j} - \theta_{1j}^\top Q_{1j} - \delta_j) \\ & + \textstyle\sum_{i\in\mathcal{I}(H)}(\mu_i + \sum_{j\in N_i}\sum_{k=1}^2(\gamma_{ijk}^\top a_{ik} - \nu_{ijk}D_{ijk})) \\ & \text{s.t. } (\lambda,\mu,\nu,\beta,\gamma,\theta_1,\theta_2,\phi,\delta) \text{ be a feasible solution to D2}^+, \end{aligned} \qquad (\text{F}_j^{\text{CP2(1)}})$$

for all $j \in N$. If the value of the optimal solution of $\text{F}_j^{\text{CP2(0)}}$ (resp., $\text{F}_j^{\text{CP2(1)}}$) is greater than UB, we can fix z_j at 0 (resp., at 1).

Our *strong fixing* procedure for CP2 fixes all possible variables z_j in CP2 at 0 and 1, in the context of Theorem 6, by using a given upper bound UB on the optimal solution value of CP2, and solving problems $\text{F}_j^{\text{CP2(0)}}$ and $\text{F}_j^{\text{CP2(1)}}$, for all $j \in N$.

Chapter 9

Computational experiments for CP2

Our numerical experiments in this section have two goals: to analyze the efficiency of the strong-fixing procedure for CP2, and to compare the solutions of CP2 and SCP.

To generate a test instance to CP2, it is necessary to generate the areas where the SLSs can be located. It is possible to use an instance of CP and then, given the set of n points generated by Alg. 3 (see step 5), simply restrict the feasible areas to a parameterizable convex region around each point. See Fig 9.1 for an example where the convex regions are squares. We note that we can generate a feasible instance of

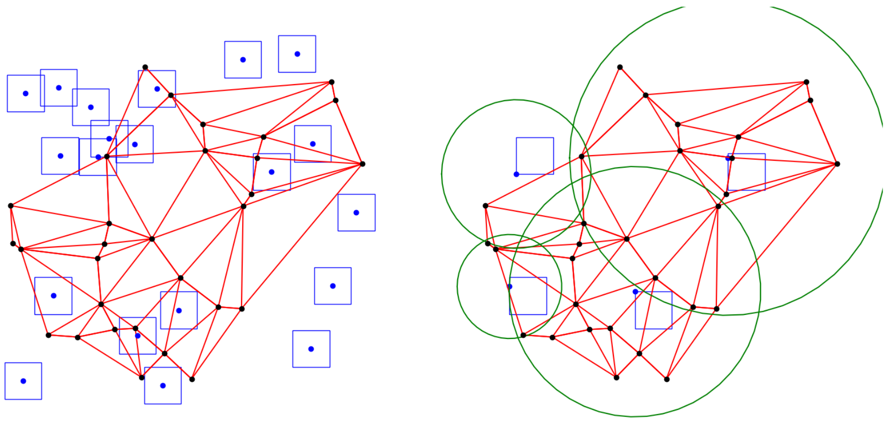

Figure 9.1: Instance data (left) and its optimal solution (right)

CP2 from an infeasible instance of CP in this way (see Fig 9.2).

Then, we have generated test instances to CP2 in the same way we did for SCP, that is, with Alg. 3. The edges in H are given by all the subintervals computed in step 10 of Alg. 3, and the convex sets Q_j, $j = 1, \ldots, n$, are boxes centered on the n points generated in step 5, with sides of length $\ell = 0.05$. We note that if we had $\ell = 0$, CP2 would be equivalent to SCP; by increasing ℓ, we give more flexibility to the possible locations of the centers of the circles that must cover H. Because

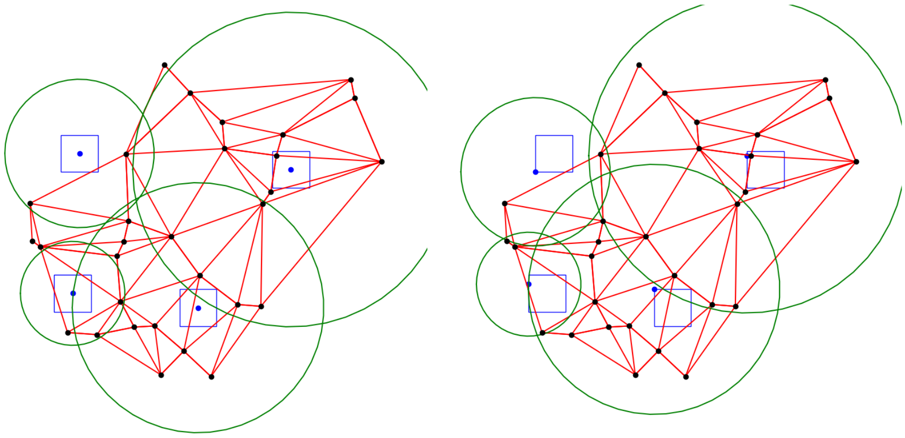

Figure 9.2: Solution was not feasible for CP (left) and it's optimal for CP2 (right)

solving the MISOCP CP2 is more expensive than solving SCP, and as it gives more flexibility in selecting SLS locations, we experiment here with smaller instances and with $\nu = 0.05n$. Thus, compared to the experiments in Chapter 6, where we set $\nu = 0.03n$, we have fewer SLS candidates for a given graph with ν nodes, but we can locate them at any point in the given boxes. We generated 10 instances for each $n \in \{20, 40, 60, 80, 100\}$. Our framework is the same as described in Chapter 6. Our implementation is in `Python`, using `Gurobi` v. 10.0.2 to solve the optimization problems.

Because `Gurobi` does not handle constraints involving norms directly, it is necessary to rewrite the following constraints of CP2:

$$\|x_j - a_{i(k)}\| \leq r_j + M_{ijk}(1 - y_{ij}), \quad \forall i \in \mathcal{I}(H),\, j \in N_i\,,\, k \in \{1, 2\}.$$

Here $a_{i(k)}, r_j, M_{ijk}$ are given constants of the problem, while y_{ij} is a binary variable that determines the constraint that is acually imposed, wich is either $\|x_j - a_{i(k)}\| \leq r_j$ or $\|x_j - a_{i(k)}\| \leq r_j + M_{ijk}$. Considering that M_{ijk} is a 'Big-M' parameter for the model, these constraints are equivalent to either $(x_j - a_{i(k)})^\top (x_j - a_{i(k)}) \leq r_j^2$ or $(x_j - a_{i(k)})^\top (x_j - a_{i(k)}) \leq r_j^2 + M_{ijk}^2$. Therefore, we can replace the constraints with the equivalent quadratic expressions:

$$(x_j - a_{i(k)})^\top (x_j - a_{i(k)}) \leq r_j^2 + M_{ijk}^2(1 - y_{ij}), \quad \forall i \in \mathcal{I}(H),\, j \in N_i\,,\, k \in \{1, 2\}.$$

The same applies to the norm constraints in the dual problem D2, which can be reformulated as:

$$\gamma_{ijk}^\top \gamma_{ijk} \leq \nu_{ijk}^2\,, \forall i \in \mathcal{I}(H), j \in N_i\,,\, k \in \{1, 2\}.$$

In Table 9.1, we show statistics for our experiment. The number of nodes and edges in H, and the number of SLS candidates, are given respectively by '$|\mathcal{V}(H)|$', '$|\mathcal{I}(H)|$', and 'n'. The value of the optimal solution for the two proposed formulations is given by 'opt SCP' and 'opt CP2'. The number of SLSs installed in the optimal solutions for SCP and CP2 is given by 'Σz_j^*' for each problem. The elapsed times to solve CP2 before and after we fix variables by strong fixing procedure are given respectively by '`Gurobi` original time' and '`Gurobi` reduced time', and the number of variables fixed at 0 and 1 by strong fixing are given respectively by 'n_{SF}^0' and 'n_{SF}^1'. Finally, the time to run the strong fixing procedure is given by 'strong fixing time'.

