



UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Petra Maria Bartmeyer

A new quadratic relaxation for binary variables with applications to the reliability of electrical distribution networks, segmentation of nerve root images, and distance geometry problems

Uma nova relaxação quadrática para variáveis binárias com aplicações a confiabilidade de redes de energia elétrica, a segmentação de imagens médicas de raízes nervosas e a problemas de geometria de distâncias

Campinas
2020

Petra Maria Bartmeyer

A new quadratic relaxation for binary variables with applications to the reliability of electrical distribution networks, segmentation of nerve root images, and distance geometry problems

Uma nova relaxação quadrática para variáveis binárias com aplicações a confiabilidade de redes de energia elétrica, a segmentação de imagens médicas de raízes nervosas e a problemas de geometria de distâncias

Thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering, in the area of Automation.

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutora em Engenharia Elétrica, na área de Automação.

Orientador: Prof. Christiano Lyra Filho

Este exemplar corresponde à versão final da tese defendida pela aluna Petra Maria Bartmeyer e orientada pelo Prof. Christiano Lyra Filho

Campinas
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

Bartmeyer, Petra Maria, 1990-
B284n A new quadratic relaxation for binary variables with applications to the reliability of electrical distribution networks, segmentation of nerve root images, and distance geometry problems / Petra Maria Bartmeyer. – Campinas, SP : [s.n.], 2020.

Orientador: Christiano Lyra Filho.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Otimização. 2. Redes elétricas. 3. Interpretação de imagens. 4. Geometria de distâncias. I. Lyra Filho, Christiano, 1951-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Uma nova relaxação quadrática para variáveis binárias com aplicações a confiabilidade de redes de energia elétrica, a segmentação de imagens médicas e a problemas de geometria de distâncias

Palavras-chave em inglês:

Optimization

Electrical networks

Image interpretation

Distance geometry

Área de concentração: Automação

Titulação: Doutora em Engenharia Elétrica

Banca examinadora:

Christiano Lyra Filho [Orientador]

Luciana Salete Buriol

André Carlos Ponce de Leon Ferreira de Carvalho

Paulo Augusto Valente Ferreira

Fábio Luiz Usberti

Data de defesa: 27-04-2020

Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-9081-7879>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3505800978965858>

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidata: Petra Maria Bartmeyer

RA: 160903

Data da Defesa: 27 de abril de 2020

Título da Tese: "***A new quadratic relaxation for binary variables with applications to the reliability of electrical distribution networks, segmentation of nerve root images, and distance geometry problems***".

Prof. Dr. Christiano Lyra Filho (Presidente)

Prof. Dra. Luciana Salete Buriol

Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho

Prof. Dr. Paulo Augusto Valente Ferreira

Prof. Dr. Fábio Luiz Usberti

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós Graduação da Faculdade de Engenharia Elétrica e de Computação.

Acknowledgements

I owe my deepest gratitude to my advisor Prof. Christiano Lyra for the friendship, patience, motivation, enthusiasm, and support during these four years. Prof. Christiano, thank you for guiding my scientific and personal development - I could not have imagined having a better advisor and mentor.

I am grateful to Caio dos Santos and Prof. Fernanda Arioli for the work and contributions delivered during the explorations in electrical engineering applications.

I would like to show my gratitude to Prof Leif Havton for the invitation and the support during the period that I spent at the University of California - Los Angeles. I am also grateful to Natalia Biscola for introduced me to the neuroscience field and the supporting friendship.

I wish to show my appreciation for the help in generating instances and the conversations about the unassigned Distance Geometry Problem delivered by Prof. Carlile Lavor.

I am indebted to the friends that I made during my undergraduate years; I thank for the inspired conversations, knowledge, and emotional support without which this thesis would not have been possible. For my family, I cannot find words to express my gratitude for supporting me throughout my graduation years; your endless belief in my capacities was essential.

I also like to thank Prof. Fernando Von Zuben for the kindness in receiving my colleagues and me in the Lbic during this last year; it was a valuable time. I am grateful for the friends that I made and the knowledge that I gained because of it.

This research received support from CNPq – Brazilian National Council for Scientific and Technological Development. It was also supported by FAEPEX – Fund for Support to Teaching, Research, and Outreach Activities - UNICAMP. The period at the "University of California - Los Angeles" was supported by UCLA-Health.

Resumo

Como o título sugere, o foco desta pesquisa é o desenvolvimento de uma nova relaxação quadrática para problemas binários, sua formalização em resultados teóricos, e a aplicação dos novos conceitos em aplicações a confiabilidade de redes de energia elétrica, a segmentação de imagens médicas de nervos e a problemas de geometria de distâncias. Modelos matemáticos contendo variáveis de decisões binárias podem ser usados para encontrar as melhores soluções em processos de tomada de decisões, normalmente caracterizando problemas de otimização combinatória difíceis. A solução desses problemas em aplicações de interesse prático requer um grande esforço computacional; por isso, ao longo dos últimos anos, têm sido objeto de pesquisas na área de metaheurísticas. As ideias aqui desenvolvidas abrem novas perspectivas para a abordagem desses problemas apoiando-se em métodos de otimização não-lineares, área que vem sendo povoada por "solvers" muito eficientes.

Inicialmente, explorando aspectos formais, a relaxação desenvolvida é particularizada para um problema de otimização quadrática binária irrestrita. O relaxamento permite o desenvolvimento de três estruturas para abordar esta classe de problemas, e explora a convexidade da função objetivo para obter melhorias computacionais. Estudos de casos compararam o relaxamento proposto com os relaxamentos similares apresentados na literatura.

Foram desenvolvidas três aplicações para os desenvolvimentos teóricos da pesquisa. A primeira aplicação envolve a melhoria da confiabilidade de redes de energia elétrica. Especificamente, aborda o problema de definir a melhor alternativa para a alocação de sensores na rede, o que permite reduzir os efeitos de ocorrências indesejáveis e ampliar a resiliência das redes. A segunda aplicação envolve o problema de segmentação de imagens médicas associadas a estruturas de nervos. A abordagem proposta interpreta o problema de segmentação como um problema de otimização binária, onde medir cada axônio significa encontrar um ciclo Hamiltoniano, um caso do problema do caixeiro viajante; a solução desses problemas fornece a estatística descritiva para um conjunto de axônios, incluindo

o número (de axônios), os diâmetros e as áreas ocupadas. A última aplicação elabora um modelo matemático para o problema de geometria de distâncias sem designação, área ainda pouco estudada e com muitos aspectos em aberto. A relaxação desenvolvida na pesquisa permitiu resolver instâncias com mais de vinte mil variáveis binárias. Esses resultados são bons indicadores dos benefícios alcançáveis com os aspectos teóricos da pesquisa, e abrem novas perspectivas para as aplicações, que incluem inovações em nanotecnologia e bioengenharia.

Palavras-chave: Relaxação binária, otimização quadrática binária irrestrita, funções quadráticas, confiabilidade de redes de energia elétrica, interpretação de imagens médicas, problema de geometria de distâncias.

Abstract

As the title suggests, the focus of this research is the development of a new quadratic relaxation for binary problems, its formalization in theoretical results, and the application of the new concepts in applications to the reliability of electric power networks, segmentation of nerve root images, and distance geometry problems. Mathematical models with binary decision variables can be used to find the best solutions for the decision-making process, usually leading to difficult combinatorial optimization problems. The solution to these problems in practical applications requires a high computational effort; therefore, over the past years, it has been the subject of research in the area of metaheuristics. The ideas developed in this thesis open new perspectives for addressing these problems using nonlinear optimization approaches, an area that has been populated by very efficient solvers.

The initial developments explore the formal aspects of the relaxation in the context of a quadratic unconstrained binary optimization problem. The use of the proposed relaxation allows to create three structures to deal with this class of problems and explores the objective function convexity to improve the computational performance. Case studies compare the proposed relaxation with the previous relaxations proposed in the literature.

Three new applications were developed to explore the theoretical developments of this research. The first application concerns the improvement of the reliability of electric power distribution networks. Specifically, it deals with the problem of defining the best allocation for remote fault sensors, allowing to reduce the consequence of the faults and to improve the resilience of the networks. The second application explores the segmentation of medical images related to nerve root structures. The proposed approach regards the segmentation problem as a binary optimization problem, where measuring each axon is equivalent to finding a Hamiltonian cycle for a particular case of the traveling salesman problem; the solution to these problems provides the descriptive statistics of the axon set, including the number of axons, their diameters, and the area used by each axon.

The last application designs a mathematical model for the unassigned distance geometry problem, an incipient research area with many open problems. The relaxation developed in this research allowed to solve instances with more than twenty thousand binary variables. These results can be seen as good indicators of the benefits attainable with the theoretical aspects of the research and opens new perspectives for applications, which include innovations in nanotechnology and bio-engineering.

Key-words: Quadratic relaxation, quadratic unconstrained binary optimization, quadratic functions, reliability of power distribution networks, segmentation of nerve root images, distance geometry problems.

List of Figures

4.1	<i>Gap-to-target</i> plot for instances <i>beas100a</i> , <i>gka7c</i> , and <i>gka1d</i> , respectively.	37
4.2	Nonzeros entries of the instances <i>beas100a</i> (left), <i>gka7c</i> (center) e <i>gak1d</i> (right), where <i>nz</i> is the total number of nonzero entries.	38
4.3	<i>Gap-to-target</i> plot for P_{LC} with $\beta = 50$ (left) and QCQP (right) for instance <i>beas100a</i>	39
4.4	Values of the objective function, in logarithmic scale, with and without the parameter β calculated by Proposition (1) (With P1 and Without P1, respectively)	40
4.5	Performance profile of the gap for the instance of the test sets from Beasley (1998), Billionnet & Elloumi (2007), and Glover et al. (1998).	41
5.1	Iteration between sensor faults and the substation.	46
5.2	Topology and device configuration for the IEEE123 buses network. Image adapted from Alam et al. (2018).	53
5.3	Total improvement of the detection-time. The image comprises the number of indicators per recloser on the x-axis and the detection time on the y-axis. Image from Santos et al. (2019).	54
5.4	Position of the indicators for scenarios with two, three, eight, and twelve fault sensors per recloser. Image from Santos et al. (2019).	55
6.1	Representation of a nerve root connect to a spinal cord (left) and a cross section of a nerve root highlighting the axons in the structure (right).	58
6.2	A motoneuron with the myelin sheath (left) and an axon structure covered by myelin sheath, which can be recognized by as a dark contour presented on scope images (right).	59
6.3	The images on the left, in grayscale, were took using TEM microscope. The images on the right, in blue, are images from LM. Images adapted from Gerber et al. (2019) and Garcia et al. (2003) , respectively.	60

6.4	A manually segmentation of myelinated axons using ImageJ.	61
6.5	Illustration of a Closed Knight tour (left) and an Open Knight tour (right), for a 8×8 chessboard.	66
6.6	Image reduction – left 84.134 bytes and right 836 bytes.	68
6.7	Results for the image segmentation using Algorithm 3. The processing time required was about 40 minutes.	69
6.8	The left image provide the deficient conversion of the original image, where background pixels were labeled as a myelin. One the right image, there structures identified as an axon are presented in gray.	70
6.9	The left image provide the deficient conversion of the original image, where pixels of the myelin were not in white. One the right image, there structures identified as an axon are presented in gray.	70
6.10	Neighbor for the flood-fill algorithm considering eight neighbors.	71
6.11	Axons identified by the Algorithm 4, in less than 50 seconds.	73
7.1	Image adapted from the thesis of S.R. Gujarathi	79
7.2	Image adapted from Gujarathi et al. (2014).	83
7.3	Example of two protein conformation instances with 20 vertices (right) and 100 vertices (left), generated using the procedure described in (Lavor 2006)	86

List of Tables

4.1	Comparison between the average gap per class $100, 250, 500, 1000$, and 2500 from Beasley (1998).	36
4.2	Comparison between the average gap for classes a, c, d, e , and f from (Glover et al. 1998).	36
4.3	Comparison between the average gap for classes $100, 120.3, 120.8, 150.3$, and 150.8 from (Billionnet & Elloumi 2007).	37
6.1	Computational times for the Hamiltonian cycle identification.	67
6.2	Comparison values of the measurements to analyze the quality of the segmentation Algorithm 4.	75
7.1	Influence of the penalty parameter β in the convergence of the solution approach.	84
7.2	Computational results for the randomly generated instances.	84
7.3	Data about instances and solutions. Table from Bartmeyer & Lyra (2020). .	86

Nomenclature

β Penalty parameter for the Lagrangian relaxation

aDGP Assigned distance geometry problem

DSC Sørensen–Dice coefficient

FN False negative

FP False positive

KKT Karush-Kuhn-Tucker conditions

LM Light microscope

MTZ Miller–Tucker–Zemlin constraints

PPV Positive predict value

QPQC Quadratically constrained quadratic program

QUBO Quadratic unconstrained binary optimization

SIMP Solid isotropic material with penalization

TEM Transmission electron microscopy

TN True negative

TP True positive

TPR True positive rate

uDGP Unassigned distance geometry problem

Contents

1	Introduction	17
I	Theoretical results	19
2	Selected concepts in integer and quadratic optimizations	20
2.1	Introduction	20
2.2	Binary programming	21
2.3	Quadratic programming	23
2.3.1	Lagrangian duality and penalty methods	23
2.3.2	Quadratically constrained quadratic program	24
3	A new quadratic reformulation for binary problems	26
3.1	Introduction	26
3.2	The proposed quadratic relaxation	27
3.2.1	Matrix representation of the quadratic constraint	29
3.3	Discussion	31
4	Quadratic unconstrained binary optimization problem	32
4.1	Introduction	32
4.2	Lagrangian relaxation	34
4.3	Computational studies	35
4.3.1	Influence of the initial solution	36
4.3.2	Studies with positive semi-definite matrix	39

4.3.3	Lower bound quality — comparison with previous relaxations in the literature	40
4.4	Discussion	41
II	Applications	43
5	Fault sensor placement for temporary faults identification	44
5.1	Introduction	44
5.2	Mathematical model	47
5.2.1	Local search with a Monte Carlo simulation	51
5.3	Case study	53
5.4	Discussion	56
6	Automatic segmentation for myelinated axons	57
6.1	Introduction	58
6.2	Nerve root images	59
6.3	Strategies for myelinated axon segmentation	60
6.4	Approach One – A combinatorial optimization approach	62
6.4.1	Hamiltonian cycle	62
6.4.2	Steps for the image segmentation using Hamiltonian cycles	64
6.4.3	Computational studies	65
6.5	Approach Two – A flood-fill approach	69
6.5.1	Image preprocessing	69
6.5.2	Flood-fill algorithm	71
6.5.3	Image segmentation using the flood-sill algorithm	72
6.5.4	Computational studies	73
6.6	Discussion	76
7	Unassigned distance geometry problem	78
7.1	Introduction	79
7.2	The unassigned distance geometry problem	80
7.3	Mathematical model	81
7.4	Computational studies	82
7.4.1	Influence of the penalty parameter β	83
7.4.2	Influence of the multiplicity	84

7.4.3	Protein conformation	85
7.5	Discussion	87
III	Conclusion and Future works	88
8	Conclusions	89
Bibliography		92
A	Talks and papers	99
A.1	Conferences participation	99
A.2	Abstracts	99
A.3	Conference papers	100
A.4	Journal papers - published	100
A.5	Journal papers submitted and in preparation	101
A.6	Talks	101

CHAPTER 1

Introduction

The core of this thesis is the proposal of a new quadratic relaxation for binary variables. A set of theoretical results gives support for the proposed methodology, which is applied to innovative formulations for problems in the reliability of power distribution systems, medical image interpretation, and distance geometry problems.

The text is organized into three parts. The first part contains the main concepts of integer and quadratic programming that give the foundation of this research and the theoretical aspects of the relaxation. The second part contains applications of the relaxation for the placement of fault sensors, medical image segmentation, and geometry distance problems. The third part contains general discussions and conclusions.

The first part starts with an overview of the theoretical aspects of binary programming and quadratic programming in Chapter 2. The Lagrangian duality theory for quadratic programming and necessary and sufficient properties for global solutions of quadratic programming are presented. The chapter ends with theoretical aspects of the quadratically constrained quadratic program. Chapter 3 presents the proposed relaxation for binary variables. The first theoretical result proves that the constraint set used in the relaxation adds integrality property to the problem. The second result shows that a single constraint can replace the constraint set without entailing the loss of generality. The convexity properties of the single constraint are explored in the final part of the chapter.

Chapter 4 extends the theoretical results to the quadratic unconstrained binary optimization (QUBO) and uses the QUBO model to evaluate the quality of the proposed methodology. The first part of the chapter presents the theoretical results from the interaction between the

QUBO model formulation and the proposed relaxation. Next, numerical studies compare the proposed relaxation with previous relaxations on the literature. The chapter ends with evaluations of the relaxation robustness related to the initial point solution and the existence of elements in the diagonal of the objective function matrix.

Part II provides three applications split into three chapters. Chapter 5 takes advantage of the results from Chapter 4 to model and solve the problem of placing fault sensors in a distribution network. The problem is modeled as an integer linear programming problem where the objective function minimizes the detection time of the faults by the recloser. Since the faults have a stochastic behavior, a local search uses a Monte Carlo simulation to evaluate and improve the quality of the solutions provided by the optimization model. A case study explores the benefits and limitations of the proposed methodology.

Chapter 6 introduces a medical image segmentation application interpreted as a combinatorial optimization problem. This application was modeled using the Hamiltonian Cycle problem to avoid segmentation methods based on pattern recognition, such as machine learning techniques. The idea of using binary relaxation aims to overcome the NP-completeness nature of the Hamiltonian Cycle problem. However, even with the relaxation of the binary variables, this strategy requires a series of resizing steps, which may introduce noises that alter the quality of the image segmentation. The alternative to tackle the problem is the proposal of a flood-fill algorithm that requires a lighter image processing phase to handle this medical image segmentation problem.

Chapter 7 explores the unassigned distance geometry problem, one of the branches of the distance geometry field. This chapter offers a new mathematical formulation for problems using binary and continuous variables in a combination of linear and nonlinear constraint sets. The use of the proposed relaxation allows to tackle high-dimension instances. Computational studies explore the quality of the proposed model and the advantages of relaxing the binary variables.

In the final part of this thesis, Chapter 8 presents the main conclusions about the proposed methodology and the main contributions in each one of the applications.

Part I

Theoretical results

CHAPTER 2

Selected concepts in integer and quadratic optimizations

This chapter summarizes some of the classical results in optimization that provides the basis for the theoretical contributions in this thesis. It contains concepts about binary and quadratic programming, such as definitions, computational complexity, and solution approaches. Section 2.2 brings the concepts related to binary programming, including solution techniques using linearization and relaxation for binary variables. Section 2.3 summarizes concepts related to the formulation of quadratic programming problems, Lagrangian duality, penalty methods, and quadratically constrained quadratic programming.

2.1 Introduction

Considering the following optimization problem

$$\min \quad f(x)$$

$$\text{s.t.} \quad x \in \Omega,$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \in \mathbb{R}^n$, and $\Omega \subseteq \mathbb{R}^n$.

In the case where f is convex, the components of the vector x are continuous, and the constraint set Ω is a convex set, the optimization problem can usually be solved in polynomial time, using techniques such as the interior-point methods (Bazaraa & Shetty 2012).

However, problems with integer variables are non-convex, the search space is discrete, and the direct use of the methods developed for continuous optimization does not guarantee the achievement of global optimal solutions or even feasible solutions. In such cases,

global optimal solutions can be achieved with the combination of methods for continuous optimization and enumeration techniques ([Morrison et al. 2016](#)). Some of these approaches are the cutting plane technique and the branch-and-bound technique, but none of them is a polynomial-time algorithm ([Wolsey & Nemhauser 1999](#)). If the integer variables are in a binary domain such as $\{0, 1\}$ or $\{-1, 1\}$, the problem is characterized as a binary optimization problem, which in general is NP-hard ([Garey & Johnson 1979](#)). These aspects are discussed in the following section.

2.2 Binary programming

An optimization problem belongs to the integer programming class if the optimization variables x are defined in an integer domain. In the general case, integer programming problems are NP-hard; however, special cases are solvable in polynomial time, such as the matching problem and some knapsack problems, shortest path, max-flow ([Garey & Johnson 1979](#)). A subclass of integer programming is the binary problem, which can contain $\{0, 1\}$ or $\{-1, 1\}$ variables. Binary variables are employed to model decision, assignment, and location problems and they are frequently employed in engineering problems, such as optimization of the design of structures ([Stolpe & Sandal 2018](#)) and the modeling of distances in molecular structures ([Liberti et al. 2011](#)).

The nature of the objective function and the constraint set are the essential information for the decision about a suitable solution method for a binary optimization problem. When considering a linear objective function and linear constraints, the solution methods are frequently based on implicit enumeration techniques, where systematic evaluations create a solution tree to guide the search for the optimal solution. Another idea is the cutting-plane methods that proposes an equivalent continuous problem to replace the integer programming problem, by adding a suitable set of constraints to the continuous formulations. Each constraint added to the problem works as a valid inequality cutting the convex hull and improving the description of the polyhedron around the optimum. The combination of implicit enumeration and the cutting-plane strategies is frequent used in optimization solvers, because it improves the computational effort and the numerical stability.

The general case of nonlinear binary optimization problems is NP-hard due to both the nature of the constraint set and the binary variables. For the cases where the constraint set is composed by the product of variables there are linearization techniques to convert the product of two binary variables into a single new binary variable, creating a new optimization

problem with a linear constraint set and binary variables ([Liberti 2007](#), [Anthony et al. 2017](#), [Mallach 2018](#)).

An example of these strategies is the McCormick envelopes ([McCormick 1976](#)). The McCormick envelopes technique creates a new set of linear constraints to replace the product $x_i x_j$ by using the bounds of x_i ($\underline{x}_i, \bar{x}_i$) and x_j ($\underline{x}_j, \bar{x}_j$). The product is computed by a new variable $y_{ij} = x_i x_j$, and the following set of constraints is added to the problem.

$$\begin{aligned} y_{ij} &\geq \underline{x}_i \bar{x}_j + x_i \underline{x}_j - \underline{x}_i \underline{x}_j \\ y_{ij} &\geq \bar{x}_i \underline{x}_j + x_i \bar{x}_j - \bar{x}_i \bar{x}_j \\ y_{ij} &\leq \bar{x}_i \bar{x}_j + x_i \underline{x}_j - \bar{x}_i \underline{x}_j \\ y_{ij} &\leq x_i \bar{x}_j + \underline{x}_i x_j - \underline{x}_j \bar{x}_j. \end{aligned}$$

The number of new variables depends on the number of quadratic terms; each multiplication pair entails a new variable and four additional constraints. The reformulation results in a linear problem with binary variables.

Another idea to deal with the complexity introduced by the binary variables is the relaxation of the binary variables into a continuous interval. This strategy allows to tackle the problem with optimization techniques for continuous variables. The relaxation techniques may include additional constraints to ensure the integrality of the solution. Three quadratic relaxation procedures to tackle binary optimization problems are next reported ([Kochenberger et al. 2014](#)). In the first one, the binary variables $x_i \in \{0, 1\}$ are relaxed as $x_i \in [0, 1]$, with the addition of quadratic constraints $x_i(x_i - 1) = 0$ — one constraint for each binary variable in the original model.

In the second procedure, binary variables $y_i \in \{-1, 1\}$ replace the binary variables $x_i \in \{0, 1\}$, using the transformation $y_i = 2x_i - 1$. Following, the variables y_i are relaxed as $y_i \in [-1, 1]$ with the addition of the constraints $y_i^2 = 1$. Both transformations map the original combinatorial optimization problems into nonconvex and nonlinear optimization problems. Note that the first and second relaxation approaches are equivalent because $y_i^2 = 1 \iff (2x_i - 1)^2 = 1$; therefore, $4x_i(x_i - 1) = 0$ and $x_i(x_i - 1) = 0$.

The third idea to relax binary variables is to include in the objective function a penalization term known as the artificial power law, or solid isotropic material with penalization (SIMP). The binary variables x_i are replaced by $(x_i)^p$ in the objective function, where $p \geq 1$ and $x_i \in [0, 1]$. The SIMP approach was proposed by [Bendsøe \(1989\)](#) and improved by [Martinez \(2005\)](#). [Martinez \(2005\)](#) proved that a binary solution is assured for the SIMP

methodology if a constraint limiting the maximum number of binary variables equal to one is included in the formulation. This strategy does not entail extra variables in the formulation; however, the requirement that the number of binary variables equal to one is known a priori restrains the domain of applications of the SIMP approach.

2.3 Quadratic programming

An optimization problem can be called a quadratic programming problem when the objective function f is quadratic and the constraint set is linear — the quadratic programming area also includes problems with linear objective function and quadratic constraint sets. The solution methods for quadratic programming include interior-point methods, conjugate gradient method, gradient projection method, trust-region method and some extensions of the simplex algorithm ([Bertsekas 1997](#)). Necessary and sufficient optimality conditions for quadratic problems are based on Lagrangian duality whose main concepts are presented next.

2.3.1 Lagrangian duality and penalty methods

Consider the following optimization problem with equality and inequality constraint sets,

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \\ & h(x) \leq 0, \\ & x \in \Omega, \end{aligned} \tag{2.1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $g = (g_1, \dots, g_k)$, and $h = (h_1, \dots, h_m)$ are convex functions, with $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, k$ and $h_j : \mathbb{R}^n \rightarrow \mathbb{R}, j = 1, \dots, m$. All the functions are supposed to be differentiable, *i.e.*, they are in \mathcal{C}^1 .

The Lagrangian function of problem (2.1) is defined as (2.2),

$$L(x, \lambda, \mu) = f(x) + \lambda^T g(x) + \mu^T h(x), \tag{2.2}$$

where $\lambda = (\lambda_1, \dots, \lambda_k)$ and $\mu = (\mu_1, \dots, \mu_n)$.

The Lagrangian function (2.2) is a key concept to obtain the optimality conditions for the Problem (2.1), expressed, for instance in the Karush-Kuhn-Tucker (KKT) conditions ([Luenberger & Ye 2003](#)). Also, the Lagrangian function can be used to design optimization

approaches to obtain the optimal solutions to the problem (2.1) (Lasdon 2011).

In this research, the Lagrangian ideas are revisited with the perspective of the penalty methods, as follows.

Consider again the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega \end{aligned} \tag{2.3}$$

Now consider the problem (2.4) that expresses the minimization of a Lagrangian function associated to the Problem (2.3),

$$\min_x \quad L(x, \beta) = f(x) + \beta g(x). \tag{2.4}$$

For the problems studied in the thesis, the Lagrangian multiplier β is interpreted as a penalty parameter, the function f is linear or quadratic, and the function g is a single quadratic function, $g : \mathbb{R}^n \rightarrow \mathbb{R}$.

Using the perspective of the penalty methods (Luenberger & Ye 2003), there is a multiplier β for which an optimal solution x^* to the problem (2.4) is an optimal solution to the problem (2.3) if f is continuous and $g(x)$ satisfy three following conditions:

1. $g(x)$ is continuous;
2. $g(x) \geq 0 \forall x \in \mathbb{R}^n$;
3. $g(x) = 0 \iff x \in \Omega$.

2.3.2 Quadratically constrained quadratic program

An extension of the quadratic programming is the quadratically constrained quadratic programming problem (QPQC), which is NP-hard in the general case (Köppe 2012, Burer & Letchford 2012). The QPQC is frequently used to model statistical problems (Albers et al. 2011); in this research, it is used to model the unconstrained distance geometry problem presented in Chapter 7.

The QPQC is a mixed-integer nonlinear programming problems for which the objective function f is quadratic and the constraint set is composed by linear and quadratic constraints,

with at least one quadratic constraint. The following model summarizes the QPQC.

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0, \quad i = 1, \dots, k \\ & h(x) = 0, \quad j = 1, \dots, m \\ & x \in \Omega, \end{aligned}$$

where $f(x) = x^T Qx + c^T x$, $g = (g_1, \dots, g_k)$, and $h = (h_1, \dots, h_n)$, with $g_i(x) = A_i x - b$ for $i = 1, \dots, k$, and $h_j(x) = x^T M_j x + d^T x - a$ for $j = 1, \dots, n$.

For the cases where the matrix of the objective function Q and the matrix of the quadratic constraints M are positive semi-definite, the QCQP can be solved in polynomial time ([Anstreicher 2012](#)).

If the objective function is nonconvex, there are algorithms able to tackle the QPQC by using consecutive linearization techniques ([Audet et al. 2000](#), [Linderoth 2005](#)), convex approximations ([Anstreicher 2012](#)), semidefinite programming (SDP) ([Nesterov et al. 2000](#)), or decomposition based on difference of convex (DC) functions ([Horst & Thoai 1999](#), [Zheng et al. 2011](#)).

A new quadratic reformulation for binary problems

This chapter presents the main new concepts proposed to convert a binary optimization problem into an equivalent continuous optimization problem. A set of constraints is included in the original binary optimization problem, allowing its reformulation as an equivalent continuous optimization problem. Formal results show that the set of new constraints can be packed into a single quadratic constraint, and the equivalence between the original and the relaxed problems is demonstrated. A matrix representation of the constraint is also explored; the insights provided by the eigenvalues of the matrix allows unveiling attractive properties for the Lagrangian function of the continuous equivalent problem.

3.1 Introduction

Mathematical models with binary decision variables can be used to find the best solutions for decision-making processes, usually leading to difficult combinatorial optimization problems. In general, the exact solutions to these problems require high computational efforts ([Garey & Johnson 1979](#)), precluding the solution of large-scale problems. A strategy to tackle the computational burden is to relax the binary variables and devise constraints that should induce the value of the relaxed variables to a binary domain. An ideal relaxation technique would be able to obtain binary solutions with easily handling constraints that allow reducing the overall computation effort.

Consider the following standard binary optimization problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in \Omega, \end{aligned} \tag{3.1}$$

where $f : \{0, 1\}^n \rightarrow \mathbb{R}$, $x \in \{0, 1\}^n$, and $\Omega \subseteq \mathbb{R}^n$. The proposed relaxation maps the original binary problem presented in (3.1) into an equivalent continuous problem with relaxed variables, using additional continuous variables and a single additional constraint; this relaxed formulation is represented in (3.2)

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & x \in \Omega, \end{aligned} \tag{3.2}$$

where x is a vector of continuous variables and $\Omega \subseteq \mathbb{R}^n$.

3.2 The proposed quadratic relaxation

This thesis explores the benefits of a new quadratic relaxation for $\{0, 1\}$ binary variables. The idea is to relax all binary variables ($x \in \{0, 1\}$) for the continuous interval $[0, 1]$. However, it is necessary to devise additional constraints to the problem; otherwise, the binary feasibility could not be guaranteed.

The ideas proposed here introduces a set of auxiliary variables y , each y associated to a single binary variable x , with $y \in [0, 1]$. All variables x of the original problem are allowed to vary in the interval $[0, 1]$ (i.e. $x \in [0, 1]$). A set of constraints $(x - y)^2 = 1$, one for each pair of variables (x, y) , is also introduced in the problem to assure the binary feasibility of the new formulation.

As the proposed strategy requires the set of new variables $y_i \in [0, 1]$, one for each $x_i \in \{0, 1\}$, the formulation using continuous variables doubles the number of variables when compared with the original binary model. However, the addition set of constraints $(x_i - y_i)^2 = 1$ can be represented by the single constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$, which packs all the terms $(x_i - y_i)^2 = 1$; in other words, only a single quadratic constraint is added to the original constraint set — this idea can also be applied to the relaxations mentioned in Chapter 2.

The constraint $(x - y)^2 = 1$ restricts the possible solutions to two options, $x = 1$ and

$y = 0$, or $x = 0$ and $y = 1$ — the solution is binary whatever the case. Lemma 1 formalizes this result.

Lemma 1. *The binary variables x_i in problem P can be relaxed to $x_i \in [0, 1]$, by adding a set of continuous variables $y_i \in [0, 1]$ and a set of constraints $(x_i - y_i)^2 = 1$ ($i = 1, \dots, n$).*

Proof. Indeed, the only solutions of $(x_i - y_i)^2 = 1$ for $x_i \in [0, 1]$ and $y_i \in [0, 1]$ are $x_i = 0$ and $y_i = 1$, or $x_i = 1$ and $y_i = 0$; whichever case, the solutions are binary. ■

The n quadratic constraints required to keep the integrality property of the model could make the solution of the relaxed formulation as challenging as the original binary formulation. However, because it is possible to convert the set of n constraints $(x_i - y_i)^2 = 1$ into the single quadratic constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$, representing all the $(x_i - y_i)^2 = 1$ terms, only one constraint is added to the original formulation. Lemma 2 presents these ideas as a formal result.

Lemma 2. *Assume that the binary variables $x_i \in \{0, 1\}$, ($i = 1, \dots, n$) are relaxed as described in Lemma 1. The set of n quadratic constraints $(x_i - y_i)^2 = 1$ is equivalent to the single constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$.*

Proof. Note that the maximal value of $(x_i - y_i)^2$ for $x_i \in [0, 1]$ and $y_i \in [0, 1]$ is equal to 1. Therefore, the maximal value of $\sum_{i=1}^n (x_i - y_i)^2$ for $x_i \in [0, 1]$ and $y_i \in [0, 1]$ is equal to n . In other words, the constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$ is satisfied when each term $(x_i - y_i)^2$ reaches the maximum value. By Lemma 1, the solution is binary. ■

Now, consider again the binary optimization problem P .

$$(P) \quad \begin{aligned} & \min && f(x) \\ & \text{s.t.} && x \in \Omega \end{aligned}$$

where $f : \{0, 1\}^n \rightarrow \mathbb{R}$, $x \in \{0, 1\}^n$, and $\Omega \subseteq \mathbb{R}^n$.

The constraint set $x \in \{0, 1\}^n$ can be replaced, using the results of Lemma 2. Therefore, the problem P is converted into an equivalent continuous optimization problem P_C .

$$(P_C) \quad \begin{aligned} & \min && f(x) \\ & \text{s.t.} && \sum_{i=1}^n (x_i - y_i)^2 = n \\ & && x \in \Omega, \quad x \in [0, 1]^n, \quad y \in [0, 1]^n, \end{aligned}$$

where $\Omega \subseteq \mathbb{R}^n$. The equivalence between the problems P and P_C is demonstrated by Theorem 1.

Theorem 1. P_C and P are equivalent formulations in the sense that $P_C \subseteq P$ and $P \subseteq P_C$.

Proof. Lemma 1 and Lemma 2 prove that any feasible solution for the problem P_C is binary. This theorem should demonstrate the additional result that there is an unique transformation that maps a feasible solution for P into a feasible solution for P_C with the same value for the objective function $f(x)$, and conversely.

Assume that \hat{x} is a feasible solution for P . It is possible to build a feasible solution (\tilde{x}, \tilde{y}) , for P_C , with the rule $\tilde{x}_i = \hat{x}_i$ and $\tilde{y}_i = 1 - \hat{x}_i$ ($i = 1, \dots, n$), with the same value for the objective function, $f(\hat{x})$.

Conversely, suppose that (\hat{x}, \hat{y}) is a feasible solution for P_C . From Lemma 1 and Lemma 2, (\tilde{x}, \tilde{y}) is binary. Therefore, \tilde{x} is a feasible solution for P , with the same value for the objective function, $f(\tilde{x})$. \blacksquare

As mentioned in Chapter 2, the properties of the quadratic constraint set are essential to define a solution strategy. The next subsection relies on a matrix representation of the quadratic constraint set $\sum_{i=1}^n (x_i - y_i)^2 = n$ that allows to unveil useful properties for solving the continuous reformulation P_C in Chapter 4.