Comparing the optimal solution values of both models, we see that the flexibility regarding the location of SLSs is largely exploited by the solution for CP2, which reduces the cost of SLS installations by 24% on average. It is interesting to note that this cost reduction is not always achieved by installing fewer SLSs. In fact, in some cases, the cost reduction was achieved by installing 3 or 4 more SLSs.

Regarding variable fixing, we note that no variable could be fixed by the standard reduced-cost fixing strategy, where the optimal values of the dual variables β_{ij}, corresponding to the constraints $y_{ij} \geq 0$ in $\overline{\mathrm{CP2}}$, are used in Theorem 4. On the other hand, applying the strong fixing strategy was effective in several instances. We first applied strong fixing on the variables y_{ij} by solving $\mathrm{F}_{ij}^{\mathrm{CP2}}$. Then, motivated by Thms. 5 and 6, we applied strong fixing on the variables z_j by solving $\mathrm{F}_j^{\mathrm{CP2(0)}}$ and $\mathrm{F}_j^{\mathrm{CP2(1)}}$, aiming to fix them at 0 or 1. We observed that the same variables z_j that could be fixed at 0, by solving $\mathrm{F}_{ij}^{\mathrm{CP2}}$, for each variable y_{ij}, and considering Theorem 5, could be fixed by solving $\mathrm{F}_j^{\mathrm{CP2(0)}}$ for each variable z_j. The clear advantage of this observation is that, in general, there are many fewer problems $\mathrm{F}_j^{\mathrm{CP2(0)}}$ to be solved than problems $\mathrm{F}_{ij}^{\mathrm{CP2}}$. Furthermore, when solving $\mathrm{F}_j^{\mathrm{CP2(1)}}$, we could also fix variables z_j at 1.

Six out of the ten smallest instances were solved to optimality through strong fixing, having all variables fixed at 0 or 1. We also note that attempting to fix variables to 1 using $\mathrm{F}_j^{\mathrm{CP2(1)}}$ is less effective than fixing them to 0; only on the smaller instances was it possible to fix variables at 1.

Although a more careful implementation of strong fixing is needed to make it more practical, as concluded in Chapter 6 for SCP, we see it as a promising procedure to reduce the sizes of the instances of CP2 and lead to their faster solution. Comparing the solution times of CP2 before and after we fixed the variables, on the 31 instances where variables were fixed, we have an average time-reduction factor of 60%. The time-reduction factor is the ratio of the elapsed time used by `Gurobi` to solve the problem after we applied strong fixing, to the elapsed time used by `Gurobi` to solve the original problem. We would like to highlight the impact of strong fixing on instance 46, for which fixing 3 variables at 0 led to a time-reduction factor of

#	$\lvert\mathcal{V}(H)\rvert$	$\lvert\mathcal{I}(H)\rvert$	n	opt SCP	opt CP2	Σz_j^* SCP	Σz_j^* CP2	`Gurobi` original time	n_{SF}^0	n_{SF}^1	strong fixing time	`Gurobi` reduced time
1	10	161	20	0.45	0.25	4	3	6.58	15	1	5.80	1.78
2	10	135	20	0.37	0.30	7	7	3.46	13	7	4.15	0.00
3	10	164	20	0.38	0.26	4	5	5.75	14	3	5.91	1.84
4	10	126	20	0.37	0.24	6	4	3.64	16	4	3.87	0.00
5	10	189	20	0.46	0.35	3	2	8.32	18	2	7.15	0.00
6	10	194	20	0.62	0.53	3	3	9.21	17	3	8.72	0.00
7	10	137	20	0.27	0.20	5	4	4.46	16	4	4.50	0.00
8	10	307	20	0.16	0.16	2	2	12.23	18	2	11.16	0.00
9	10	165	20	0.33	0.27	3	3	7.20	16	2	6.37	1.55
10	10	156	20	0.61	0.37	8	7	4.12	4	2	6.51	3.12
11	20	421	40	0.62	0.42	10	6	29.34	14	1	33.19	11.71
12	20	441	40	0.54	0.41	10	7	36.81	16	2	36.41	14.13
13	20	466	40	0.25	0.19	6	3	66.61	21	0	38.69	13.05
14	20	419	40	0.63	0.44	10	9	48.33	10	0	38.45	29.78
15	20	477	40	0.37	0.34	8	7	64.52	5	1	42.96	41.14
16	20	498	40	0.34	0.27	8	7	83.13	1	0	59.09	144.98
17	20	536	40	0.58	0.43	8	8	60.18	10	0	58.58	24.64
18	20	430	40	0.58	0.45	6	5	26.83	15	0	39.36	14.77
19	20	367	40	0.37	0.27	8	7	17.09	12	1	30.32	11.75
20	20	412	40	0.36	0.26	10	9	59.90	10	0	41.48	32.20
21	30	791	60	0.43	0.32	7	10	229.94	6	0	153.03	175.96
22	30	730	60	0.46	0.34	9	11	199.66	0	0	121.65	199.66
23	30	825	60	0.45	0.37	11	10	225.36	0	0	155.59	225.36
24	30	828	60	0.26	0.20	11	7	93.40	1	0	150.03	175.27
25	30	827	60	0.28	0.20	7	8	197.69	9	0	155.22	150.33
26	30	826	60	0.28	0.21	9	8	528.42	5	0	150.14	244.13
27	30	920	60	0.63	0.46	11	9	336.27	0	0	181.68	336.27
28	30	779	60	0.44	0.34	12	12	128.48	1	0	129.27	205.00
29	30	832	60	0.35	0.27	10	8	178.64	1	0	152.29	176.81
30	30	820	60	0.33	0.28	8	8	172.71	0	0	143.70	172.71
31	40	1195	80	0.42	0.29	10	12	802.56	0	0	314.79	802.56
32	40	1228	80	0.30	0.23	10	13	627.10	1	0	332.78	403.99
33	40	1199	80	0.64	0.46	13	10	636.39	0	0	346.73	636.39
34	40	1150	80	0.55	0.39	11	13	678.80	2	0	294.97	846.46
35	40	1392	80	0.27	0.22	11	12	1186.35	0	0	454.88	1186.35
36	40	1297	80	0.62	0.46	13	10	4861.33	0	0	362.33	4861.33
37	40	1150	80	0.48	0.38	10	9	685.53	2	0	302.88	805.20
38	40	1455	80	0.50	0.38	12	11	926.16	3	0	467.83	827.88
39	40	1257	80	0.65	0.50	8	9	610.34	0	0	362.82	610.34
40	40	1221	80	0.25	0.18	10	12	1484.83	0	0	344.11	1484.83
41	50	1713	100	0.42	0.34	10	10	7368.55	0	0	743.31	7368.54
42	50	1788	100	0.29	0.24	9	9	7282.17	0	0	791.29	7282.17
43	50	2101	100	0.42	0.35	9	8	3981.00	0	0	1059.55	3981.00
44	50	1695	100	0.47	0.39	11	9	7364.34	0	0	665.12	7364.34
45	50	1856	100	0.30	0.26	12	8	7397.04	0	0	798.32	7397.04
46	50	1866	100	0.49	0.34	9	8	5948.14	3	0	854.47	2877.52
47	50	1811	100	0.35	0.29	10	9	7376.13	0	0	732.93	7376.13
48	50	1914	100	0.46	0.40	12	10	7402.76	0	0	845.46	7402.76
49	50	1850	100	0.38	0.30	8	12	5292.99	0	0	815.92	5292.99
50	50	1744	100	0.60	0.48	11	8	7376.88	0	0	722.58	7376.88

Table 9.1: CP2 reduction experiment and solution comparison with SCP

48%, and even if we consider the time spent on the strong fixing procedure we obtain a time-reduction factor of 62%.

Chapter 10

Heuristic to generate upper bounds for fixing variables in CP2

In order to evaluate the greatest impact of the strong-fixing procedure, for the numerical experiments discussed in the previous chapters, the upper bounds UB for SCP and CP2 used to fix variables, were given by the optimal solution value of the instances, computed in advance. Here, we test the strong-fixing procedure for our more difficult problem CP2 in a more practical way, where UB is given by a heuristic solution.

We note that for the instances generated for the numerical experiments discussed in Chapters 6 and 9, the optimal binary variable $\bar{z}$ for an instance of SCP can be used to compute a feasible solution for the corresponding instance of CP2 with $z := \bar{z}$ and the same objective value $w^\top \bar{z}$ as the optimal value of SCP. Note that for an instance of CP2, we define the convex sets where the SLSs can be located as the unit squares centered at the candidate points of SCP, so we could just construct a feasible solution for CP2 by setting the positions of the SLSs as the candidate points selected at the solution of SCP. We consider this and the fact that SCP is computationally cheaper to solve than CP2, as the basis for our heuristic, which attempts to improve the upper bound for CP2 given by the optimal solution value of SCP. We seek this attempt by relaxing SCP, and obtaining like this, optimal solutions with values closer to the optimal solution of CP2. The relaxation of SCP is obtained by slightly increasing the radii of the circles, and recomputing the coverage matrix A as explained in Chapter 5. We note that, as the original radii increase, there is a risk for the optimal solution $\bar{z}$ of the relaxed SCP to lead to an infeasible solution for CP2, so we need a method to verify if it is still possible to construct a feasible solution for CP2 with $z := \bar{z}$, and therefore, be able consider $w^\top \bar{z}$ as a valid upper bound for it.

In our heuristic for CP2, we start by increasing the value of all radii r_j, $j \in N$, of the original SCP by $\alpha\%$, then we compute the new matrix A for the relaxed problem. Solving the resulting relaxed SCP we obtain a solution $\bar{z}$. We then verify if it is possible to construct a feasible solution for CP2 where we set $z := \bar{z}$. More specifically, we take the circles present in the solution of the relaxed SCP, determined by $\bar{N} = \{j \in N \;:\; \bar{z}_j = 1\}$, and we first verify if for each edge $i := [a_{i(1)}\,, a_{i(2)}] \in \mathcal{I}(H)$ of the original SCP, there is at least one $j \in \bar{N}$, for which the system

$$x \in Q_j\,,\ \|x - a_{i(1)}\| \leq r_j\,,\ \|x - a_{i(2)}\| \leq r_j \tag{10.1}$$

has a feasible solution. If there is even a single edge for which the system has no feasible solution for all j, then we certainly cannot construct a feasible solution to CP2 with $z := \bar{z}$, and we discard it. Otherwise, we investigate further if it is possible to construct a feasible solution to CP2 with $z := \bar{z}$, by solving the following problem.

$$\begin{array}{lll}
\max & \sum_{i\in\mathcal{I}(H)} v_i & \\
\text{s.t.} & v_i \leq 1 & \forall i\in\mathcal{I}(H); \\
& \sum_{j\in N_i} y_{ij} \geq v_i, & \forall i\in\mathcal{I}(H); \\
& \|x_j - a_{i(k)}\| \leq r_j + M_{ijk}(1 - y_{ij}), & \forall i\in\mathcal{I}(H),\ j\in\bar{N}_i\,,\ k\in\{1,2\}; \\
& y_{ij} \in \{0,1\}, & \forall i\in\mathcal{I}(H),\ j\in\bar{N}_i\,; \\
& v_i \in \mathbb{R},\ x_j \in Q_j \subset \mathbb{R}^d, & \forall j\in\bar{N}.
\end{array} \tag{VC}$$

where, for each $i \in \mathcal{I}(H)$, we denote by $\bar{N}_i$, the set of indices $j \in \bar{N}$ for which (10.1) has a feasible solution.

As in CP2, the binary variable y_{ij} indicates whether the circle j is covering the edge i, and the variable x_j indicates the location of the SLS indexed by j. With the new variable v_i, we formulate a relaxed form of the constraint from CP2, where we assure that all edges are covered. In the optimal solution of VC, we have $v_i = 1$ if edge i is covered by at least one circle j, and $v_i = 0$, otherwise. The optimal objective function value of VC will be equal to the total number of edges in H if and only if for each edge $i \in \mathcal{I}(H)$ of the original SCP, there is at least one $j \in \bar{N}$, for which (10.1) is feasible. In this case, the optimal solution value of the relaxed SCP, given by $w^\top \bar{z}$, is an upper bound for CP2, hopefully better than the upper bound given by the optimal solution value of the original SCP.

Problem VC aims to cover the maximum number of edges using only the circles present in $\bar{N}$. However, it is possible that for some edge i, we have at the optimal solution, $\sum_{j\in N_i} y_{ij} = 0$, in which case $v_i = 0$, and thus $\sum v_i < \sum 1 = |\mathcal{I}(H)|$. We note that even if system (10.1) has a feasible solution for at least one j, for every edge i, i.e., even if $\bar{N}_i \neq \emptyset$ for every edge i, we may have $v_i = 0$ for some edge i in the optimal solution of VC. This could occur, for example, when to cover an edge i_1, the circle j has to be centered at a given vertex of the corresponding unit square, and to cover an edge i_2, the circle j has to be centered at another vertex of the square.

Problem VC not only has fewer variables than CP2, but it is also a relaxed version of CP2 that is always feasible, resulting in a faster solution time.

If the objective value of VC is not equal to $|\mathcal{I}(H)|$, we decrease the value of α by 10% until the solution leads to a valid upper bound.

The following numerical experiments have the goal of analyzing the efficiency of the strong-fixing procedure for CP2 in a more practical approach where the optimal solution value is not initially known. We used the same test instances of Chapter 9.

For the relaxed instances of SCP we tested a percentage increase in the radii for each $\alpha \in \{10, 12, 14, 16, 18, 20\}$, when the solution of the relaxed SCP was not feasible to the original SCP, α was decreased by 10% until feasibility was achieved and an upper bound for CP2 was generated. As noted in Chapter 9, attempting to fix variables at 1 was effective only for the smaller instances, even when using the optimal solution of CP2 as an upper bound for fixing the variables. Therefore, the experiments in this chapter use Thms. 5 and 6 attempting to fix variables only at 0, by solving $\mathrm{F}_j^{\mathrm{CP2}(0)}$.