3.2.1 Matrix representation of the quadratic constraint

The matrix representation of the $\sum_{i=1}^n (x_i - y_i)^2 = n$ can be expressed by Equation 3.3 considering the vectors $x = [x_1, \dots, x_n]$ and $y = [y_1, \dots, y_n]$

$$zAz = n, \quad (3.3)$$

where $z = [x \ y]$ and $A = \begin{bmatrix} I & -I \\ -I & I \end{bmatrix}$. During the text, the notation zAz simplifies notation using the transpose vector $z^T Az$.

The singular value decomposition (SVD) presents the spectral properties of A ,

$$\begin{bmatrix} I & -I \\ -I & I \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2}I & \frac{\sqrt{2}}{2}I \\ -\frac{\sqrt{2}}{2}I & \frac{\sqrt{2}}{2}I \end{bmatrix} \begin{bmatrix} 2I & \bar{0} \\ \bar{0} & 0I \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2}I & -\frac{\sqrt{2}}{2}I \\ \frac{\sqrt{2}}{2}I & \frac{\sqrt{2}}{2}I \end{bmatrix}, \quad (3.4)$$

where $\bar{0}$ is a $n \times n$ zero matrix, and I is a identity matrix of dimension n .

Note that A is symmetric positive semidefinite, with n eigenvalues equal 0 and n eigenvalues equal 2. Since (3.3) is an equality constraint, the positive semi-definiteness property

cannot be explored; however, a Lagrangian approach allows to circumvent this difficulty.

Taking the constraint $zAz = n$ as the function $g(z) = n - zAz$, it is possible to define the problem P_L , using the Lagrangian function associated to the problem P_C ([Anstreicher & Wolkowicz 2000](#)),

$$(P_L) \quad \begin{aligned} \min \quad & f(z) + \beta g(z) \\ \text{s.t.} \quad & z \in \Omega, z \in [0, 1]^{2n}, \beta \geq 0, \end{aligned}$$

where $\Omega \subset \mathbb{R}^{2n}$.

Using the results discussed in Subsection [2.3.1](#), there is a multiplier β for which an optimal solution x^* to the problem P_L is also an optimal solution to the problem P_C if f is continuous and $g(z)$ satisfy the three following conditions: $g(z)$ is continuous; $g(z) \geq 0 \forall z \in \mathbb{R}^n$; and $g(z) = 0 \iff z = [x \ y]$ meets the constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$ ([Luenberger & Ye 2003](#)).

Theorem 2. *The Problem P_L is equivalent to the Problem P for a suitable value of β and f continuous.*

Proof. Considering the results from Theorem [1](#), that proved that the Problem P_C is equivalent to problem P , it is enough to prove that there is a value of β for which the Problem P_L is equivalent to the Problem P_C ; so by transitivity, the Problem P_L is equivalent to the Problem P .

The proof that there is a value of β for which the Problem P_L is equivalent to Problem P_C uses the penalty function approach, which requires the three properties of g stated above. It is immediate to see that $g(z)$ is continuous, because it is a quadratic function of continuous variables. The term zAz is a sum of the square of real values, for which the maximum is the positive value n , and the minimum is zero. Therefore, the maximal value of $g(z) = n - zAz$ is the positive value n , and the minimal value is zero; whichever case, $g(z) \geq 0$.

The last condition is that $g(z) = 0 \iff z = [x \ y]$ meets the constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$. Of course, if $z = [x \ y]$ meets the constraint $\sum_{i=1}^n (x_i - y_i)^2 = n$, then $g(z) = n - \sum_{i=1}^n (x_i - y_i)^2 = 0$. Conversely, if $g(z) = 0$, it follows trivially that $n - \sum_{i=1}^n (x_i - y_i)^2 = 0$; therefore, $\sum_{i=1}^n (x_i - y_i)^2 = n$.

The assumption of the continuity of the function f allows to conclude the demonstration that the Problem P_L is equivalent to the Problem P_C for a suitable value of β ; consequently, P_L is also equivalent to P for such β value.



3.3 Discussion

This chapter presented the theoretical concepts concerning the proposed relaxation for binary optimization problems. Lemma 1 proved that the proposed relaxation delivers only binary solutions. Lemma 2 proved that a single quadratic constraint can summarize the constraint set introduced by the relaxation. Theorem 1 showed the equivalence between the original formulation and the relaxed formulation for binary optimization problems; it proved that a single quadratic constraint can assure the integrality property for the relaxed formulation. Theorem 2 extended the binary properties for a Lagrangian reformulation, which will allow to handle nonconvex aspects in the application discussed in Chapter 4.

The ideas developed in this chapter open new perspectives for addressing binary problems using nonlinear continuous optimization approaches, an area that has been populated by very efficient solvers. Some innovative applications will be discussed in the Part II of the thesis.

Quadratic unconstrained binary optimization problem

This chapter explores the benefits of the new formulation in the context of quadratic unconstrained binary optimization problems (QUBO). It is worth noticing that, by applying the relaxation proposed here to the QUBO problem, we trade the complexity in dealing with a binary quadratic problem for the difficulties in dealing with a continuous nonlinear and non-convex problem. In order to tackle these difficulties, this research offers an equivalent unconstrained convex formulation using Lagrangian relaxation. The computational studies present comparisons between the proposed relaxation and the previous methodologies in the literature and present computational evidence of the benefits of the proposed formulation.

4.1 Introduction

The QUBO problem is an unconstrained optimization problem for which the optimization variables are binary and the objective function is quadratic. This NP-complete problem was proposed by Hammer & Shlifer (1971) to model service station location; it can be formulated as P_Q ¹,

$$(P_Q) \quad \begin{aligned} \max \quad & xQx; \\ \text{s.t.} \quad & x \in \{0, 1\}^n. \end{aligned}$$

The solution techniques to address the QUBO problem include exact approaches, meta-heuristics, and quantum computation (Calude et al. 2017, Chapuis et al. 2019). However,

¹The QUBO problem is formulated as a maximization problem to follow the characteristics of the instances considered in the computational studies.

due to the NP-complete nature of the QUBO (Glover et al. 2018), the computational cost for approaching the problem using exact methods is high. This feature has been precluding the exact solution of large instances of the problem, while heuristic approaches can tackle instances of up to fifteen thousand variables. Examples of heuristic approaches for the problem are local search methodologies, such as path relinking (Wang et al. 2012), tabu search (Glover et al. 1998, Palubeckis 2004, Kochenberger et al. 2013), global equilibrium search (Pardalos et al. 2008), and specialized local search methods (Boros et al. 2007, Wang et al. 2014). Bio-inspired methods such as genetic algorithms (Merz & Freisleben 1999) and memetic algorithms (Merz & Katayama 2004) have also been applied and they can handle instances of up to 2500 variables.

Exact approaches used to approach the QUBO problem include semidefinite programming (Helmberg & Rendl 1998) and valid inequalities (Glover et al. 2018). Billionnet & Elloumi (2007) uses branch-and-cut methods to solve instances of up to 80 variables. Lagrangian decomposition (Mauri & Lorena 2012) can tackle problems with up to 500 variables and find optimal solutions for instances with up to 100 variables. Another idea extensively explored is the use of relaxation techniques. Examples of these studies are Liberti (2007), Sherali & Smith (2007), Hansen & Meyer (2009), and Gueye & Michelon (2009). Approaches using preprocessing and reformulations techniques to reduce the overall complexity of the QUBO problem were explored by Pörn et al. (2017), Rodriguez Heck & Crama (2018), Crama & Rodríguez-Heck (2017), and Lewis & Glover (2017).

The solution of the QUBO problem via exact approaches faces two main difficulties. The first concerns the convexity of the objective function, which is related to the definiteness of the matrix Q . The second aspect comes from the binary nature of the optimization variables.

The approach proposed here converts the binary formulation into a continuous formulation by applying the relaxation and the penalization ideas presented in Section 3.2.1. This approach allows to increase the size of the QUBO instances tackled by exact methods.

Consider the problem P_Q relaxed as problem P_{CQ} , which uses the matrix representation discussed in Section 3.2.1.

$$(P_{CQ}) \quad \begin{aligned} \max \quad & z \tilde{Q} z \\ \text{s.t.} \quad & z A z = n \\ & z \in [0, 1]^{2n} \end{aligned} \tag{4.1}$$

where $\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}$ and $z = [x \ y]$.

A Lagrangian relaxation applied to the problem P_{CQ} allows to deal with the nonconvexity that comes from the equality constraint (4.1). These aspects are discussed in the next section.

4.2 Lagrangian relaxation

Considering the Lagrangian relaxation of the problem P_{CQ} given by (4.2).

$$\begin{aligned} P_{LQ} \quad \max \quad & z\tilde{Q}z - \beta(n - zAz) \\ \text{s.t.} \quad & z \in [0, 1]^{2n} \end{aligned} \tag{4.2}$$

The objective function in the problem P_{LQ} can be restated as the sum of three quadratic terms: $z\tilde{Q}z$, zAz , and zIz . The term zIz comes from the fact that z have n entries equal to one and n entries equal to zero, implying that $z^Tz = n$.

From Theorem 2 in Section 3.2.1, Problem P_{LQ} is equivalent to Problem P_Q (and P_{CQ}) for a suitable value of β . Going further, Proposition 1 proves that it is possible to choose a β value for which the formulation P_{LQ} is concave whenever Q is negative semi-definite (in other words, the relaxation does not jeopardizes the good features of the problem Q in the formulation P_Q).

Proposition 1. *For a suitable choice of β , whenever \tilde{Q} is negative semi-definite, the problem P_{LQ} has a concave objective function.*

Proof. The objective function of problem P_{LQ} can be rewritten as

$$\begin{aligned} P_{LQ} \quad \max \quad & z(\tilde{Q} + \beta A - \beta I)z \\ \text{s.t.} \quad & z \in [0, 1]^{2n} \end{aligned} \tag{4.3}$$

The condition for the objective function of P_{LQ} to be concave is that all the eigenvalues of $(\tilde{Q} + \beta A - \beta I)$ are non-positive, implying that $(\tilde{Q} + \beta A - \beta I)$ is negative semi-definite; i.e., $\lambda_{\min}(\tilde{Q} + \beta A - \beta I) \leq 0$ and $\lambda_{\max}(\tilde{Q} + \beta A - \beta I) \leq 0$.

Let $\lambda_{\min}(\tilde{Q})$, $\lambda_{\min}(A)$, and $\lambda_{\min}(I)$ be the minimum eigenvalues of \tilde{Q} , A , and I , respectively. The minimum eigenvalue of the matrix $(\tilde{Q} + \beta A - \beta I)$ is lower than the sum of the minimum eigenvalue of each term (Merikoski & Kumar 2004),

$$\lambda_{\min}(\tilde{Q} + \beta A - \beta I) \leq \lambda_{\min}(\tilde{Q}) + \lambda_{\min}(\beta A) - \lambda_{\min}(\beta I).$$

From Equation (3.4), $\lambda_{\min}(A) = 0$, then $\lambda_{\min}(\beta A) = 0$. Considering $\lambda_{\min}(\beta I) = \beta \lambda_{\min} I = \beta$, $\lambda_{\min}(\tilde{Q} + \beta A - \beta I) \leq \lambda_{\min}(\tilde{Q}) - \beta$. Since $\lambda_{\min}(\tilde{Q} + \beta A - \beta I) \leq (\lambda_{\min}(\tilde{Q}) - \beta)$, if $\beta \geq \lambda_{\min}(\tilde{Q})$.

Therefore, $\lambda_{\min}(\tilde{Q} + \beta A - \beta I) \leq 0$.

Now, let $\lambda_{\max}(\tilde{Q})$, $\lambda_{\max}(A)$, and $\lambda_{\max}(I)$ be the maximum eigenvalues of \tilde{Q} , A , and I , respectively. For the maximum eigenvalue $\lambda_{\max}(\tilde{Q} + 2\beta - \beta I) \leq (\lambda_{\max}(\tilde{Q}) + \beta)$, because $\lambda_{\max}(\beta A) = 2\beta$. Therefore, $(\lambda_{\max}(\tilde{Q}) + \beta) \leq 0$ and $\beta \leq -\lambda_{\max}(\tilde{Q})$.

Considering both conditions, $(\beta \leq -\lambda_{\max}(\tilde{Q}))$ and $(\beta \geq \lambda_{\min}(\tilde{Q}))$, whenever \tilde{Q} is negative semi-definite, there is a suitable choice of β for which the relaxed formulation is concave. ■

The reformulation of the QUBO problem as P_{LQ} and the Proposition 1 are explored in the computational studies presented in the next section.

4.3 Computational studies

Computational studies investigate the quality of the proposed relaxation in comparison with the quadratic relaxations from the literature. They also investigate the influence of the parameter β for the convergence of the problems.

Two sets of computational studies are performed. The first set comprises instances available in the literature for which the definiteness of matrix Q is not known a priori. For this set, the formulations expressed in problems P_{CQ} (4.1), P_{LQ} (4.3) are compared with the previous relaxation approaches from the literature.

The second set of computational studies is composed of a collection of designed instances with semi-definite Q (consequently, \tilde{Q} is also semi-definite). These instances allow comparisons between a diagonal perturbation, which will be expressed in Problem (4.5), with the alternative for setting a sufficiently large value for β (given by Proposition 1).

The computational tests start with an evaluation of the most suitable choice of the penalization parameter β . Table 4.1, Table 4.2, and Tables 4.3 present the gap ((4.4)) between the value of the objective function for the best solution in the literature (*best_solution*) and the value of the objective function for the equivalent formulations using the binary relaxation (*relaxation_solution*).

$$gap\% = 100 \times \frac{best_solution - relaxation_solution}{best_solution} \quad (4.4)$$

The computational test are performed using 135 instances from three QUBO problems libraries Beasley (1998), Billionnet & Elloumi (2007), and Glover et al. (1998). Solutions were obtained with the solver *fmincon* of Matlab R2016a in an Ubuntu 18.04 environment, with 16 GB of memory, Intel Core i7, CPU 3.20 GHz×12.

Table 4.1, Table 4.2, and Table 4.3 present the results for each library; the best results are highlighted. The columns *Instances* present the number of instances per class, the columns n provide the dimension of the instance, the columns *Density* provide the average density of the class (nonzero entries of the instance divided by the total number of entries). The rows *Optimal* and *Time(s)* present the number of optimal solutions per class and the total computational times (in seconds), respectively. The columns β and *QCQP* present the average gap for each class, respectively.

Table 4.1: Comparison between the average gap per class 100, 250, 500, 1000, and 2500 from [Beasley \(1998\)](#).

Classes	Instances	n	Density	β				QCQP
				10	50	100	250	
2500	10	2500	10%	**	0.62	0.46	0.92	1.86
1000	10	1000	10%	0.81	0.91	0.24	0.91	1.56
500	10	500	10%	2.07	1.27	1.07	1.28	3.58
250	10	250	10%	1.32	0.46	0.63	1.91	0.81
100	10	100	10%	0.20	3.59	0.94	1.10	6.81
Optimal				0	0	0	0	0
Time(s)				**	1506	16870	9255	777

** there is an instance without feasible solution

Table 4.2: Comparison between the average gap for classes *a*, *c*, *d*, *e*, and *f* from [\(Glover et al. 1998\)](#).

Classes	Instances	n	Density	β				QCQP
				10	50	100	250	
<i>a</i>	8	50-100	10-50%	0.62	0.68	0.54	0.30	7.46
<i>c</i>	7	40-100	10-80%	1.09	1.16	1.04	1.34	5.26
<i>d</i>	10	100	10-100%	1.94	2.45	1.87	1.51	9.79
<i>e</i>	5	200	10-50%	1.22	1.42	1.14	1.18	8.15
<i>f</i>	5	500	10-100%	6.33	6.07	5.77	5.95	15.55
Optimal				0	5	5	3	1
Time(s)				72	120	980	740	70

4.3.1 Influence of the initial solution

Since the relaxed formulations can be nonconvex due to the properties do matrix Q , the choice of an initial point is crucial to achieve a global solution. This analysis considers three instance, all of them with same dimension ($n = 100$) and with the same sparsity (10%). The entries of *beas100a* are integer numbers in the interval [-100;100]; for the instance *gka7c*, the entries outside of the matrix diagonal are integer numbers in the interval [-50;50], and the diagonal entries are integer numbers in the interval [-100;100]; for the instance *gka1d*, entries

Table 4.3: Comparison between the average gap for classes 100, 120.3, 120.8, 150.3, and 150.8 from (Billionnet & Elloumi 2007).

Classes	Instances	n	Density	β				QCQP
				10	50	100	250	
100	10	100	100%	1.59	2.22	1.69	1.12	11.15
120.3	10	120	30%	1.66	1.93	1.59	1.39	10.65
120.8	10	120	80%	2.83	1.89	1.61	2.25	8.54
150.3	10	150	30%	1.42	2.05	1.94	1.43	12.08
150.8	10	150	80%	0.90	1.30	1.42	1.01	6.87
Optimal Time(s)				2	2	1	6	0
				90	93	89	81	12

outside of the matrix diagonal are in the interval [-50;50] and the diagonal entries are in the interval [-75;75]. Each instance was executed 200 times, using random initial points in the interval $[0, 1]^n$.

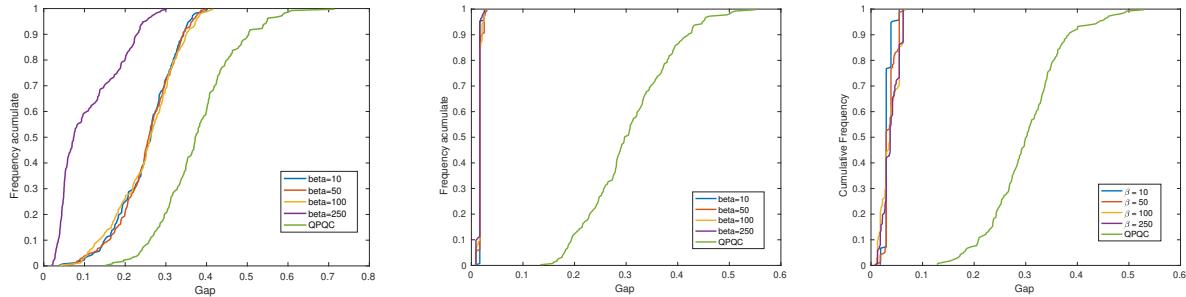


Figure 4.1: *Gap-to-target* plot for instances *beas100a*, *gka7c*, and *gka1d*, respectively.

Figure 4.1 helps to understand the behavior of the relaxations with respect to the initial point using *gap-to-target* plots. The *gap-to-target* plot is an adaptation of the ideas from the *time-to-target* plot proposed by Aiex et al. (2007). The idea behind of the *time-to-target* plot is to measure the robustness of the solution approach considering the random parameters. The idea behind the *gap-to-target* plot is to measure the robustness of the relaxed approach with respect to random initial points. In another words, the analysis regarding the execution times was replaced by the analysis of the gap between the solution achieved for each initial point and the best objective function value reported in the literature, considering the maximum execution time as 3600 seconds (one hour).

We noted that instance *gka7c* and *gka1d*, when using the Lagrangian relaxation, present gaps smaller than 5%, while *beas100a* has 30% in the best case. For the QCQP approach, the worst performance for *gka7c* and *gka1d* is 55% of gap and for *beas100a* is 75%. A difference between the instances occur in the sparsity pattern of the matrix Q , as illustrated in Figure 4.2. It is worth observing that the instance *gka7c* and *gka1d* that achieved the best

performance are the instances with nonzero diagonal elements. This observation matches with the well known result that the diagonal values are related to the definiteness of the matrix ([Sing 1976](#), [Thompson 1977](#)); this feature can help or disturb the convergence of exact solution approaches.

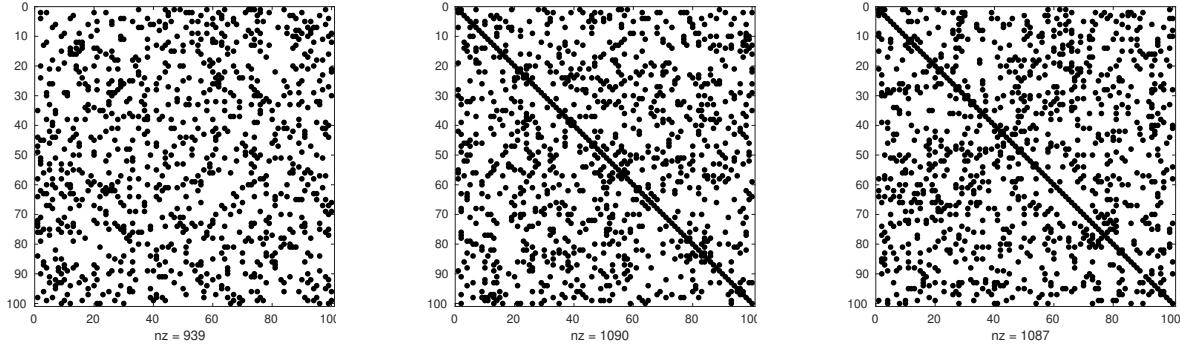


Figure 4.2: Nonzeros entries of the instances *beas100a* (left), *gka7c* (center) e *gak1d* (right), where *nz* is the total number of nonzero entries.

Therefore, the presence of nonzero diagonal elements can be the reason for the better performance of the QCQP relaxation for the instances *gka7c* and *gak1d* (with respect to *beas100a*). When considering the Lagrangian approach (β), the sum of the diagonal entries of Q with the β values helps to approach the convexity of the objective function, improving the convergence (see [Proposition 1](#)). In order to improve the performance of instances with zero entries in the matrix diagonal, such as the instance *beas100a*, a diagonal perturbation δ is proposed, as described in [Equation \(4.5\)](#). To implement such a perturbation preserving the value of the objective function, positive and negative perturbation values $n\delta$ ($n\delta \in \mathbb{R}$) are added to the objective function. The idea of improving the matrix diagonal is based on the concepts of diagonal regularization from the interior point optimization ([Gondzio 2012](#)).

$$P_{LQP} \quad \max \quad z\tilde{Q}z + n\delta - n\delta \quad (4.5)$$

Using the results from [Section 3.2](#), it is immediate to see that the number of entries equal to one in z is n . Therefore, $n\delta$ can be written as $\delta z^T z$, and [Equation 4.5](#) can be expressed as [Equation \(4.6\)](#) and [Equation \(4.7\)](#).

$$\max \quad z\tilde{Q}z + \delta z^T z - n\delta \quad (4.6)$$

$$\max \quad z(\tilde{Q} + \delta I)z - n\delta \quad (4.7)$$

Additional computational studies evaluate the benefits of using the perturbation δ for

solving the instance *beas100a*, considering the Lagrangian approach (with $\beta = 50$) and the QCQP approach. The *gap-to-target* on Figure 4.3 compare the use of positive and negative perturbation δ with the case without perturbation $\delta = 0$. Note that, for both cases, a suitable choice of δ improved the solution performance in more than 20% for the P_{LC} and 40% for the QCQP.

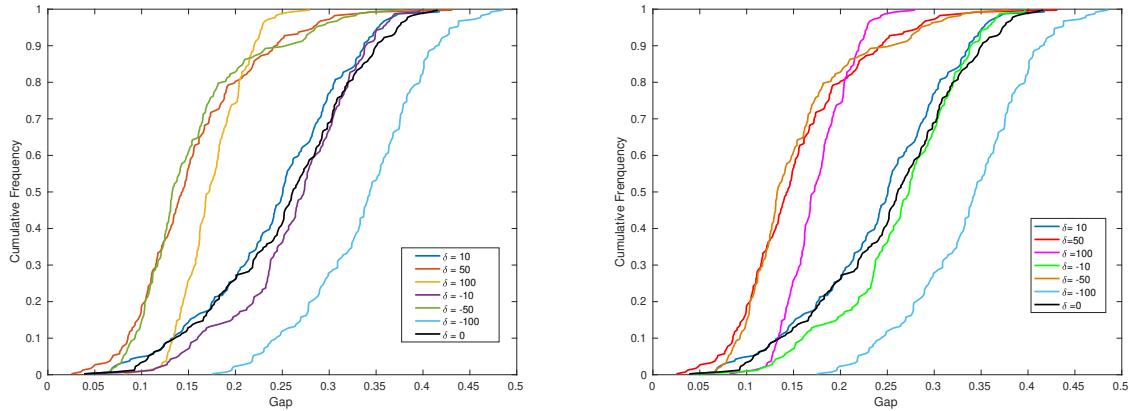


Figure 4.3: *Gap-to-target* plot for P_{LC} with $\beta = 50$ (left) and QCQP (right) for instance *beas100a*.

4.3.2 Studies with positive semi-definite matrix

These studies were designed to evaluate the benefits of the convex properties given by Proposition 1. Four set of instances were generated, each set with 10 instances with density of 10% and entries in the range $[-100, 100]$ ².

The β values provided by Proposition 1 (P1) were given by $\beta = \frac{-\max(eig(H))}{n} - 100$, where $\max(eig(H))$ is the largest eigenvalue of matrix H. For the approach without the use of Proposition (1), $\delta = 250$ and $\beta = 100$. The metric used to evaluate the quality of these β are the value of the objective function.

The maximal execution time required by the largest instances was 150 seconds. The results presented in Figure 4.4 show that the tailored values for β given by Proposition 1 improve the solution quality for all instances. Furthermore, the benefits of applying these results increase as the size of the instances increases.

²https://github.com/petrabartmeyer/QUBO_instance

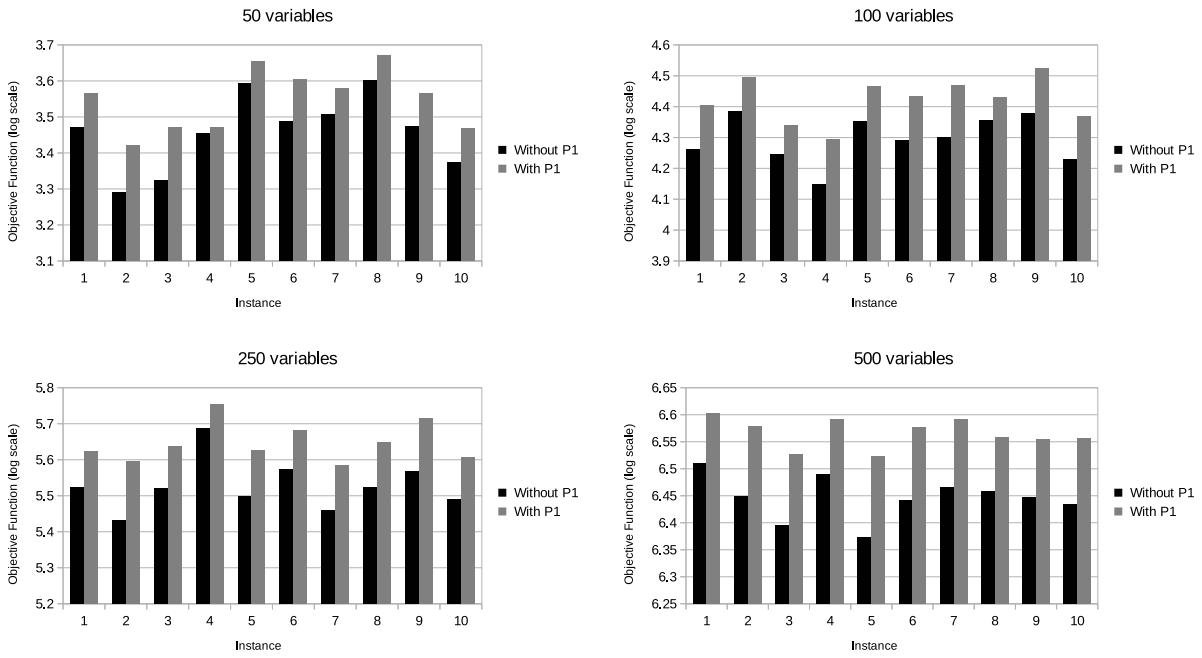


Figure 4.4: Values of the objective function, in logarithmic scale, with and without the parameter β calculated by Proposition (1) (With P1 and Without P1, respectively)

4.3.3 Lower bound quality — comparison with previous relaxations in the literature

This study presents a comparison between the proposed relaxation and the relaxations $x(x - 1) = 0$, and $x^2 = 1$. The results are shown in Figure 4.5, using a *performance profile* chart. The *performance profile* was proposed by [Dolan & Moré \(2002\)](#); it represents the percentage of instances solved in less than a given gap — all solutions are optimal if the gap is equal to zero. The x-axis in Figure 4.5 represents the gap calculated by Equation (4.4); the y-axis represents the percentage of problems solved with less than a given gap.

The computational tests for the relaxations $x^2 = 1$ and $x(x - 1) = 0$ considered the Lagrangian represented in Equation (4.9). No diagonal perturbation δ was used for these tests, because, as shown in Equation (4.8) and Equation (4.9), these relaxations already entail diagonal perturbations — it is worth reminding that approaches $x^2 = 1$ and $x(x - 1) = 0$ are similar (see Section 2.2).

$$\max \quad xQx - \beta(xIx - x) \tag{4.8}$$

$$\max \quad x(Q - \beta I)x - \beta x \tag{4.9}$$

With exception of the relaxation $x^2 = 1$, the others approaches have similar performance for about 60% of the instances; however, for the other 40% of the instances there is a major

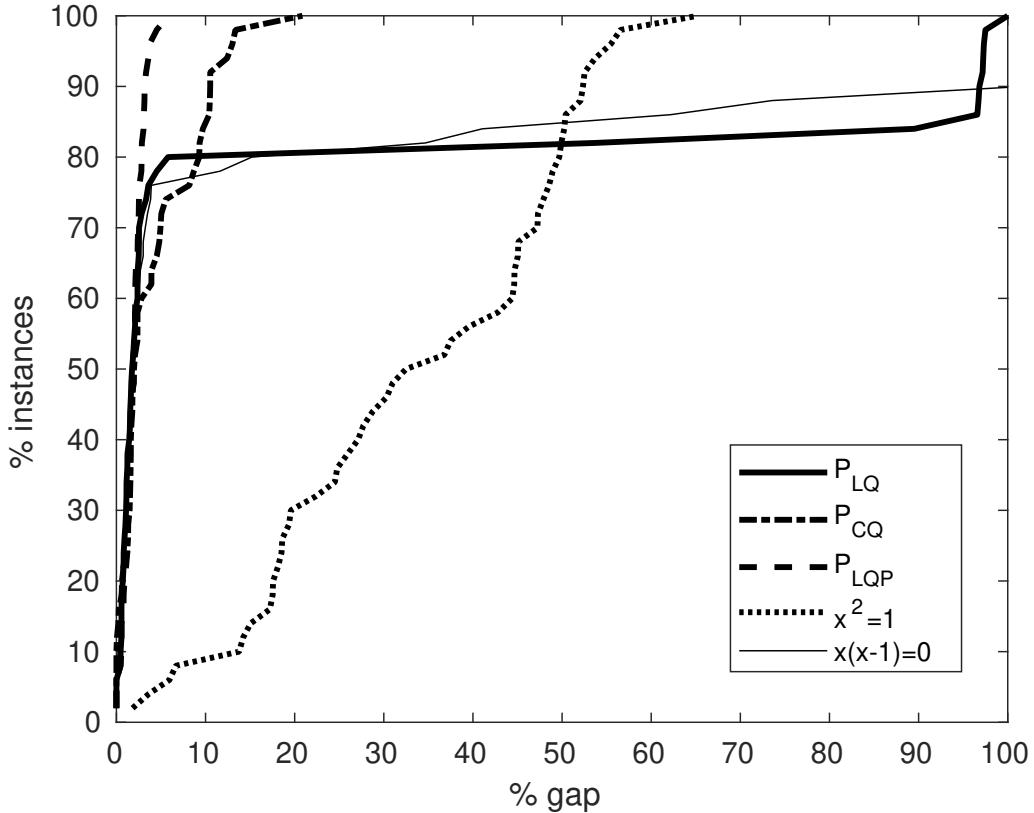


Figure 4.5: Performance profile of the gap for the instance of the test sets from [Beasley \(1998\)](#), [Billionnet & Elloumi \(2007\)](#), and [Glover et al. \(1998\)](#).

difference. The approaches P_{LQ} and $x(x - 1) = 0$ have the worst performances.

The approaches P_{CQ} and P_{LQP} have a more robust performance; P_{CQ} achieved less than 25% of gap, considering the whole instance set. The approach P_{LQP} achieved the best performance, with less than 5% of gap in the worst case.

The results in Figure 4.5 allows the conclusion that the quadratic relaxations provide good lower-bounds for the QUBO problem. Furthermore, the proposed approach using Lagrangian ideas and diagonal perturbation provides the best overall results.

4.4 Discussion

This chapter explored the benefits of the relaxation proposed in Section 3.2 when approaching the quadratic unconstrained binary optimization problem (QUBO). This NP-complete problem with nonlinear objective function and binary variables has attracted much interest over the latter years because of new applications, such as network reliability (discussed in the next section), and the recent developments of heuristic and exact approaches to address the problem. This research activeness and the intrinsic difficulty of the QUBO

problem make it a severe testbed for the relaxation ideas proposed in this thesis.

The theoretical results in this chapter proved that whenever the matrix of the objective function has definiteness properties the proposed approach is able to keep these properties. Furthermore, studies concerning the Lagrangian parameter (β) and a diagonal perturbation (δ) allowed to achieve formulations with better numerical properties.

The case studies put these results under perspective, providing comparisons with previous relaxation approaches. The results show that the benefits are consistent, with clear improvements for all the instance studied and the best overall performance when compared with the previous relaxation approaches in the literature.

Part II

Applications

Fault sensor placement for temporary faults identification

This chapter presents an optimization approach for the placement of fault sensors to reduce the detection time of temporary faults in distribution networks. This idea was inspired by the paper [Perdomo-Ortiz et al. \(2014\)](#), where it is proposed a mathematical model to diagnose electrical faults in an aircraft embedded system considering the presence of fault indicators. It is possible to interpret an embedded system as a distribution system; however, the components in the network are different and the assumption of an unlimited budget cannot hold. In this sense, this chapter explores the problem of locating fault sensors considering a maximum number of fault indicators and the reduction in the short-circuit detection time. The chapter starts with an overview of the use of fault sensors for distribution systems, followed by a mathematical formulation for the problem. The final part of this chapter evaluates the proposed mathematical model using a Monte Carlo simulation. Computational experiments show the benefits of the proposed methodology.

5.1 Introduction

Electrical network reliability is a broad field comprising measurements and analysis of frequency, duration, and extent of power outages ([Brown 2017](#)). Efforts to improve reliability indicators include the allocation of network components such as voltage and current indicators, switches, fuses, energy storage units, and fault indicators ([Chowdhury & Koval 2011, Brown 2017](#)). The study presented in this chapter focused on the use of fault indicators to detect temporary faults in distribution networks.

The fault indicators aim to improve the network information about short-circuit current,

providing feedback about the steady-state of the network for permanent and temporary faults. The identification of the existence of a permanent fault can be made by different techniques, such as measuring voltage and current at each end of the network, use customer notifications, or conduct inspections along the cables ([Saha et al. 2009](#)). The measurement of voltage and current can provide the first sign that there is a permanent failure, costumer notifications can provide information on the non-distributed energy points, but none of them is able to provide the exact fault location, which has to be found by a ground-search team. The sensor technology can provide more detailed information about the energy distribution considering the network topology. The fault indicator devices, also known as fault sensors, are equipment placed in the network that is connected with the recloser.