In Tables 10.1–10.6, we show statistics for our experiment. The number of nodes and edges in H and the number of candidate SLSs, are given respectively by '$|\mathcal{V}(H)|$', '$|\mathcal{I}(H)|$', and 'n'. The elapsed times to solve CP2 before and after we fix variables by the strong-fixing procedure are given respectively by '`Gurobi` original time' and '`Gurobi` reduced time'. The maximum number of variables that can be fixed at 0, using the optimal solution value of CP2 as an upper bound (as done in Chapter 9) and the number of variables fixed by the strong-fixing procedure, using the new upper bound obtained with our heuristic solution, are given respectively by 'Max Fix' and 'n^0_{SF}'. The elapsed time of the heuristic, the number of times we obtained a solution $\bar{z}$ of the relaxed SCP and it was not possible to construct a feasible solution for CP2 with $z := \bar{z}$, and the time to run the strong fixing procedure are given respectively by 'upper bound time', 'infeasible relaxed SCP' and 'strong fixing time'. Finally, the ratio between the total time spent in all steps and the original time needed to solve CP2 is given by 'total time factor'.

As observed in the results of 10.1, increasing the radii on SCP by only 10%, we can already get a good result for the smaller instances, with instances 5 and 8 fixing all possible variables. The best result is for instance 13, even without fixing all possible variables, the total time factor is 79%, with the reduced problem being solved in just 32% of the original `Gurobi` time. For instance 15 the total time factor is 106%, being very close to the original `Gurobi` time. Most instances did not generate an infeasible solution for the problem.

For the $\alpha = 12\%$ increase shown in Table 10.2, although there is a slight increase in the number of fixed variables for the smaller instances, the overall results are very similar to those of Table 10.1. For the $\alpha = 14\%$ increase shown in Table 10.3, all possible variables were fixed for instance 6. The most notable improvement is the fixing of 1 variable for instance 25 of size with $|\mathcal{V}(H)| = 30$. For instance 13, there was an increase in the number of variables fixed but also one infeasible solution.

For $\alpha = 16\%$ and higher increases, shown in Tables 10.4–10.6, the most notable positive result is instance 46. By applying strong fixing with an upper bound obtained with our heuristic, all the possible variables were fixed. The total time factor is 56% for an instance that originally required over an hour to be solved. It is worth noting that an infeasible solution from the relaxed SCP was obtained initially, meaning the effective increase in radii was of 14.4% and the heuristic had to solve VC one additional time. The frequency of infeasible solutions for the original SCP from the solutions of the relaxed SCP increases for larger instances and starts to appear even in smaller ones as the radii increase, so further increases in radii are not interesting for this strategy.

From Table 9.1 it is important to note that increasing the number of possible SLSs in the instance does not necessarily lead to a higher number of SLSs present in the optimal solution, those with $n > 40$ tend to have a similar number of SLSs in their optimal solution. This directly impacts the time required to validate a solution, as the number of elements in $\bar{N}$ is similar across larger instances. Additionally, the main factor affecting the heuristic's runtime is the need to repeat the process until a feasible solution is found. Nevertheless, as seen in the case of instance 46, even under a worst-case scenario observed where three infeasible solutions were generated 10.6, compared to the best-case scenario 10.4 where only once the solution was invalid and all variables were fixed, the total time factor increased by only 1%.

#	$\lvert\mathcal{V}(H)\rvert$	$\lvert\mathcal{I}(H)\rvert$	n	Gurobi original time	Max Fix	n^0_{SF}	upper bound time	infeasible relaxed SCP	strong fixing time	Gurobi reduced time	total time factor
1	10	161	20	6.58	15	2	1.17	0	5.21	6.02	1.88
2	10	135	20	3.46	13	4	1.25	0	3.23	2.71	2.05
3	10	164	20	5.75	14	2	1.58	0	4.73	4.88	1.96
4	10	126	20	3.64	16	2	1.23	0	3.24	3.10	2.09
5	10	189	20	8.32	18	18	1.50	0	6.26	1.44	1.12
6	10	194	20	9.21	17	13	1.54	0	7.04	2.76	1.24
7	10	137	20	4.46	16	15	1.25	0	3.77	1.24	1.42
8	10	307	20	12.23	18	18	2.84	0	9.40	2.83	1.24
9	10	165	20	7.20	16	9	1.22	0	5.51	4.34	1.56
10	10	156	20	4.12	4	0	1.69	0	3.66	4.17	2.28
11	20	421	40	29.34	14	0	4.04	0	20.27	29.04	1.84
12	20	441	40	36.81	16	8	4.74	0	22.68	37.30	1.78
13	20	466	40	66.61	21	11	4.06	0	27.15	21.47	0.79
14	20	419	40	48.33	10	2	3.88	0	22.05	32.61	1.22
15	20	477	40	64.52	5	4	4.33	0	22.51	41.36	1.06
16	20	498	40	83.13	1	0	18.92	3	28.45	82.78	1.57
17	20	536	40	60.18	10	0	11.43	1	34.40	59.57	1.77
18	20	430	40	26.83	15	1	7.38	1	25.65	26.80	2.27
19	20	367	40	17.09	12	1	3.51	0	18.56	28.62	3.03
20	20	412	40	59.90	10	0	4.41	0	23.84	59.79	1.47
21	30	791	60	229.94	6	0	16.93	1	79.12	229.94	1.42
22	30	730	60	199.66	0	0	30.45	3	58.03	199.66	1.44
23	30	825	60	225.36	0	0	7.70	0	73.04	225.36	1.36
24	30	828	60	93.40	1	0	8.68	0	71.61	93.40	1.86
25	30	827	60	197.69	9	0	7.36	0	84.21	197.69	1.46
26	30	826	60	528.42	5	0	7.87	0	75.45	528.42	1.16
27	30	920	60	336.27	0	0	26.24	2	86.96	336.27	1.34
28	30	779	60	128.48	1	0	8.57	0	60.86	128.48	1.54
29	30	832	60	178.64	1	0	7.54	0	71.91	178.64	1.44
30	30	820	60	172.71	0	0	8.06	0	68.76	172.71	1.44
31	40	1195	80	802.56	0	0	12.27	0	151.54	802.56	1.20
32	40	1228	80	627.10	1	0	12.37	0	160.54	627.10	1.28
33	40	1199	80	636.39	0	0	12.42	0	164.05	636.39	1.28
34	40	1150	80	678.80	2	0	13.02	0	144.24	678.80	1.23
35	40	1392	80	1186.35	0	0	12.91	0	217.08	1,186.35	1.19
36	40	1297	80	4861.33	0	0	56.29	3	173.03	4,861.33	1.05
37	40	1150	80	685.53	2	0	11.66	0	147.51	685.53	1.23
38	40	1455	80	926.16	3	0	14.29	0	230.38	926.16	1.26
39	40	1257	80	610.34	0	0	11.74	0	173.21	610.34	1.30
40	40	1221	80	1484.83	0	0	25.26	1	161.29	1,484.83	1.13
41	50	1713	100	7368.55	0	0	18.86	0	338.68	7,368.54	1.05
42	50	1788	100	7282.17	0	0	19.44	0	375.21	7,282.17	1.05
43	50	2101	100	3981.00	0	0	22.20	0	492.87	3,981.00	1.13
44	50	1695	100	7364.34	0	0	54.72	2	317.71	7,364.34	1.05
45	50	1856	100	7397.04	0	0	38.43	1	372.06	7,397.04	1.06
46	50	1866	100	5948.14	3	0	40.10	1	415.68	5,948.14	1.08
47	50	1811	100	7376.13	0	0	55.61	2	352.09	7,376.13	1.06
48	50	1914	100	7402.76	0	0	19.89	0	401.97	7,402.76	1.06
49	50	1850	100	5292.99	0	0	57.35	2	380.90	5,292.99	1.08
50	50	1744	100	7376.88	0	0	33.14	1	346.48	7,376.88	1.05