There are different types of fault sensors, and all of them identify the faults by checking if the network properties satisfy the steady-state network parameters. The differentiation of the devices occurs depending on information delivered during the feedback, such as current, voltage, or even visual indication, which helps ground teams. Indeed, fault sensors can comprise more than one feature and provide multiples feedback, which allows a more detailed description of the network state. The use of fault sensors to locate permanent fault has been investigated ([Mahapatro & Khilar 2013](#)), and it enables to compute quality indices such as system average interruption duration index (SAIDI), system average interruption frequency index (SAIFI) and customer average interruption duration index (CAIDI). Examples of these explorations are the immune algorithm of [Ho et al. \(2010\)](#), the mixed-integer linear programming model proposed by [Farajollahi et al. \(2016\)](#), and the multi-objective approach of [Acosta et al. \(2018\)](#).

In distribution networks, reclosers are able to perform automation schemes for power restoration, which allows them to remove short-circuit currents in the network. As soon as the recloser identifies the fault, it starts an operation protocol to remove the short-circuit current; if the current error remains after the protocol attempts, the network is de-energized. However, the recloser only starts the removing process when the current error backs to the recloser. During this spam time, the current error is traveling in the network, possibly causing damages. In this sense, sensors provide feedback for the recloser as soon as a current error is identified, allowing the recloser to perform protocol before the short-circuit travels to the recloser. It is worthwhile to highlight that the recloser protocol is not affected by the sensors.

Figure 5.1 illustrates a distribution network with fault sensors. The fault indicators, called sensors in the illustration, are placed in some bars of the network. The sensors devices

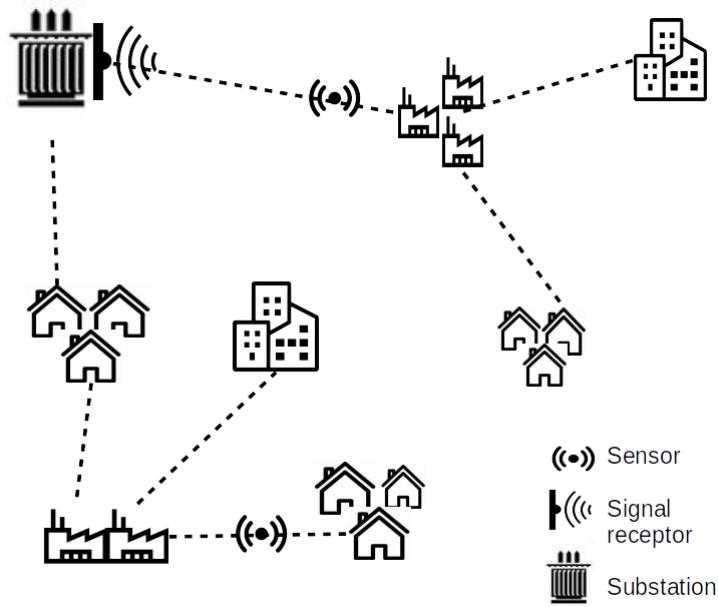


Figure 5.1: Iteration between sensor faults and the substation.

first establish communication with the signal receptor, which communicate with the reclosers. The information collected by the sensors is transmitted to a recloser in a time span shorter than the time required to trigger a recloser by a short-circuit current. This characteristic allows the use of fault sensors to reduce the detection time of temporary faults, improving the reliability of the system.

In an ideal scenario a fault sensor could be placed in each edge of the network, as proposed by Perdomo-Ortiz et al. (2014) for embedded systems in an airspace bus. However, the assumption of large budgets for improving the reliability of spacecrafts is usually not true for improving the reliability of a distribution network. Therefore, to achieve the best improvement in reliability with the allocation of fault sensors in distribution networks, it is necessary to identify the maximum number of devices that fits a given budget and to unveil the optimal use of these devices.

This chapter proposes an integer programming model to identify the optimal placement of fault sensors in a distribution network, considering the benefits of reducing the short-circuit detection time for temporary faults. The technical assumptions considered here are the following

- Reclosers can interrupt the normal operation to remove temporary faults;
- The fault sensors are able to communicate with the reclosers using a specific communication device;

- The communication time between each possible sensor position and a recloser is known a priori;
- There is a maximum number of sensors in the network protected by a recloser (upstream a recloser);
- Each node of the network is assigned to a sensor with the shortest communication time;
- The faults studied can occur with a single-phase or with the three-phases.

The following sections present the main contributions of the research developed in this chapter – the design of a strategy that allows the reduction of the propagation of temporary failures in distribution networks, a mathematical model for the problem, and the proposal of a solution strategy combining integer programming and Monte Carlo simulation. The next section presents an interpretation of the placement of indicator faults in distribution networks from a binary programming perspective. A deterministic mathematical model is proposed for the problem. Following, random aspects of the problem are considered in a local search, which uses a Monte Carlo simulation to improve the solutions. Section 5.3 presents a case study where computational experiments illustrate the tradeoff between the number of fault sensors and the reduction in the detection times.

5.2 Mathematical model

The proposed mathematical model minimizes the fault detection time considering the best fault sensor placement for a given topology of the distribution network. Considering the technical assumptions made above, the maximum number of indicators is represented by S , and for the recloser as s_r , D gives the maximum allowed distance between the recloser and an indicator; F represents the possible faults types. The parameter T_{ik} is the time before the recloser be performed, for a fault of the type k in the node i . The model comprises by single-phase and three-phase faults, and their respective recloser trip times (slow or quick).

The TS_{ij} is the recloser activation time for a fault in the downstream node j and an indicator in node i . The topology of the network is described by the binary matrix M_{ij} , where $M_{ij} = 1$ if i and j are in the same branch and i proceed j , zero otherwise. For each recloser r , there is a set of the nodes assigned to it; this set of nodes is called the protection zone of the recloser r .

The distance between each node i and the assigned recloser is given by d_i . The variables x_i and L_{ij} are binary; $x_i = 1$ if there is an indicator located in the node i , zero otherwise; $L_{ij} = 1$ if the indicator located in i monitors node j .

$$\min \quad \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^F T_{ik}(1 - L_{ij}) + \sum_{i=1}^n \sum_{j=1}^n TS_{ij}L_{ij} \quad (5.1)$$

$$\text{s.t.} \quad \sum_{i=1}^n x_i \leq S \quad (5.2)$$

$$\sum_{i \in R_r} x_i \leq s_r, \forall r \quad (5.3)$$

$$L_{ij} \leq M_{ij}, \forall i, j \quad (5.4)$$

$$L_{ij} \leq x_j, \forall i, j \quad (5.5)$$

$$\sum_{i=1}^n L_{ij} \leq 1, \forall j \quad (5.6)$$

$$d_i L_{ij} \leq D, \forall i, j \quad (5.7)$$

$$L_{ij} \in \{0, 1\}, x_i \in \{0, 1\} \quad \forall i, j \quad (5.8)$$

The objective function (5.1) minimizes the detection time for all the nodes, considering the nodes with and without sensors. Constraint (5.2) establishes the maximum number of fault sensor for the network. Constraint set (5.3) sets the maximum number of fault sensor for recloser. For constraint set (5.4), the matrix M represents the topology of the network, and it limits the values of L . Constraint set (5.5) relates the sensor position variables x_i and the protection variables L_{ij} . Constraint set (5.6) limits to one the number of sensors assigned to a node, the idea is to avoid that a fault node to be double-counted in the objective function since it can be detected by more than one sensor. The maximum distance radius between recloser and sensor is computed by constraint set (5.7). Constraint set (5.8) establishes the variables domain.

The formulation (5.1)-(5.7) is a binary linear problem, which can be approached by commercial solvers for integer programming, such as Cplex ([Cplex 2009](#)) and Gurobi ([Bixby 2014](#)). However, for large instances, solution approaches by integer programming may not be the suitable strategy to address the problem. In such cases, the results from Chapter 3 and Chapter 4 can be a better alternative to handle the problem. These ideas can be applied in two ways. The first consider the direct application of the ideas from Chapter 3, considering the Lagrangian relaxation of the problem.

The formulation considering the Lagrangian relaxation proposed in Section 3.2 can be stated as follows.

$$\begin{aligned}
\min \quad & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^F T_{ik}(1 - L_{ij}) + \sum_{i=1}^n \sum_{j=1}^n TS_{ij}L_{ij} + \dots \\
& \dots + \beta(n^2 - \sum_{i=1}^n \sum_{j=1}^n (L_{ij} - W_{ij})^2) + \gamma(n - \sum_{i=1}^n (x_i - y_i)^2) \\
\text{s.t.} \quad & \sum_{i=1}^n x_i \leq S \\
& \sum_{i \in R_r} x_i \leq s_r, \forall r \\
& L_{ij} \leq M_{ij}, \forall i, j \\
& L_{ij} \leq x_j, \forall i, j \\
& \sum_{i=1}^n L_{ij} \leq 1, \forall j \\
& d_i L_{ij} \leq D, \forall i \forall j \\
& L_{ij} \in [0, 1], W \in [0, 1], x_i \in [0, 1], y_i \in [0, 1] \quad \forall i, j
\end{aligned}$$

where W and y are the additional variables required by the proposed relaxation (Section 3.2), and the parameters β and γ are the penalty parameter required by the Lagrangian relaxation. This formulation has a quadratic objective function and continuous variables. Therefore, nonlinear continuous optimization solvers, such as Knitro ([Waltz & Plantenga 2011](#)) and Ipopt ([Wächter & Biegler 2009](#)), can be used to tackle the problem.

The second idea comes from the paper [Perdomo-Ortiz et al. \(2014\)](#), where the linear model proposed is rewritten as a quadratic unconstrained binary optimization problem (QUBO) – [Perdomo-Ortiz et al. \(2014\)](#) were motivated to use the QUBO formulation because they considered it better suited to solve the problem via quantum computing. The idea used to convert model (5.1)-(5.8) into a QUBO formulation is presented as follow; a matrix representation of the problem (5.1)-(5.8) is adopted.

$$\begin{aligned}
\min \quad & c^T z \\
\text{s.t.} \quad & Az = b \\
& z \in \{0, 1\},
\end{aligned}$$

where the matrix A represents the constraint sets, b is the right-hand, and c is the cost vector. The matrix A is a block-matrix composed by the constraints of (5.2)-(5.8). A necessary condition for the inequality constraint into an equality representation is the use of

a slack-variable (s_j) for each constraint in the formulation. The cost of the slack-variables are null. The variable z is the composition of the variables set x_i in a vector \mathbf{x} , the variables set L in a vector representation $\mathbf{L} = [\mathbf{L}_1, \dots, \mathbf{L}_n]$, where L_i are all the entries of the i -th row of the matrix L , and the slack-variables s_i in a vector representation \mathbf{s} , i.e., $z = [\mathbf{x} \ \mathbf{L} \ \mathbf{s}]$.

The notations $\mathbf{1}_k$ and $\mathbf{0}_k$ represent the unitary and the null vectors with k entries. The notations $\bar{\mathbf{1}}_{k,n}$ and $\bar{\mathbf{0}}_{k,n}$ represent the unitary and the null matrix with dimension $k \times n$. The identity matrix of dimension $n \times n$ is given by \mathbb{I}_n . The first constraint block (B1) is related to (5.2) and (5.3) are given by

$$B1 = [\mathbf{1}_n \ \mathbf{0}_{n^2} \ 1 \ \mathbf{0}_{k-1}].$$

The block representation of (5.5) is given by (B2)

$$B2 = [\bar{\mathbf{O}}_{n^2,n} \ \bar{\mathbf{1}}_{n^2,n^2} \ \mathbb{I}_{n^2}].$$

Block B3 is composed by n parcels to represent the constraint set (5.5)

$$B3_i = [-\mathbb{I}_n \ \bar{\mathbf{1}}_{n,n} \ \mathbb{I}_n], i = 1, \dots, n.$$

Block B4 represents the constraint set (5.6)

$$B4 = [\bar{\mathbf{O}}_{n^2,n} \ \bar{\mathbf{1}}_{n^2,n^2} \ \mathbb{I}_{n^2}]$$

The last block (B5) summarizes the constraint set (5.7)

$$B5 = [\bar{\mathbf{0}}_{n^2,n^2} \ \mathbb{I}_{n^2} \ \mathbb{I}_n].$$

The right-hand side vector of the constraint set (b) is represented by

$$b = [S \ \mathbf{M} \ \mathbf{0}_n \ \mathbf{1}_{n^2} \ \mathbf{D}_n]. \quad (5.9)$$

where \mathbf{M} is the vector representation of the matrix M , in the same way that \mathbf{L} represents the matrix L .

The matrix formulation allows rewriting Problem (5.1)-(5.8) as a quadratic unconstrained binary optimization (QUBO) problem as follows:

$$\min \quad c^T z - \beta(Az - b)^T(Az - b)$$

$$\text{s.t.} \quad z \in \{0, 1\}^n$$

where β is a penalty value. This formulation can also be addressed with the results from Chapter 3 and Chapter 4.

Whichever the optimization model and solution strategy adopted to handle the problem, the local search with a Monte Carlo simulation described in the next section is used to improve the sensor allocation by considering the stochastic aspects of the problem.

5.2.1 Local search with a Monte Carlo simulation

The local search strategy uses a Monte Carlo simulation ([Billinton & Allan 1992](#)) to evaluate the quality of the solutions provided by the deterministic formulations discussed in the previous section.

In order to improve the quality of the results provided by the deterministic mathematical model, a local search heuristic is applied to the solutions provided by the model (5.1)-(5.8) — the computational effort required to address the instance in the case study did not require the relaxation approach. The more significant improvements obtained by the local search heuristic occurs for the topologies with a small number of fault sensors. Indeed, this result should be expected, because as the number of indicators increases, the difference between the stochastic and deterministic model decreases. The next pseudocode summarizes these ideas.

```

Data:  $TS, T, NC$ 
Result: best change per configuration
for  $1 \dots NC$  do
     $t_1 \leftarrow$  Find the smallest contribution time among the sensors in the configuration
    remove the sensor with the smallest contribution
    for  $edge \in feeder$  do
         $edge \leftarrow test\_sensor$ 
         $t_2 \leftarrow$  Run the Monte Carlo simulation
        if  $t_2 > t_1$  then
            | remove  $test\_sensor$ 
        end
    end
end
```

Algorithm 1: Local search procedure.

The idea presented in Algorithm 1 is to change the position of the indicator with the smallest benefit for the detection time. Note that just a single indicator is changed in each configuration. The decision for a single indicator analysis comes from the high computational effort entailed by the Monte Carlo simulation each time that a new configuration is tested.

The Monte Carlo simulation is used to evaluate the quality of the changes proposed by

the local search. Each Monte Carlo iteration considers a different recloser protocol, which given by the combination of two quick operations and an odd number of slow operations. The detection time for each node time with faults indicators (T) and without fault indicators (TS) are estimated using the faster and slower recloser operation times for single-phase and three-phase; it was generated four different fault values ($F = 4$) for each node. The fault indicator (T) was calculated as $T = t_{dtc} + t_{Tx1} + t_{Tx2} + t_{prs}$, where t_{dtc} is the time required by the sensor to detect the fault; t_{Tx1} is the transmission rate between sensor and recloser; t_{Tx2} is the transmission rate between receptor and recloser; t_{prs} is the data processing rate.

The simulation starts with the choice of a faulty node using a uniform distribution; the closest upstream indicator is identified. This identification procedure selects the shortest operating time of the indicator avoiding double-counting for the same node; in the mathematical model, L_{ij} plays this role.

Data: TS, T, NC, NS

Result: total time per node

while $iter \leq max_iter$ **do**

Randomly choose the number of fast operations {0, 1, 2};

Randomly choose the number of slow operations {0, 1};

Randomly choose a node to simulate the fault;

Calculate the recloser trip time;

for $1 \dots NC$ **do**

for $1 \dots NS$ **do**

Find the indicators assigned for the node;

Choose the indicators with the smallest time;

end

Update the total time;

end

$iter \leftarrow iter + 1$

end

Algorithm 2: Monte Carlo simulation.

Algorithm (2) summarizes the Monte Carlo simulation process. The parameters of the algorithm are the number of indicators in each configuration (NS), the detection time for a given indicator configuration (TS), the detection time for the recloser (T), and the maximum number of Monte Carlo iterations (max_iter). Parameter NC represents the number of scenarios to be evaluated. The idea is to use the same set of faults to compare all the scenarios.

The next section explores the mathematical model and the local search procedure proposed here in a case study based on a network available in the literature.

5.3 Case study

The case study was designed to explore the benefits of fault sensors in twelve scenarios for the same instance; the scenarios are characterized by the number of fault sensors per recloser. The computational experiments also test the quality of the proposed model in representing the stochastic problem.

The instance considered was the test feeder IEEE-123¹ considering the configuration proposed by Alam et al. (2018). This configuration presents two reclosers and fourteen downstream fuse-saving coordination. The number of edges able to receive a fault indicator is $n = 119$; these are all the edges downstream of the reclosers. Figure 5.2 represents the network topology and the device configuration.

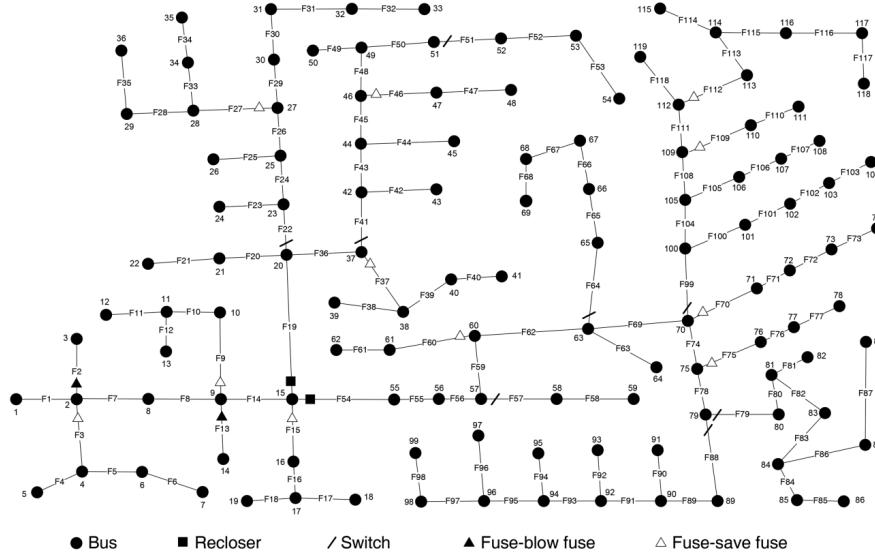


Figure 5.2: Topology and device configuration for the IEEE123 buses network. Image adapted from Alam et al. (2018).