Table 10.1: Strong Fixing with upper bound from relaxed SCP, $\alpha = 10\%$

#	$\vert\mathcal{V}(H)\vert$	$\vert\mathcal{I}(H)\vert$	n	Gurobi original time	Max Fix	n_{SF}^0	upper bound time	infeasible relaxed SCP	strong fixing time	Gurobi reduced time	total time factor
1	10	161	20	6.58	15	11	1.13	0	4.71	3.07	1.35
2	10	135	20	3.46	13	4	1.26	0	3.24	2.72	2.05
3	10	164	20	5.75	14	2	1.58	0	4.76	4.86	1.96
4	10	126	20	3.64	16	4	1.17	0	3.24	2.58	1.93
5	10	189	20	8.32	18	18	1.50	0	6.26	1.44	1.12
6	10	194	20	9.21	17	13	1.54	0	6.98	2.75	1.23
7	10	137	20	4.46	16	15	1.21	0	3.76	1.25	1.41
8	10	307	20	12.23	18	18	2.85	0	9.50	2.83	1.25
9	10	165	20	7.20	16	13	1.16	0	5.50	3.04	1.37
10	10	156	20	4.12	4	0	1.69	0	3.66	4.17	2.28
11	20	421	40	29.34	14	0	11.67	2	20.42	29.04	2.10
12	20	441	40	36.81	16	8	4.77	0	22.92	37.67	1.79
13	20	466	40	66.61	21	13	4.12	0	27.62	20.22	0.78
14	20	419	40	48.33	10	3	4.27	0	22.44	39.45	1.37
15	20	477	40	64.52	5	4	4.45	0	23.11	41.82	1.08
16	20	498	40	83.13	1	0	29.14	5	29.37	82.78	1.71
17	20	536	40	60.18	10	0	22.49	3	35.00	59.57	1.97
18	20	430	40	26.83	15	1	15.31	3	26.67	26.98	2.62
19	20	367	40	17.09	12	1	3.56	0	18.59	28.61	3.03
20	20	412	40	59.90	10	0	3.94	0	24.45	59.79	1.47
21	30	791	60	229.94	6	0	25.61	2	79.83	229.94	1.46
22	30	730	60	199.66	0	0	7.98	0	58.96	199.66	1.34
23	30	825	60	225.36	0	0	7.53	0	74.11	225.36	1.36
24	30	828	60	93.40	1	0	16.69	1	72.09	93.40	1.95
25	30	827	60	197.69	9	1	6.63	0	85.45	180.14	1.38
26	30	826	60	528.42	5	0	7.67	0	76.96	528.42	1.16
27	30	920	60	336.27	0	0	8.98	0	87.86	336.27	1.29
28	30	779	60	128.48	1	0	8.61	0	61.79	128.48	1.55
29	30	832	60	178.64	1	0	8.47	0	72.40	178.64	1.45
30	30	820	60	172.71	0	0	8.07	0	68.87	172.71	1.45
31	40	1195	80	802.56	0	0	23.11	1	150.41	802.56	1.22
32	40	1228	80	627.10	1	0	35.10	2	158.34	627.10	1.31
33	40	1199	80	636.39	0	0	12.67	0	163.54	636.39	1.28
34	40	1150	80	678.80	2	0	11.87	0	141.68	678.80	1.23
35	40	1392	80	1186.35	0	0	12.41	0	212.81	1,186.35	1.19
36	40	1297	80	4861.33	0	0	69.00	4	174.12	4,861.33	1.05
37	40	1150	80	685.53	2	0	11.12	0	144.39	685.53	1.23
38	40	1455	80	926.16	3	0	14.61	0	227.15	926.16	1.26
39	40	1257	80	610.34	0	0	23.16	1	170.49	610.34	1.32
40	40	1221	80	1484.83	0	0	12.78	0	160.30	1,484.83	1.12
41	50	1713	100	7368.55	0	0	19.20	0	341.33	7,368.54	1.05
42	50	1788	100	7282.17	0	0	53.69	2	378.85	7,282.17	1.06
43	50	2101	100	3981.00	0	0	25.23	0	501.26	3,981.00	1.13
44	50	1695	100	7364.34	0	0	90.35	4	323.94	7,364.34	1.06
45	50	1856	100	7397.04	0	0	19.40	0	373.61	7,397.04	1.05
46	50	1866	100	5948.14	3	0	61.15	2	415.09	5,948.14	1.08
47	50	1811	100	7376.13	0	0	93.49	4	353.10	7,376.13	1.06
48	50	1914	100	7402.76	0	0	19.32	0	401.34	7,402.76	1.06
49	50	1850	100	5292.99	0	0	97.54	4	378.35	5,292.99	1.09
50	50	1744	100	7376.88	0	0	50.24	2	343.70	7,376.88	1.05

Table 10.2: Strong Fixing with upper bound from relaxed SCP, $\alpha = 12\%$

#	$\|\mathcal{V}(H)\|$	$\|\mathcal{I}(H)\|$	n	`Gurobi` original time	Max Fix	n_{SF}^0	upper bound time	infeasible relaxed SCP	strong fixing time	`Gurobi` reduced time	total time factor
1	10	161	20	6.58	15	11	1.13	0	4.76	3.06	1.36
2	10	135	20	3.46	13	5	1.19	0	3.25	2.49	1.97
3	10	164	20	5.75	14	8	1.29	0	4.75	3.19	1.62
4	10	126	20	3.64	16	10	1.00	0	3.26	1.62	1.62
5	10	189	20	8.32	18	18	1.50	0	6.31	1.44	1.12
6	10	194	20	9.21	17	17	1.45	0	6.93	1.39	1.07
7	10	137	20	4.46	16	16	1.12	0	3.80	1.09	1.36
8	10	307	20	12.23	18	18	2.85	0	9.55	2.86	1.26
9	10	165	20	7.20	16	15	1.19	0	5.47	2.23	1.26
10	10	156	20	4.12	4	0	1.68	0	3.69	4.17	2.29
11	20	421	40	29.34	14	0	15.41	3	20.49	29.04	2.24
12	20	441	40	36.81	16	15	4.16	0	22.88	14.71	1.15
13	20	466	40	66.61	21	14	8.25	1	27.70	18.85	0.83
14	20	419	40	48.33	10	3	4.30	0	22.36	39.71	1.38
15	20	477	40	64.52	5	4	4.45	0	22.96	41.95	1.08
16	20	498	40	83.13	1	0	39.04	7	29.21	82.78	1.82
17	20	536	40	60.18	10	0	5.13	0	35.12	59.57	1.68
18	20	430	40	26.83	15	1	18.96	4	26.14	26.99	2.74
19	20	367	40	17.09	12	1	3.53	0	18.52	28.52	3.02
20	20	412	40	59.90	10	0	3.80	0	23.47	59.79	1.46
21	30	791	60	229.94	6	0	8.58	0	79.97	229.94	1.39
22	30	730	60	199.66	0	0	14.52	1	59.13	199.66	1.37
23	30	825	60	225.36	0	0	15.43	1	74.31	225.36	1.40
24	30	828	60	93.40	1	0	24.22	2	72.28	93.40	2.03
25	30	827	60	197.69	9	1	13.19	1	85.59	179.77	1.41
26	30	826	60	528.42	5	0	14.57	1	76.28	528.42	1.17
27	30	920	60	336.27	0	0	16.37	1	88.18	336.27	1.31
28	30	779	60	128.48	1	0	8.57	0	62.31	128.48	1.55
29	30	832	60	178.64	1	0	16.50	1	71.20	178.64	1.49
30	30	820	60	172.71	0	0	8.06	0	68.06	172.71	1.44
31	40	1195	80	802.56	0	0	34.10	2	150.01	802.56	1.23
32	40	1228	80	627.10	1	0	47.28	3	158.53	627.10	1.33
33	40	1199	80	636.39	0	0	46.91	3	162.24	636.39	1.33
34	40	1150	80	678.80	2	0	11.96	0	143.23	678.80	1.23
35	40	1392	80	1186.35	0	0	39.32	2	215.08	1,186.35	1.21
36	40	1297	80	4861.33	0	0	95.40	6	171.86	4,861.33	1.05
37	40	1150	80	685.53	2	0	11.05	0	144.97	685.53	1.23
38	40	1455	80	926.16	3	0	27.23	1	225.57	926.16	1.27
39	40	1257	80	610.34	0	0	12.41	0	173.73	610.34	1.30
40	40	1221	80	1484.83	0	0	12.59	0	161.51	1,484.83	1.12
41	50	1713	100	7368.55	0	0	33.64	1	340.38	7,368.54	1.05
42	50	1788	100	7282.17	0	0	68.06	3	377.59	7,282.17	1.06
43	50	2101	100	3981.00	0	0	64.37	2	497.83	3,981.00	1.14
44	50	1695	100	7364.34	0	0	104.83	5	322.81	7,364.34	1.06
45	50	1856	100	7397.04	0	0	20.74	0	376.95	7,397.04	1.05
46	50	1866	100	5948.14	3	0	100.71	4	413.67	5,948.14	1.09
47	50	1811	100	7376.13	0	0	108.64	5	350.17	7,376.13	1.06
48	50	1914	100	7402.76	0	0	19.06	0	394.92	7,402.76	1.06
49	50	1850	100	5292.99	0	0	111.52	5	377.79	5,292.99	1.09
50	50	1744	100	7376.88	0	0	81.96	4	343.84	7,376.88	1.06