The maximum number of indicators per recloser is equal to twelve, with a transmission rate time equal to $2ms$, and a maximum Euclidean distance between recloser and fault sensor as 6.5km. These parameters are technical limitations of the device used as a reference ².

The mathematical model (5.1)-(5.8) was implemented in the AMPL (Fourer. et al. 2003) and solved using CPLEX (Cplex 2009). The Monte Carlo simulation was coded in Python 3.7. Both processes run in a desktop PC under Ubuntu 18.04 with a Core i7 processor and 16 GB of RAM. The software OpenDSS (Dugan 2012) calculates the admittance matrix.

The uniform distribution selects the nodes during the evaluation. It was considered ten million faults to evaluate the quality of the model (5.1)-(5.8), the average selection per node

¹<http://sites.ieee.org/pes-testfeeders/resources/>

²<https://selinc.com/pt/products-section/fault-indicators-sensors/>

was 10 000 with standard deviation of 329.68.

The computational studies focus on detection times, and they are summarized in Figure 5.3. The dotted red line illustrates the ideal configuration where there are indicators on all feeder edges. The solid curve interpolated the detection time for the scenarios from one up to twelve fault sensors per recloser.

The solid curve shows that the deterministic models can minimize the stochastic problem. However, the picks for $S = 3$ and $S = 12$ bring the evidence that the linear programming model loss quality in some scenarios, which emphasizes the advantages of using the Monte Carlo simulation to check failures and a local-search procedure to improve the solution quality.

The configurations with three and four devices per recloser highlighted the relationship between the installation site and the contribution of the indicators. The lower the number of sensors, the higher the impact caused by a change in the sensor location. Figure 5.3 shows that the maximum decrease occurs for the scenario with two fault sensors per recloser. This scenario is illustrated in Topology A in Figure 5.4. Topology B in Figure 5.4 present the scenario with three fault sensors per recloser. The difference of the topology occurs for fault sensors in the protection zone of $R2$; these changes are responsible for increasing the detection time when compared with Topology A.

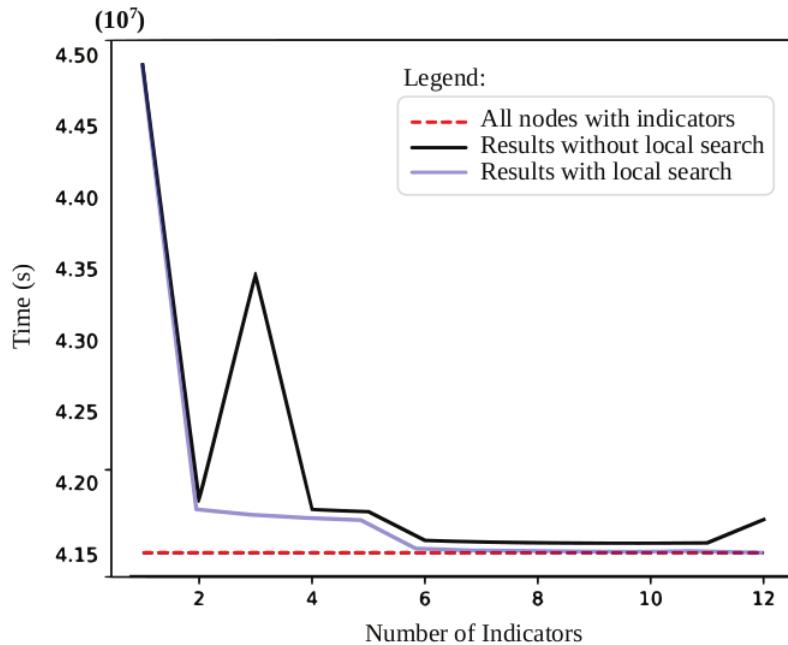


Figure 5.3: Total improvement of the detection-time. The image comprises the number of indicators per recloser on the x-axis and the detection time on the y-axis. Image from [Santos et al. \(2019\)](#).

The detection time remains in decline until the configuration with eight fault indicators. This topology is represented by Topology C in Figure 5.4. The detection time backs to increase in the last configuration (Topology D). Note that benefits close to the scenario of maximum observability can be achieved with a reduced number of fault indicators. As shown in Figure 5.3, scenarios with more sensors present solutions closer to the optimal scenario, represented by the dashed red line.

The light blue line presents the results after the local search procedure. Note that the local search is able to overcome the difficulties related to the scenarios $S = 3$ and $S = 12$ generating a continuous decreasing curve from the scenario $S = 1$ to the scenario $S = 12$.

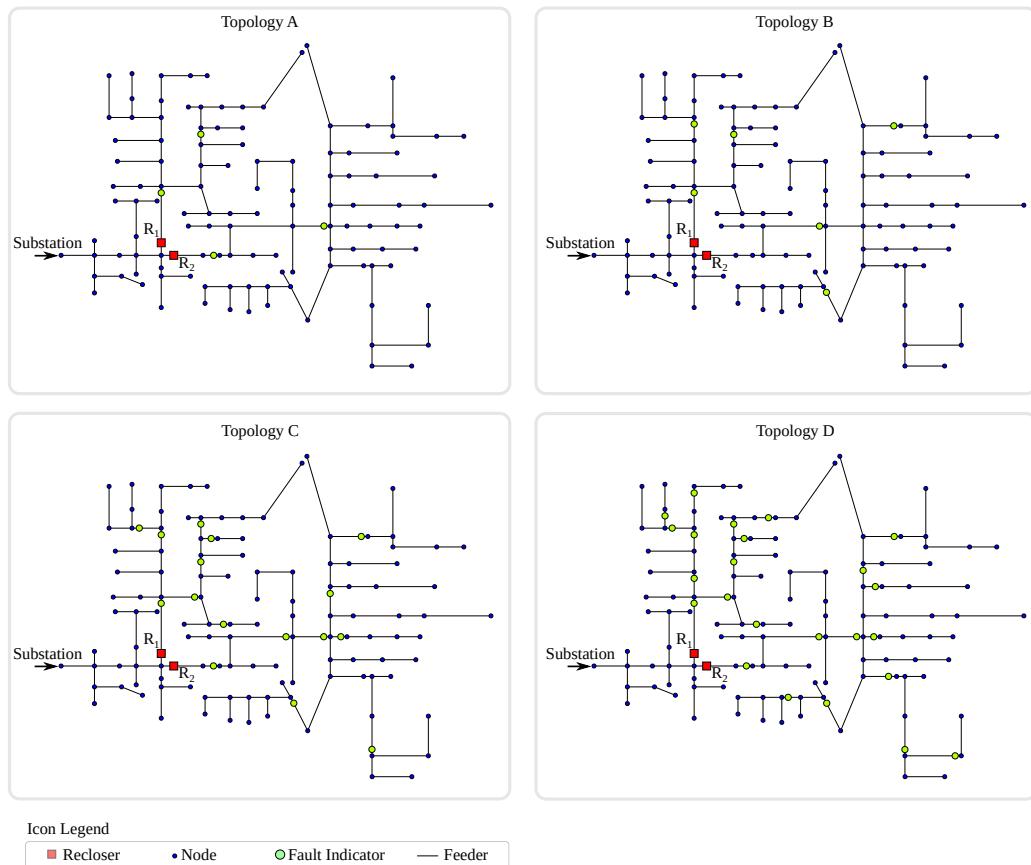


Figure 5.4: Position of the indicators for scenarios with two, three, eight, and twelve fault sensors per recloser. Image from [Santos et al. \(2019\)](#).

A multiobjective perspective can also be used to interpret the results presented above ([Miettinen 2012](#)). The curve generated by the scenarios can be seen as a tradeoff between the number of fault indicators and the maximum reduction in the detection time. In this sense, it is a bi-objective problem where the first objective is to minimize the detection time, and the second objective is the reduction in the number of fault indicators. The solution strategy adopted can be interpreted as an ϵ -constrained method, where the number of fault indicators was added to the constraint set, and the ϵ value is discrete, because it represents the number

of fault indicators. The conflict between these two objectives is observed in Figure 5.3. Note that, with this interpretation, scenarios with more than six fault indicators per recloser are dominated solution, since the extra number of fault indicators did not improve the reduction in the detection time.

5.4 Discussion

This chapter presented a new approach to the allocation of fault sensors in adaptive protection systems. The contributions of the research are the design of the optimization model for the reduction of the propagation of temporary failures in distribution networks, and the solutions strategies, including the development of the local search with a Monte Carlo simulation to consider random aspects of the problem.

The computational studies showed that the proposed mathematical model works as an approximation for the stochastic problem and successfully handles the allocation of fault sensors without dealing with the stochastic variables. The local search strategy allowed to improve the solutions with the additional capability of dealing with the random aspects of the problem; in other words it improves the adherence of the approach to the real-world problems.

The methodology presented competitive results when compared with a ideal scenario where all the edges carry a faulty sensor. The computational studies indicated that the allocation of fault sensors to around 5% of the edges can provide a reduction in the detection time that is very close to the maximum reduction.

CHAPTER 6

Automatic segmentation for myelinated axons

This chapter addresses the problem of segment myelinated axon in microscope images. The approach of this problem using combinatorial optimization and with a coloring algorithm are innovative aspects of this chapter. They are alternatives for the current approaches in the literature based on pattern recognition.

The interpretation of this application as a binary optimization problem originate from the idea that an image can be mapped in a graph. This interpretation allows to consider each Hamiltonian cycle in the graph as a myelinated axon. The investigation of the existence of Hamiltonian cycles can be modeled as an optimization problem with binary variables and linear constraints. Computational studies illustrate the benefits and limitations of using the relaxation proposed in Chapter 3 to address the automatic segmentation for myelinated axon images with a combinatorial optimization perspective.

The alternative for the combinatorial optimization approach explored in this chapter is based on the flood-fill graph coloring algorithm, enhanced with myelin sheath information. The benefits and limitations of the two ideas are discussed; the quality of this approach is evaluated with a large set of statistical tests.

These investigations were mainly developed during a six month research period at the "Brain Research Institute" of the "University of California—Los Angeles", under the supervision of Prof. Leif Havton.

6.1 Introduction

The computer vision area has significant applications in medical image analysis. Applications go from the segmentation of bones and organs (Minnema et al. 2018), to diagnosis support using machine learning techniques (Lee et al. 2017, Leo et al. 2017). Essentially, the computer vision has been applied to reduce the human efforts in high time-consuming tasks as measurements and counting of structures in medical images. An example of these tasks is the measurement of axons from nerve roots images, which is the object of this research.

To better understand the characteristics of these applications, Figure 6.1, on the left-hand side, illustrates how the nerve roots connect to the spinal cord. Each branch arriving on the spinal cord is a nerve root composed of hundreds or thousands of axons, as illustrated on the right-hand side of Figure 6.1.

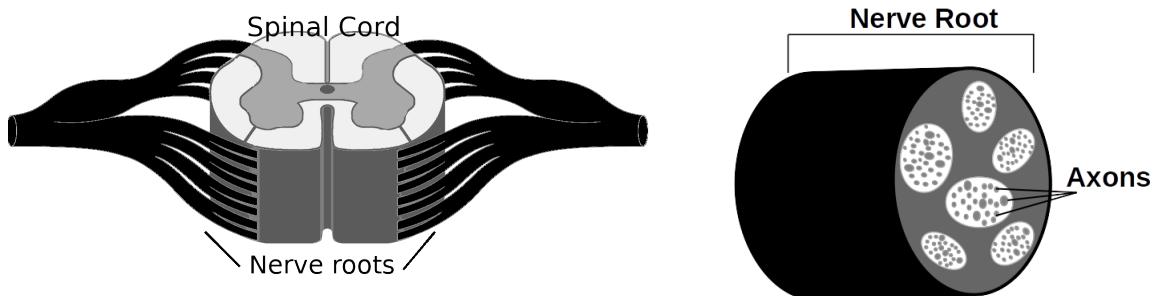


Figure 6.1: Representation of a nerve root connect to a spinal cord (left) and a cross section of a nerve root highlighting the axons in the structure (right).

The neurons connect to the spinal cord using dendrites and to the muscles by axons terminals. The left-hand side of Figure 6.2 represents these situation. The information travels between the two extreme points through axons, whose caliber, length, and myelin thickness affect the information transmission rate. Figure 6.2 illustrates the presence of myelin sheath covering the axon. The myelin sheaths are generated by Schwann cells that wrap the axon in layers of myelin sheaths. The existence of myelinated sheaths improves the information transmission; the more myelin sheaths, the faster the information transfer. In microscope images, the myelin sheaths are identified by dark closed shapes, and the area inside of these closed contours are the axons.

In the context of this research, the information to be obtained from nerve root images are area, diameter, number of axons, myelin thickness, and spatial distribution of the axons. Retrieving these data from an image is usually a time consuming procedure. The imaging process starts with the harvesting of the nerve roots, followed by chemical processing and

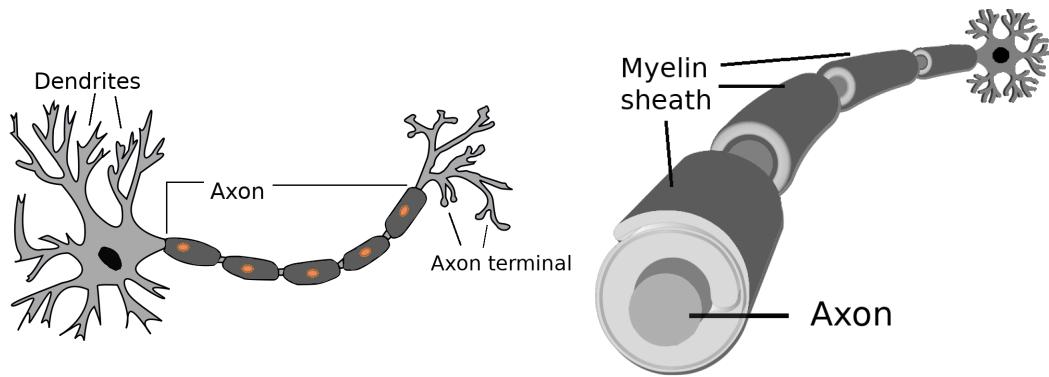


Figure 6.2: A motoneuron with the myelin sheath (left) and an axon structure covered by myelin sheath, which can be recognized by as a dark contour presented on scope images (right).

embedding techniques. In this last step, the samples are cut perpendicular to the nerve root orientation then placed on slides for further scope analysis. All these processes can reduce the quality of the sample, turning the measurements more difficult.

The next section discusses the necessary information about nerve root image acquisition and presents the expected outcome of the axon indication in the images after the laboratory processing steps.

6.2 Nerve root images

There are different imaging techniques able to handle nerve roots. Non-invasive techniques, such as MRI and X-ray, provide *in vivo* images but are not able to identify small structures. On the other hand, microscopy techniques such as TEM (transmission electron microscopy) and LM (light microscopy) provide better image quality, but it requires a sequence of preprocessing steps before obtaining the desired imaging. Figure 6.3 gives examples of images from TEM and LM scopes; it illustrates the range of shapes and structures that the segmentation process looks for. The axons are inside the dark closed shape generated by the myelin sheath.

The images analyzed in this research are from LM scopes. The image quality varies due to the technical preparation, the scope, and the quality of the material. These variations are a challenge for segmentation techniques based on pattern recognition, such as proposed by [Zaimi et al. \(2016, 2018\)](#).

Microscopy images present different sizes depending on the magnification used during the process. The higher the magnification, and the larger the number of individual images, the more time-consuming is the image processing. The image size also depends on the animal

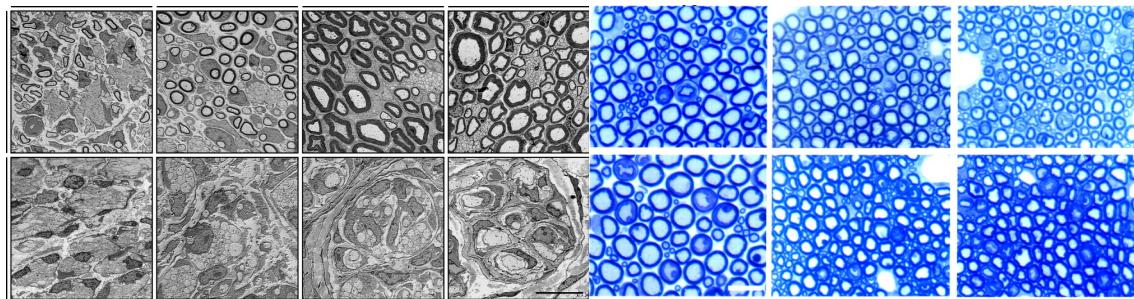


Figure 6.3: The images on the left, in grayscale, were took using TEM microscope. The images on the right, in blue, are images from LM. Images adapted from [Gerber et al. \(2019\)](#) and [Garcia et al. \(2003\)](#), respectively.

species, on the spinal cord level, and on the natural differences between animals. Another important factor is the magnification used to take the pictures; whichever the magnification, the quality of the images obtained also depends on the type and on the brand of scope used to obtain it. For instance, the image processing of a macaque nerve root with an $100\times$ magnification can require more than 200 individual images, each one requiring around 17MB. The next step is a tiling process to join the individual images, obtaining a complete image that usually requires over 300MB of storage space. The final step is the image analysis and quantification of the attributes, which is the subject of the methodologies developed in this chapter.

Two approaches are proposed for the segmentation of the images. Both strategies were developed to avoid the dependence on the images shapes, because such a dependence is the main reason for the difficulties with the techniques that rely on pattern recognition. The guiding principle is to regard the myelinated axons as a solid structure with a closed contour; the proposed strategies strive to identify the contours with a combinatorial optimization perspective and with a graph search technique. These strategies are addressed in the following sections.