Table 10.3: Strong Fixing with upper bound from relaxed SCP, $\alpha = 14\%$

#	$\|\mathcal{V}(H)\|$	$\|\mathcal{I}(H)\|$	n	Gurobi original time	Max Fix	n^0_{SF}	upper bound time	infeasible relaxed SCP	strong fixing time	Gurobi reduced time	total time factor
1	10	161	20	6.58	15	15	1.45	0	4.71	1.77	1.36
2	10	135	20	3.46	13	5	1.18	0	3.24	2.48	1.97
3	10	164	20	5.75	14	8	1.29	0	4.73	3.19	1.62
4	10	126	20	3.64	16	10	1.00	0	3.25	1.61	1.62
5	10	189	20	8.32	18	18	1.50	0	6.32	1.45	1.12
6	10	194	20	9.21	17	17	1.46	0	6.97	1.43	1.07
7	10	137	20	4.46	16	16	1.12	0	3.82	1.09	1.36
8	10	307	20	12.23	18	18	2.85	0	9.66	3.02	1.26
9	10	165	20	7.20	16	15	1.24	0	5.61	2.30	1.26
10	10	156	20	4.12	4	0	3.10	1	3.79	4.17	2.29
11	20	421	40	29.34	14	0	18.50	4	19.97	29.04	2.24
12	20	441	40	36.81	16	15	4.06	0	22.71	14.32	1.15
13	20	466	40	66.61	21	14	12.27	2	27.29	18.25	0.83
14	20	419	40	48.33	10	3	4.17	0	21.84	39.12	1.38
15	20	477	40	64.52	5	4	4.50	0	22.64	41.40	1.08
16	20	498	40	83.13	1	0	42.32	8	29.03	82.78	1.82
17	20	536	40	60.18	10	0	5.14	0	34.76	59.57	1.68
18	20	430	40	26.83	15	1	21.89	5	25.59	26.20	2.74
19	20	367	40	17.09	12	1	3.42	0	17.97	28.24	3.02
20	20	412	40	59.90	10	0	7.65	1	23.30	59.79	1.46
21	30	791	60	229.94	6	2	7.25	0	78.65	166.23	1.39
22	30	730	60	199.66	0	0	21.34	2	58.43	199.66	1.37
23	30	825	60	225.36	0	0	22.59	2	73.71	225.36	1.40
24	30	828	60	93.40	1	0	39.67	4	71.36	93.40	2.03
25	30	827	60	197.69	9	1	25.57	3	84.33	178.62	1.41
26	30	826	60	528.42	5	0	22.31	2	77.17	528.42	1.17
27	30	920	60	336.27	0	0	24.47	2	88.10	336.27	1.31
28	30	779	60	128.48	1	0	16.65	1	61.82	128.48	1.55
29	30	832	60	178.64	1	0	15.77	1	72.89	178.64	1.49
30	30	820	60	172.71	0	0	30.38	3	68.56	172.71	1.44
31	40	1195	80	802.56	0	0	55.45	4	150.35	802.56	1.23
32	40	1228	80	627.10	1	0	70.32	5	158.92	627.10	1.33
33	40	1199	80	636.39	0	0	57.44	4	161.91	636.39	1.33
34	40	1150	80	678.80	2	0	34.28	2	142.84	678.80	1.23
35	40	1392	80	1186.35	0	0	51.89	3	213.68	1,186.35	1.21
36	40	1297	80	4861.33	0	0	109.25	7	174.09	4,861.33	1.05
37	40	1150	80	685.53	2	0	22.52	1	146.71	685.53	1.23
38	40	1455	80	926.16	3	0	41.39	2	229.40	926.16	1.27
39	40	1257	80	610.34	0	0	24.44	1	173.33	610.34	1.30
40	40	1221	80	1484.83	0	0	23.69	1	163.71	1,484.83	1.12
41	50	1713	100	7368.55	0	0	64.92	3	340.92	7,368.54	1.05
42	50	1788	100	7282.17	0	0	82.52	4	378.82	7,282.17	1.06
43	50	2101	100	3981.00	0	0	87.55	3	501.79	3,981.00	1.14
44	50	1695	100	7364.34	0	0	122.68	6	322.54	7,364.34	1.06
45	50	1856	100	7397.04	0	0	38.10	1	377.87	7,397.04	1.05
46	50	1866	100	5948.14	3	3	40.45	1	414.17	2,884.48	0.56
47	50	1811	100	7376.13	0	0	129.80	6	355.23	7,376.13	1.06
48	50	1914	100	7402.76	0	0	56.74	2	399.00	7,402.76	1.06
49	50	1850	100	5292.99	0	0	150.45	7	382.18	5,292.99	1.09
50	50	1744	100	7376.88	0	0	99.41	5	347.00	7,376.88	1.06

Table 10.4: Strong Fixing with upper bound from relaxed SCP, $\alpha = 16\%$

#	$\lvert\mathcal{V}(H)\rvert$	$\lvert\mathcal{I}(H)\rvert$	n	Gurobi original time	Max Fix	n^0_{SF}	upper bound time	infeasible relaxed SCP	strong fixing time	Gurobi reduced time	total time factor
1	10	161	20	6.58	15	15	2.66	1	4.83	1.83	1.41
2	10	135	20	3.46	13	5	1.22	0	3.32	2.55	2.01
3	10	164	20	5.75	14	8	1.32	0	4.84	3.30	1.66
4	10	126	20	3.64	16	10	1.00	0	3.25	1.61	1.62
5	10	189	20	8.32	18	18	1.54	0	6.30	1.45	1.12
6	10	194	20	9.21	17	17	1.47	0	6.99	1.40	1.08
7	10	137	20	4.46	16	16	1.15	0	3.82	1.09	1.37
8	10	307	20	12.23	18	18	2.86	0	9.65	2.88	1.27
9	10	165	20	7.20	16	15	2.25	1	5.49	2.25	1.41
10	10	156	20	4.12	4	0	4.41	2	3.70	4.17	2.94
11	20	421	40	29.34	14	0	21.80	5	19.97	29.04	2.44
12	20	441	40	36.81	16	15	4.05	0	22.27	14.30	1.11
13	20	466	40	66.61	21	15	16.59	3	27.14	17.15	0.92
14	20	419	40	48.33	10	3	8.28	1	21.77	39.06	1.44
15	20	477	40	64.52	5	4	4.50	0	22.52	41.44	1.06
16	20	498	40	83.13	1	0	46.16	9	28.95	82.78	1.91
17	20	536	40	60.18	10	0	16.06	2	35.34	59.57	1.86
18	20	430	40	26.83	15	1	25.40	6	26.26	26.90	2.98
19	20	367	40	17.09	12	1	3.53	0	18.30	28.43	3.00
20	20	412	40	59.90	10	0	4.40	0	24.54	59.79	1.48
21	30	791	60	229.94	6	2	15.12	1	79.35	166.55	1.14
22	30	730	60	199.66	0	0	28.46	3	58.21	199.66	1.43
23	30	825	60	225.36	0	0	29.12	3	73.68	225.36	1.46
24	30	828	60	93.40	1	0	47.20	5	71.24	93.40	2.27
25	30	827	60	197.69	9	1	33.28	4	85.02	179.21	1.50
26	30	826	60	528.42	5	0	35.26	4	76.72	528.42	1.21
27	30	920	60	336.27	0	0	31.49	3	87.10	336.27	1.35
28	30	779	60	128.48	1	0	32.99	3	60.98	128.48	1.73
29	30	832	60	178.64	1	0	22.70	2	71.67	178.64	1.53
30	30	820	60	172.71	0	0	36.66	4	67.70	172.71	1.60
31	40	1195	80	802.56	0	0	67.84	5	152.66	802.56	1.27
32	40	1228	80	627.10	1	0	82.07	6	159.80	627.10	1.39
33	40	1199	80	636.39	0	0	67.09	5	164.01	636.39	1.36
34	40	1150	80	678.80	2	0	46.38	3	146.07	678.80	1.28
35	40	1392	80	1186.35	0	0	67.20	4	214.89	1,186.35	1.24
36	40	1297	80	4861.33	0	0	119.98	8	171.59	4,861.33	1.06
37	40	1150	80	685.53	2	0	43.83	3	144.30	685.53	1.27
38	40	1455	80	926.16	3	0	53.08	3	225.73	926.16	1.30
39	40	1257	80	610.34	0	0	35.31	2	172.12	610.34	1.34
40	40	1221	80	1484.83	0	0	34.23	2	162.69	1,484.83	1.13
41	50	1713	100	7368.55	0	0	77.07	4	339.56	7,368.54	1.06
42	50	1788	100	7282.17	0	0	98.18	5	377.48	7,282.17	1.07
43	50	2101	100	3981.00	0	0	105.15	4	497.42	3,981.00	1.15
44	50	1695	100	7364.34	0	0	138.32	7	322.49	7,364.34	1.06
45	50	1856	100	7397.04	0	0	54.88	2	376.01	7,397.04	1.06
46	50	1866	100	5948.14	3	3	55.54	2	412.84	2,886.73	0.56
47	50	1811	100	7376.13	0	0	143.94	7	352.49	7,376.13	1.07
48	50	1914	100	7402.76	0	0	74.15	3	401.86	7,402.76	1.06
49	50	1850	100	5292.99	0	0	169.63	8	382.99	5,292.99	1.10
50	50	1744	100	7376.88	0	0	116.59	6	348.81	7,376.88	1.06

Table 10.5: Strong Fixing with upper bound from relaxed SCP, $\alpha = 18\%$

#	$\lvert\mathcal{V}(H)\rvert$	$\lvert\mathcal{I}(H)\rvert$	n	`Gurobi` original time	Max Fix	n^0_{SF}	upper bound time	infeasible relaxed SCP	strong fixing time	`Gurobi` reduced time	total time factor
1	10	161	20	6.58	15	15	3.72	2	4.73	1.78	1.55
2	10	135	20	3.46	13	5	1.19	0	3.26	2.48	1.97
3	10	164	20	5.75	14	8	2.45	1	4.74	3.20	1.82
4	10	126	20	3.64	16	10	1.00	0	3.30	1.62	1.63
5	10	189	20	8.32	18	18	1.50	0	6.30	1.45	1.12
6	10	194	20	9.21	17	17	1.47	0	7.07	1.45	1.09
7	10	137	20	4.46	16	16	1.17	0	3.82	1.09	1.38
8	10	307	20	12.23	18	18	2.86	0	9.60	2.85	1.26
9	10	165	20	7.20	16	15	3.31	2	5.51	2.25	1.56
10	10	156	20	4.12	4	0	5.76	3	3.69	4.17	3.26
11	20	421	40	29.34	14	0	26.12	6	20.51	29.04	2.61
12	20	441	40	36.81	16	15	4.28	0	22.97	14.51	1.15
13	20	466	40	66.61	21	15	21.67	4	27.93	17.59	1.01
14	20	419	40	48.33	10	3	11.94	2	22.31	39.64	1.54
15	20	477	40	64.52	5	4	8.91	1	23.18	41.91	1.15
16	20	498	40	83.13	1	0	51.65	10	29.04	82.78	1.97
17	20	536	40	60.18	10	0	20.63	3	35.44	59.57	1.94
18	20	430	40	26.83	15	1	28.69	7	25.92	26.84	3.09
19	20	367	40	17.09	12	1	7.03	1	18.32	28.63	3.22
20	20	412	40	59.90	10	0	4.68	0	24.30	43.59	1.21
21	30	791	60	229.94	6	2	22.66	2	79.18	166.55	1.17
22	30	730	60	199.66	0	0	34.84	4	58.29	199.66	1.47
23	30	825	60	225.36	0	0	35.47	4	73.39	225.36	1.48
24	30	828	60	93.40	1	0	54.76	6	71.46	93.40	2.35
25	30	827	60	197.69	9	1	39.06	5	84.90	178.72	1.53
26	30	826	60	528.42	5	0	42.51	5	77.04	528.42	1.23
27	30	920	60	336.27	0	0	39.55	4	87.57	336.27	1.38
28	30	779	60	128.48	1	0	40.76	4	62.10	128.48	1.80
29	30	832	60	178.64	1	0	29.90	3	73.00	178.64	1.58
30	30	820	60	172.71	0	0	44.58	5	68.76	172.71	1.66
31	40	1195	80	802.56	0	0	66.63	6	151.60	802.56	1.27
32	40	1228	80	627.10	1	0	93.16	7	161.09	627.10	1.41
33	40	1199	80	636.39	0	0	78.05	6	164.89	636.39	1.38
34	40	1150	80	678.80	2	0	57.02	4	146.52	678.80	1.30
35	40	1392	80	1186.35	0	0	79.62	5	215.82	1,186.35	1.25
36	40	1297	80	4861.33	0	0	133.83	9	173.47	4,861.33	1.06
37	40	1150	80	685.53	2	0	54.08	4	145.77	685.53	1.29
38	40	1455	80	926.16	3	0	65.84	4	225.60	926.16	1.31
39	40	1257	80	610.34	0	0	45.93	3	171.25	610.34	1.36
40	40	1221	80	1484.83	0	0	43.54	3	162.22	1,484.83	1.14
41	50	1713	100	7368.55	0	0	93.18	5	343.77	7,368.54	1.06
42	50	1788	100	7282.17	0	0	112.68	6	375.14	7,282.17	1.07
43	50	2101	100	3981.00	0	0	123.15	5	498.59	3,981.00	1.16
44	50	1695	100	7364.34	0	0	155.31	8	321.52	7,364.34	1.06
45	50	1856	100	7397.04	0	0	71.09	3	373.96	7,397.04	1.06
46	50	1866	100	5948.14	3	3	73.14	3	409.67	2,887.85	0.57
47	50	1811	100	7376.13	0	0	158.53	8	352.21	7,376.13	1.07
48	50	1914	100	7402.76	0	0	91.18	4	401.03	7,402.76	1.07
49	50	1850	100	5292.99	0	0	183.95	9	380.79	5,292.99	1.11
50	50	1744	100	7376.88	0	0	128.68	7	343.59	7,376.88	1.06