6.3 Strategies for myelinated axon segmentation

The usual approach to measure the myelinated axons in a nerve root image is manually identifying the contours of each axon. Following, it is necessary to count the number of axons (the area inside of each dark contour) in each image, to evaluate the area of the axon and the area of the myelin sheath (the dark contours around the axons), to estimate the thickness of each myelin sheath and the diameter of each axon. A complementary analysis appraises the spatial distribution of the axons in the nerve root by identifying the position

of each axon with respect to the complete image. It goes without saying that the whole process is very time-consuming (the analysis of a large nerve root may take more than 40 man-hours of a specialized researcher).

Figure 6.4 illustrates the desired segmentation of a myelinated axon image. The axons were segmented using the ImageJ software (Schneider et al. 2012), where the contours were obtained manually. The yellow curves in Figure 6.4 represent the segmentation lines; the outer contours represent the external limit of the myelin sheaths and the inner contours represent the limit of axon areas, which are also the internal limits of the myelin sheaths. The numbers inside each structure are the labels of the structures; this information allows to locate and to identify each myelin sheath and each axon.

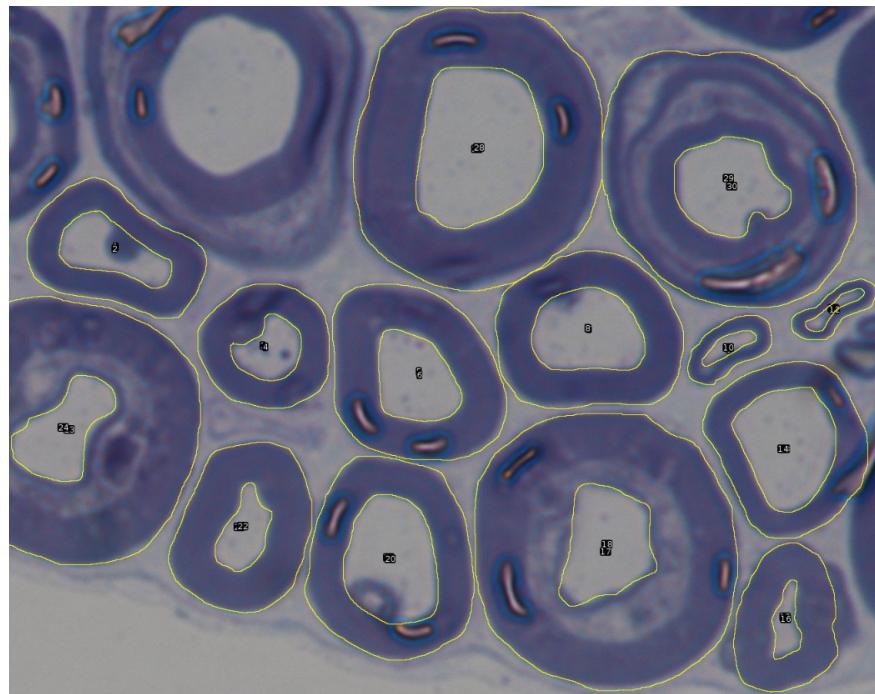


Figure 6.4: A manually segmentation of myelinated axons using ImageJ.

Some strategies for automatic identification and measurement of the myelinated axons have been proposed in the literature (Zaimi et al. 2016, 2018). These strategies are based on pattern recognition techniques that can provide different results, depending on the quality of the images and the method applied to obtain them. The main drawback of these approaches is that the quality of the images have a large variability depending on the species of the tissues, the sizes of the images, and the lab protocols. The main motivation of this research is to achieve a segmentation methodology that do not depend as much on these variable aspects of image processing steps and laboratory equipment.

The next sections propose two alternatives to measure and quantify the attributes of nerve root images; both of them rely on the assumption that a myelin sheath is as a closed

shape. The first approach combines a graph representation of the image with combinatorial optimization techniques. The identification process maps the closed shape element in the original image into a Hamiltonian cycle in the graph representation.

The second segmentation procedure also rely on the myelin sheath as the key information in the segmentation process. In this procedure, the myelin sheath limits the advance of a flood-fill algorithm (a graph search algorithm). The simplicity of the flood-fill algorithm reduces the computational effort with respect to the approach based on combinatorial optimization, allowing it to deal with larger images.

6.4 Approach One – A combinatorial optimization approach

The main goal of the approach presented here is to convert the image analysis problem discussed in the previous sections into a problem of identifying Hamiltonian cycles (an instance of the traveling salesman optimization problem), where myelinated pixels are vertices in the graph connected by the edges to "neighboring" myelinated pixels. These graphs are composed by subgraphs, which can be labeled as an axon if it contains a Hamiltonian cycle.

The identification of the Hamiltonian cycles is an NP-complete problem ([Garey & Johnson 1979](#)). The use of the relaxation proposed in Chapter 3 can bring innovative aspects to the Hamiltonian cycle formulations, allowing to increase the size of instances solvable by quadratic optimization solvers, such as Ipopt ([Wächter & Biegler 2009](#)) and Knitro ([Waltz & Plantenga 2011](#)).

6.4.1 Hamiltonian cycle

The problem of finding a Hamiltonian cycle is an instance of the traveling salesman problem where the distance c_{ij} between two cities i, j is one if they are directly connected and infinity otherwise. If the optimal value of the objective function is equal to the number of cities (n), there is a Hamiltonian cycle in the graph; however it may include undesirable sub-tours.

The model (6.1)-(6.4) presents a integer linear programming formulation for the Hamil-

tonian cycle, in which sub-tours are not avoided.

$$\min \sum_{i=1}^n \sum_{j \neq i, j=1}^n x_{ij} \quad (6.1)$$

$$\text{s.t. } \sum_{i=1, i \neq j}^n x_{ij} = 1; \quad j = 1, \dots, n \quad (6.2)$$

$$\sum_{j=1, j \neq i}^n x_{ij} = 1; \quad i = 1, \dots, n \quad (6.3)$$

$$x_{ij} \in \{0, 1\}; \quad i, j = 1, \dots, n \quad (6.4)$$

where $x_{ij} = 1$ if the arc $i \rightarrow j$ is part of the Hamiltonian cycle, zero otherwise. Equation (6.1) calculate the objective function. Constraint sets (6.2) and (6.3) assure that each arc participates only once in the composition of the Hamiltonian cycle. However, these two constraint sets are not able to avoid sub-tours in the solution. The sub-tours can be avoided with the additional constraints (6.5)-(6.7), called Miller–Tucker–Zemlin (MTZ) constraints (Miller et al. 1960).

$$u_i - u_j + nx_{ij} \leq n - 1; \quad i = 2, \dots, n, \quad j = 2, \dots, n \quad (6.5)$$

$$0 \leq u_i \leq n - 1; \quad i = 1, \dots, n \quad (6.6)$$

$$u_i \geq 0; \quad i = 1, \dots, n \quad (6.7)$$

Although the MTZ constraints introduce a new set of integer variable u_i , this is counter-balanced because the number of constraints introduced by (6.5)-(6.7) to control the sub-tour increases only as a polynomial with the number of nodes in the graph. The polynomial number of constraints is a positive side of the MTZ approach that can be associated with the relaxation results presented in Chapter 3. This hybrid strategy can provide an alternative to other sub-tour elimination techniques that require an exponential number of constraint (Bektaş & Gouveia 2014, Applegate et al. 2006).

Therefore, the mathematical model (6.1)-(6.7) was relaxed by adding the proposed penalty term in the objective function

$$\min \sum_{i=1}^n \sum_{j \neq i, j=1}^n x_{ij} + \beta \sum_{i=1}^n \sum_{j \neq i, j=1}^n (x_{ij} - y_{ij})^2,$$

considering $x_{ij} \in [0, 1]$, $y_{ij} \in [0, 1]$, and β is a positive constant large enough to guarantee the binary feasibility (as discussed in Chapter 3).

The Hamiltonian cycle problem is part of the segmentation procedure described in the

following section, where it identifies the myelin sheath. The idea behind this application is to consider as a myelin sheath the graphs with closed cycles, that represents the closed shapes in the original images.

6.4.2 Steps for the image segmentation using Hamiltonian cycles

The segmentation procedure described in this section considers the image interpreted as a graph. The myelinated pixels in the original image become nodes of the graph, and the neighbor pixels form the edges of the graph if they are also myelinated pixels. For a graph to be considered as an axon representation, the graph has to contain a Hamiltonian cycle, i.e., there is a path that visits each vertex exactly once and the starting and ending point are the same.

The computational effort required by the segmentation approach is related to the number of pixels in the image; for this reason, the first step to identify Hamiltonian cycles in the image is a preprocessing procedure to reduce the image dimension. The resizing procedure is followed by a re-scaling process to preserve the original scale. Since there are axons from $0.1\mu m$ up to $40\mu m$ in the same image, the choice of a single resizing parameter can lead to poor image conversion. For this reason, the resizing procedure uses a sequence of resizing parameter r_k based on the myelin thickness. The value r_k is set to the number of pixels in the largest sequence of myelinated pixels in the image – considering sequences in the vertical or horizontal orientation of the image.

The proposed identification algorithm runs one iteration for each (k), from the thicker to the thinner myelin sheath. Each time that a myelin structure is identified, it is removed from the original image, and it will not be part of the next resizing and segmentation process. The idea of this protocol is to avoid double-counting and improve the segmentation algorithm performance.

The axon identification process stops when the resized image does not present myelinated pixels; following, a new segmentation process is achieved for the next resized image. The overall process stops after running the segmentation procedure for the last r_k parameter. The descriptive statistic for each axon is computed with a pixel counting, considering each r_k value; this measurement process provides the area, the diameter, and the myelin thickness of the axon. Additional information such as the inner area of the myelin sheath, and the total axon area are also computed. The whole segmentation process using the Hamiltonian cycle computation is summarized in the Algorithm 3.

```

Result: List of the axons
for dimension of  $r$  do
    resize the image using the reduction factor
    create the complete graph using the resized image
    candidates  $\leftarrow$  list of subgraphs
    while candidate_list  $\neq \emptyset$  do
        perform a Hamiltonian cycle verification
        if candidate is a Hamiltonian cycle then
            axon_list  $\leftarrow$  candidate;
            remove candidate of the candidate_list
        end
        remove candidate pixels from the original image;
        remove candidate pixels from the resized image;
    end
end

```

Algorithm 3: Approach One – axon identification based on Hamiltonian cycle.

The next subsection presents the computational studies with the proposed relaxation for the Hamiltonian cycle problem proposed in Section 6.4.1, followed by studies of the application of this idea to the segmentation of nerve root images. The first part of the subsection presents a comparison between the original and relaxed formulation of the Hamiltonian cycle problem; the second part presents the results of the image segmentation considering Approach One.

6.4.3 Computational studies

The first part of the section analyzes the benefits of the proposed relaxation for the Hamiltonian cycle problem. It starts with the description of instance set used during the evaluation, followed by the computational experiments comparing the integer and the relaxed formulation of the Hamiltonian cycle problem. The second part of the section illustrates the performance of Approach One. It is discussed the resizing process applied in Approach One and the quality of the segmentation, as well as, the computational effort required by the segmentation process.

Hamiltonian cycle explorations

The instances considered during the computational experiments for the Hamiltonian cycle come from the Knight Tour problem. The choice for the Knight tour problem instances is due to the similarities between the number of feasible moves for a Knight in a chessboard and the number of possible neighbors of a myelinated pixel. In both cases the maximum number of edges starting in a node is eight, with the exception of the borders (for both the

chessboard and the image).

The Knight tour problem is an NP-hard problem in the general case — there are polynomial-time algorithms for some particular cases (Pohl 1967). The challenge is to find a move sequence that allows visiting all the square of a chessboard only once. If the sequence starts and ends in the same square of the chessboard, it is called Closed Knight tour; otherwise, the sequence is an Open Knight Tour. The Closed Knight tour problem can be interpreted as a Hamiltonian cycle problem (Pohl 1967), while the Open Knight Tour is a Hamiltonian Path problem (Conrad et al. 1994). Figure 6.5 illustrates both cases for a 8×8 chessboard; images adapted from the American Federation of Chess website ¹.

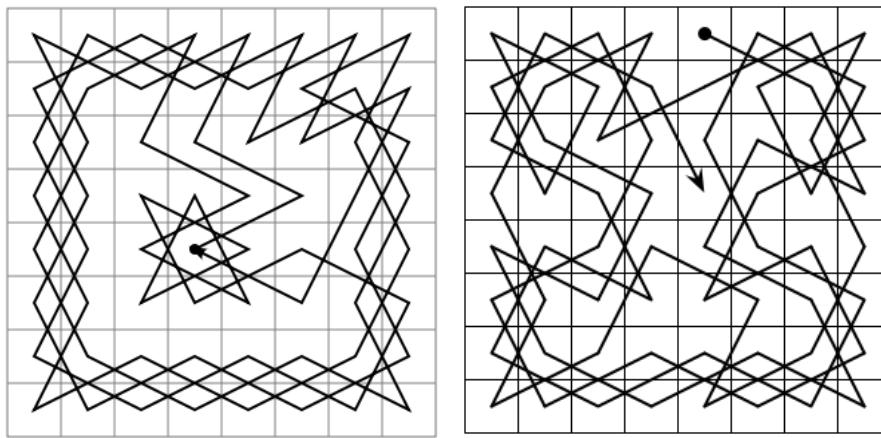


Figure 6.5: Illustration of a Closed Knight tour (left) and an Open Knight tour (right), for a 8×8 chessboard.

One of the first mentions of a systematic solution study for the Closed Knight Tour problem goes back to Leonard Euler in 1759 (Takefuji 1992). During the last decades, different solution strategies were proposed for the problem, such as evolutionary algorithm (Gordon & Slocum 2004), ant colony (Delei et al. 2009), and neural networks (Takefuji 1992). Further studies addressed the feasibility of the problem for rectangular (Schwenk 1991), irregular shapes, and 3D chessboards (Bai et al. 2010).

In the following computational studies for the Closed Knight Tour is interpreted as a graph problem, where each square of the chessboard is represented by a node in a graph. The edges that emanates from a given node are the feasible moves for the Knight from the position represented by the node.

The instances generated for these studies consider rectangular chessboards and Closed Knight Tour problem. Schwenk (1991) proved that for any $m \times n$ chessboard with $m \leq n$, a closed tour is always possible for these cases unless one of the following conditions is satisfied:

¹<https://new.uschess.org/home/>

- m and n are both odd;
- $m = 1, 2$, or 4 ;
- $m = 3$ and $n = 4, 6$, or 8 .

Avoiding the above conditions, square instances were generated with the following attributes: $m = n$, $n \geq 6$, and n an even number. The computational experiments evaluate the benefits of applying the relaxation proposed in Section 6.4.1 to identify Hamiltonian cycles represented by these Closed Knight Tour instances. The computational experiments use the mathematical modeling language JuMP ([Dunning et al. 2017](#)) to codify the model; solution were obtained with the solver Knitro ([Waltz & Plantenga 2011](#)). The studies compare the computational times required by the solution of the problem formulated in (6.1)-(6.7), which uses binary variables, and the formulation with the relaxation approach.

Table 6.1 presents the results for nine instances of the Closed Knight Tour problem, generated as described above. Column "n" gives the dimension of the chessboard, columns "Nodes" and "Edges" give, respectively, the number of nodes and the number of edges in the graph for each instance. Columns "Time(s)" give the total execution time in seconds. The information using binary variables is given in the column "Integer" and the times with the relaxed formulation are given in the column "Relaxed".

n	Nodes	Edges	Time(s)	
			Integer	Relaxed
6	36	160	0.47	1.29
8	64	336	1.85	3.24
10	100	576	3.73	7.58
12	144	880	9.28	17.65
16	256	1680	**	46.15
20	400	2736	**	173.63
22	484	3360	**	551.23
24	625	4048	**	1679.84
30	900	6496	**	**

** The instances without a solution within 3600 seconds.

Table 6.1: Computational times for the Hamiltonian cycle identification.

Note from Table 6.1 that for small instances the performance of the formulation with binary variables performs better than the formulation with relaxed variables. However, the formulation with binary variables is not able to solve instances with n larger than twelve (within the allowed 3600 seconds). In other words, the approach with binary variables was able to solve instances with up to 144 nodes and the relaxed approach allowed to solve instances with up to 625 nodes.

Image segmentation using Hamiltonian cycles

The analysis of image segmentation with Hamiltonian cycles requires the sequential use of a resizing process. Figure 6.6 illustrates this resizing process.

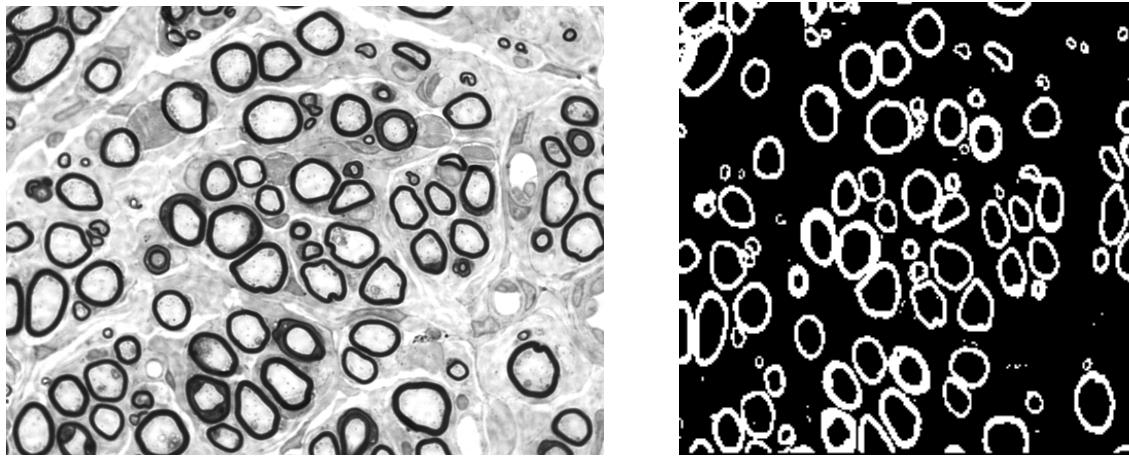


Figure 6.6: Image reduction – left 84.134 bytes and right 836 bytes.