Table 10.6: Strong Fixing with upper bound from relaxed SCP, $\alpha = 20\%$

Chapter 11

Final Remarks

Extensions for SCP. The integer-programming approach can be extended to other geometric settings. We can use other metrics, or replace balls $B(x, r(x))$ with arbitrary convex sets $B(x)$ for which we can compute the intersection of with each edge $I \in \mathcal{I}(G)$. In this way, N is just an index set, and we only need a pair of line searches to determine the endpoints of the intersection of $B(x)$ with each $I \in \mathcal{I}(G)$. We still have $|\mathcal{C}(I)| \leq 1+2n$, for each $I \in \mathcal{I}(G)$, and so the number of covering constraints in SCP is at most $(1+2n)|\mathcal{I}(G)|$. Finally, we could take G to be geodesically embedded on a sphere and the balls replaced by geodesic balls.

Strong fixing potential. The results from Chapter 10 demonstrate that it is possible to apply a reduction procedure based on strong fixing to nonlinear integer problems, using a heuristic solution value as an upper bound. As observed in all previous chapters Chapter 6, Chapter 9 and Chapter 10, for instances where variable fixing was successful, there was a significant reduction in the total solving time for the reduced version of the problem. Based on the results in Table 6.1, we can also conclude that a more rigorous implementation of the strong fixing step could yield even greater performance improvements.

An important direction is to reduce the time for strong fixing, so as to get a large number of variables fixed without solving all of the fixing subproblems. Additionally, the strong- fixing methodology is very general and could work well for other classes of mixed-integer optimization problems. It is still challenging to apply the strong fixing methodology to more complex problems, but the results show that it is feasible and leads to significant performance gains. We can conclude that, with a more efficient implementation and a good strategy to obtain a high-quality upper bound (e.g., with more efficient heuristics), strong fixing could become a standard pre-processing tool for mixed-integer optimization problems.

References

[1] BODIE, Z., KANE, A., MARCUS, A. J., 2014, *Investments*. 10th ed. New York, McGraw-Hill Education. ISBN: 9780077861674.

[2] CERULLI, M., D'AMBROSIO, C., LIBERTI, L., PELEGRÍN, M., 2021, "Detecting and solving aircraft conflicts using bilevel programming", *Journal of Global Optimization*, v. 81, n. 2, pp. 529–557. https://doi.org/10.1007/s10898-021-00997-1.

[3] CROWDER, H., JOHNSON, E. L., PADBERG, M., 1983, "Solving large-scale zero-one linear programming problems", *Operations Research*, v. 31, n. 5, pp. 803–834. https://doi.org/10.1287/opre.31.5.803.

[4] D'AMBROSIO, C., FAMPA, M., LEE, J., SINNECKER, F., 2024, "On a geometric graph-covering problem related to optimal safety-landing-site location". In: *Combinatorial Optimization*, v. 14594, *Lecture Notes in Computer Science*, Springer, pp. 16–29. https://doi.org/10.1007/978-3-031-60924-4_2.

[5] DELAUNAY, B., 1934, "Sur la sphère vide", *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, v. 6, pp. 793–800.

[6] EDELSBRUNNER, H., TAN, T. S., WAUPOTITSCH, R., 1992, "An $O(n^2 \log n)$ time algorithm for the minmax angle triangulation", *SIAM Journal on Scientific and Statistical Computing*, v. 13, n. 4, pp. 994–1008. https://doi.org/10.1137/0913058.

[7] EXACTITUDE CONSULTANCY, 2024. "Urban Air Mobility Market By Component, By Platform Operation, By Range, By Platform Architecture, By End User and Region, Global Trends and Forecast from 2023 to 2029". https://exactitudeconsultancy.com/reports/19994/urban-air-mobility-market/.

[8] FAA, 2023. "Urban Air Mobility (UAM) Concept of Operations, version 2.0". https://www.faa.gov/air-taxis/uam_blueprint.

[9] HAMMING, R. W., 1950, “Error detecting and error correcting codes”, *The Bell System Technical Journal*, v. 29, n. 2, pp. 147–160. https://doi.org/10.1002/j.1538-7305.1950.tb00463.x.

[10] JACCARD, P., 1912, “The distribution of the flora of the alpine zone”. In: *New Phytologist*, v. 11, pp. 37–50, February. http://dx.doi.org/10.1111/j.1469-8137.1912.tb05611.x.

[11] KARP, R. M., 1972, “Reducibility among Combinatorial Problems”. In: Miller, R. E., Thatcher, J. W., Bohlinger, J. D. (Eds.), *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pp. 85–103, Boston, MA, Springer US. ISBN: 978-1-4684-2001-2. https://doi.org/10.1007/978-1-4684-2001-2_9.

[12] NASA, 2021. “UAM Vision Concept of Operations (ConOps) UAM Maturity Level (UML) 4”. https://ntrs.nasa.gov/citations/20205011091.

[13] PELEGRÍN, M., D’AMBROSIO, C., 2022, “Aircraft deconfliction via mathematical programming: review and insights”, *Transportation Science*, v. 56, n. 1, pp. 118–140. https://doi.org/10.1287/trsc.2021.1056.

[14] PELEGRÍN, M., XU, L., 2023, “Continuous covering on networks: Improved mixed integer programming formulations”, *Omega*, v. 117, pp. 102835. https://doi.org/10.1016/j.omega.2023.102835.

[15] PELEGRÍN, M., D’AMBROSIO, C., DELMAS, R., HAMADI, Y., 2023, “Urban air mobility: from complex tactical conflict resolution to network design and fairness insights”, *Optimization Methods and Software*, v. 38, n. 6, pp. 1131–1343. https://doi.org/10.1080/10556788.2023.2241148.

[16] RYOO, H. S., SAHINIDIS, N. V., 1996, “A branch-and-reduce approach to global optimization”, *Journal of Global Optimization*, v. 8, pp. 107–138. https://doi.org/10.1007/BF00138689.

[17] SERHAL, K., MACIAS, J. J. E., ANGELOUDIS, P., 2021. “Demand modelling and optimal vertiport placement for airport-purposed eVTOL services”. https://optimization-online.org/?p=18217.

[18] VILLA, A., 2020. "Hub Location Problem For Optimal Vertiport Selection In Urban Air Mobility — Chicago Case Study". Masters Thesis, Politecnico di Milano, https://doi.org/10.25417/uic.22226494.v1.

[19] XIA, G., 2013, "The stretch factor of the Delaunay triangulation is less than 1.998", *SIAM Journal on Computing*, v. 42, n. 4, pp. 1620–1659. https://doi.org/10.1137/110832458.

[20] XU, L., 2021, *Optimal location of safety landing sites.* Research Report, LIX, École Polytechnique. https://hal.science/hal-03286640.