Note that both images present axon structures that are very close to each other, or even touching each other. Both cases can lead to errors during the Hamiltonian cycle verification, and the probability of having these errors increases due to the resizing process, which forces the structures to appear even closer in the resized images. Structures touching each other can create a node in the graph that may not exist in the original image, which entail errors in the segmentation process.

Computational experiments show that Algorithm 3 is able to identify the myelin sheath correctly. The segmentation procedure correctly avoids the incomplete structures in the border of the image. It was also able to remove pixels in the inner axon area that could be mistaken by myelinated axons pixels. Figure 6.7 presents the axons identified by Algorithm 3 requiring 2432 seconds to complete the process. In a rough estimation, the computational time required by the Hamiltonian cycle verification used two-thirds of the total processing time.

A word of cautions is necessary concerning the required resizing process. The successive resizing can also suppress pixels from small structures, causing some loss of quality in the approach.

In a rough estimation, the computational time required by the Hamiltonian cycle verification comprehended two-thirds of this amount. This high computational cost limited the use of this approach in full-size images, requesting a new solution approach.

The segmentation idea discussed in the next section is based in a graph search tech-

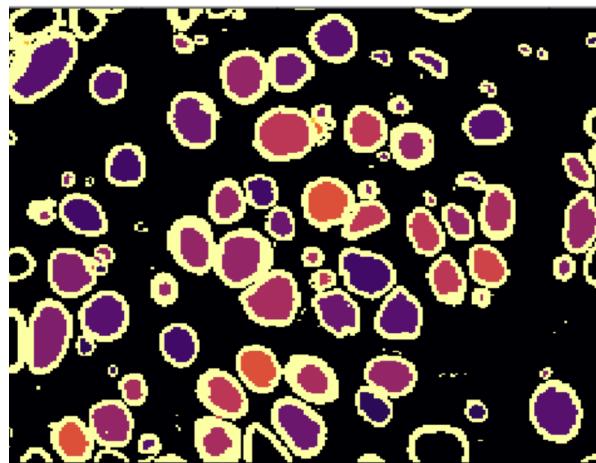


Figure 6.7: Results for the image segmentation using Algorithm 3. The processing time required was about 40 minutes.

nique that is less demanding in computational effort. This saving will allow to address the myelinated axon segmentation problem without the resizing process.

6.5 Approach Two – A flood-fill approach

As mentioned above, the flood-fill approach was developed to handle full-size images, allowing to avoid the information losses entailed by the resizing process. In this approach, the segmentation ideas uses the bounders of the axon areas (the contours defined by the myelin sheath) to limit the node coloring process of the flood-fill algorithm. Similarly to Approach One, it is also able to avoid the problems related to pattern recognition techniques.

The flood-fill approach is composed of two main steps. The first step is the image preprocessing, where the initial color image is converted into a black-and-white image. The second step is the application of a flood-fill algorithm to perform the segmentation of the myelinated axons. The next sections discuss these steps.

6.5.1 Image preprocessing

This preprocessing step converts the original images into black-and-white images, in order to avoid problems due to difference in colors that appear in the images provided by different types of scopes. Consequently, the preprocessing allows to handle color, grayscale, and black-and-white input images.

Indeed, the only requirement to assure the quality of the preprocessing step is that in the original image, before the preprocessing, the myelin sheath appears darker than the background. However, the conversion of the original image into a black-and-white image is

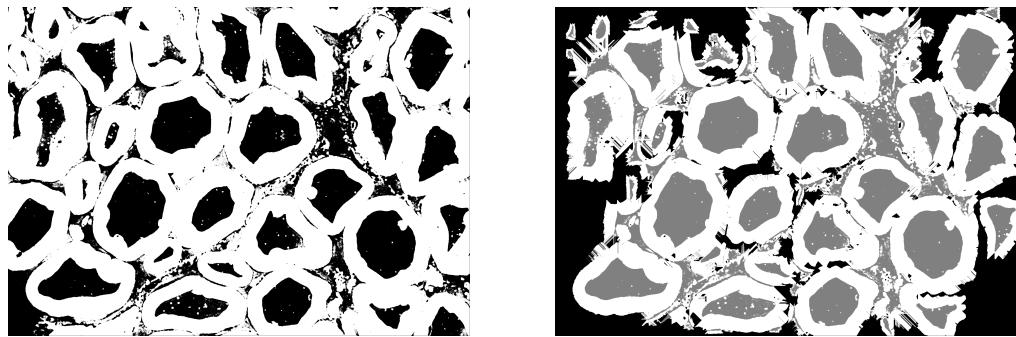


Figure 6.8: The left image provide the deficient conversion of the original image, where background pixels were labeled as a myelin. One the right image, there structures identified as an axon are presented in gray.

based on a single parameter μ . This parameter defines a threshold, above which a pixel is converted into a black; otherwise, the pixel is converted into white.

The choice of the parameter μ is essential to obtain a successful segmentation approach. The choice of a good value for this parameter is a trial and error procedure that is fundamental for the success of the whole approach. Indeed, low-quality results in the segmentation process are usually a consequence of the loss of information in converting color images to black-and-white, because a wrong choice of the parameter μ creates distortions in the image in the black-and-white images (by adding extra white or black pixels).

Figure 6.8 gives an example of these distortions. The left-hand side of figure shows a black-and-white picture with fake white pixels; these fake white pixels lead to the identification of false closed structures by the flood-fill algorithm, illustrated in the right-hand side of the figure. These errors entail wrong axon counting and wrong measurements.

Figure 6.9 illustrates another wrong choice of the parameter μ , allowing myelinated pixels to be converted into black pixels. The gaps in the myelin sheath can create fake structures, as illustrated in the right-hand figure. These errors also lead to other wrong measurements, such as evaluating internal areas larger than the real ones.

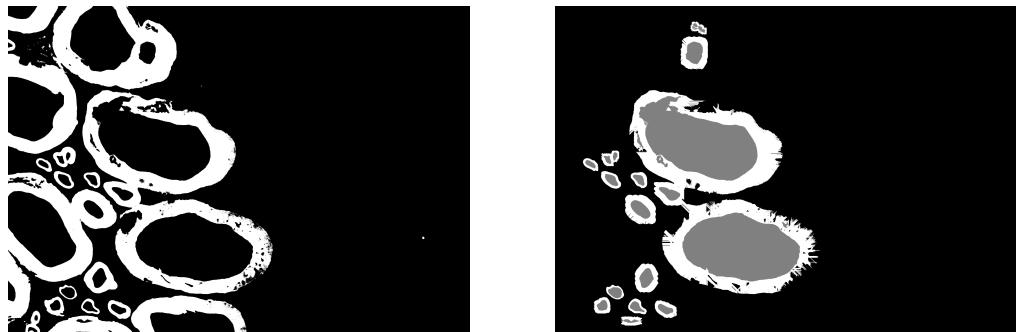


Figure 6.9: The left image provide the deficient conversion of the original image, where pixels of the myelin were not in white. One the right image, there structures identified as an axon are presented in gray.

6.5.2 Flood-fill algorithm

The flood-fill algorithm is a well-established breadth-first search for image segmentation ([Pavlidis 1979](#)) with computational complexity $O(n)$, where n is the number of pixels to be colored ([Clifford et al. 2012](#)). The algorithm represents the pixels of the image as a graph, with neighbors pixels leading to connected nodes. This application considers eight neighbors for each myelinated pixel, as illustrated in Figure [6.10](#).

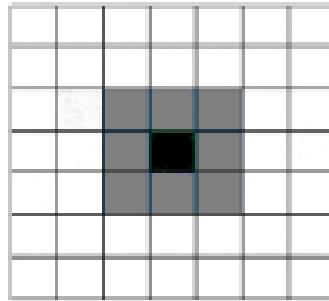


Figure 6.10: Neighbor for the flood-fill algorithm considering eight neighbors.

The flood-fill process for the myelinated axon segmentation starts in a black pixel of the image and explores the neighborhood of this pixel. Each neighbor pixel is converted into a gray pixel, and the process is started from each new gray pixel. The process stops when there is no other black neighbor to be colored.

The algorithm explored in this research uses two stop-criteria for the flood-fill algorithm. The first criterion identifies that there are no feasible neighbors. This case occurs due to one of the three following reasons: (1) the flood-fill process started inside of a closed shape, (2) all the feasible neighbors were already explored, (3) or the process started in a background pixel. Although the absence of feasible neighbors could also occur because all the possible pixels of the image were explored, the additional stopping criteria try to avoid reaching such a scenario that would entail a high computation effort (due to a large number of pixels to be visited). The additional stopping criterion halts the flood-fill process when a pixel in the border of the image is visited — the rationale of this stop-criteria is to try to provide an early identification for the cases where the initial pixel is inside an open structure, or it is outside an axon.

The following subsection presents the whole segmentation procedure with the Approach Two.

6.5.3 Image segmentation using the flood-sill algorithm

The flood-fill algorithm in the image segmentation approach starts with a list of pixels (*candidate*) with the same color of the background. Picking a pixel in the list of *candidate* the algorithm verifies if the pixel is connected to a border pixel. If it is connected to a border pixel the variable *touch_border* is true; otherwise, *touch_border* is false. The idea of the *touch_border* variable is to identify incomplete axons, which can occur in the cases where the myelin structure is incomplete (because it is in a border of the image) or in the cases where the laboratory processing caused distortions in the myelin sheath. The *candidate* pixels for which *touch_border* is false are inserted in the *axon_list*; starting from each axon in *axon_list*, the algorithm seeks to identify all the pixels belonging to the myelin sheath of this axon. The identification algorithm stops when the *candidate* list is empty. The whole identification process with the flood-fill algorithm is summarized in the Algorithm 4.

Result: Measurement of axons and myelin sheath

```

while background_list ≠ ∅ do
    candidate, touch_border ← flood-fill algorithm
    if touch_border = false then
        axon_list ← flood-fill(candidate);
        myelin ← Myelin sheath detection;
        execute measurements;
    end
    remove candidate from the background_list
end
```

Algorithm 4: Approach Two – axon identification based on flood-fill algorithm.

A positive side of the flood-fill algorithm to the application in nerve root image segmentation is that it has a linear complexity (Clifford et al. 2012), $O(n)$, where n is the number of pixels to be colored. This property allows the Algorithm 4 to handle images with thousand of axons, and avoids the application of the resizing techniques required by the Approach One.

Figure 6.11 presents a segmentation plotted above the original image. Comparing Figure 6.7 and Figure 6.11, it is possible to note a better segmentation quality on the image provided by the Algorithm 4. For instance, Figure 6.11 presents less distortion on the axons shape, mainly on the borders, and the relationship between the axon area and the myelin thickness are close to reality. However, a word of caution is necessary because the approach is sensitive to fake pixels inside the axon area and noises that appear as discontinuities in

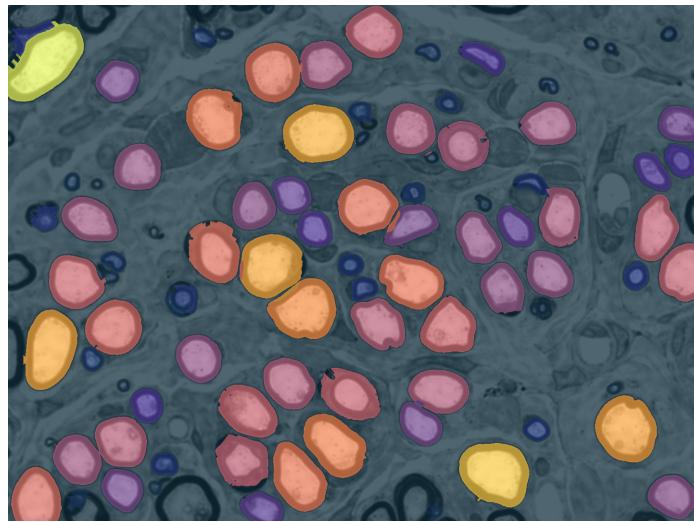


Figure 6.11: Axons identified by the Algorithm 4, in less than 50 seconds.

the myelin sheath.

Implementation

The Algorithm 4 was implemented in Julia language ([Bezanson et al. 2017](#)). Because the segmentation process can sometimes present inaccuracies, a human guided correction process was also implemented as a component of the whole process. Essentially, the correction step allows a specialist to correct the labels provided by the program, completing and removing structures. After this human step, the measurements are updated, running again the Algorithm 4.

The computational tests were performed in a PC desktop running in a Ubuntu 16.08 environment, using an i7 processor and 16GB of memory. Because the main objective of the computational studies is to analyze the quality of the automatic segmentation tool, the following results do not apply the correction step.

6.5.4 Computational studies

The strategy devised to evaluate the quality of the myelinated axon segmentation delivered by the Algorithm 4 compares the results with reference images provided by a specialist in the manual myelinated axon segmentation process (ground truth images). The parameter μ discussed in Subsection 6.5.1 was set to $\mu = 0.4$ for all images; this choice was defined by exploratory trial and error experiments. The computational tests use 39 images, with a total number of pixels above 10^8 — these real images were obtained in studies performed at the "Brain Research Institute" of the "University of California - Los Angeles".

The specialist images (used as yardsticks) were also obtained at the "Brain Research

Institute" of the "University of California-Los Angeles", under the supervision of Prof. Leif Havton. An extensive description of these images is presented in the paper mentioned in the Appendix A, currently in preparation.

The prediction quality was evaluated under five metrics proposed in the literature for image comparisons (Crum et al. 2006). Namely, these metrics are the Accuracy, the True Positive Rate (TPR), the Positive Predictive Value (PPV), the Sørensen–Dice coefficient (DSC), and the Jaccard index (Jaccard). The parameters used to compute the values of these metrics, given by the equations (6.8) – (6.12), are the number of pixels correctly identified (TP), the number of pixels incorrectly identified (FP), the number of pixels correctly rejected (TN), and the number of pixels incorrectly rejected (FN).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6.8)$$

Accuracy summarizes the percentage of total items correctly classified; the higher the accuracy value, the better the classification. However, the accuracy value can be misleading whenever the number of background pixels is much larger than number of myelinated pixels in the image; the symmetric case, when the number of myelinated pixels is much larger than the number of background pixels, can also causes misleading values for the accuracy.

$$TPR = \frac{TP}{TP + FN} \quad (6.9)$$

The True Positive Rate (TPR) is also known as recall or sensitivity metric. It measures the number of pixels correctly identified as positive, out of the total true positive pixels.

$$PPV = \frac{TP}{TP + FP} \quad (6.10)$$

The Positive Predictive Value (PPV) is also known as precision metric. The PPV is the percentage of pixels correctly identified as positive out of the total number of pixels identified as positive. The higher the TPR, the lower the number of false-positive identifications. This metrics measures the number of false-positive results, i.e., pixels identified as an axon when they are background pixels, or vice-versa.

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (6.11)$$

The Sørensen–Dice coefficient (DSC) is also known as F-1 score. The DSC is the harmonic mean between the TRP and the PPV. In the same way that TRP and PPV, the higher the

DSC value, the better the result. It means that there is a small number of false-negative and false-positive classifications.

$$Jaccard = \frac{TP}{TP + FP + FN} \quad (6.12)$$

The Jaccard index is a measure of similarity between the ground truth image and the predict image, in a zero to one range. The higher the index, the more similar are the two images.

Table 6.2 presents on the rows "Average" and "Standard Deviation" the average indices values in and standard deviation for the set 39 images. The maximum and minimum value observed for each index is presented on the rows "Maximum" and "Minimum".

	Accuracy	TRP	PPV	DSC	Jaccard
Average	0.906	0.854	0.826	0.830	0.728
Standard Deviation	0.053	0.060	0.170	0.140	0.161
Maximum	0.989	0.982	0.974	0.978	0.958
Minimum	0.756	0.739	0.147	0.246	0.140

Table 6.2: Comparison values of the measurements to analyze the quality of the segmentation Algorithm 4.

The analysis of Table 6.2 shows that all the metrics present high average value, indicating a good overall performance of the Approach Two. Especially, the metrics Accuracy and TRP present robust results, with high average value, small standard deviation, and with the worst classification case close to 0.75.

The PPV metric presents the largest standard deviation among all the metrics tested, which indicate that the number of false-positive classification can be higher in some instances. This is an expected behavior for the cases where the preprocessing phase entails loss of information, as illustrated in Figure 6.8. An ad hoc choice of the parameter μ could handle these cases.

When analyzing the Jaccard index, the results from the segmentation approach presents more than 70% of similarity with the ground truth image on the average. Performing an individual analysis of the images with lower value of Jaccard index, these cases occurred for images where the choice of the value μ entails distortions, as illustrated in Figure 6.9. Again, an ad hoc choice of the parameter μ could improve the results.

6.6 Discussion

The myelinated axon segmentation process that have been adopted in neuro research area is a time-consuming and human-dependent task that requires skilled professionals. This chapter proposed two different strategies to automate these processes.

The first strategy is based on combinatorial optimization. It used graphs to represent the nerve root image and Hamiltonian cycles to identify the closed shapes created by the myelin sheaths. Computational tests showed the advantages of using the proposed relaxation in the formulation of the Hamiltonian Cycle problem; the quadratic relaxation allowed to solve instances larger than the instances handled by the formulation considering binary variables. The first tool was able to segment test images created by parts of the original nerve root image; however, the use of this approach in real cases of the myelinated axon segmentation processes required a repeated resizing steps, which may cause undesirable noises in the information.

The second strategy uses a preprocessing phase to convert the input image into a black-and-white image, followed by the application of a flood-fill algorithm to identify and segment axon structures. This strategy was able to handle real-size images without requiring resizing steps, however the success of the preprocessing phase depends on the choice of a parameter μ , defined by trial end error procedures.

The outputs of the second strategies were compared with references images provided by specialists, using five well-established metrics for image comparisons. The case studies showed that the automate segmentation provides good quality results in most of the cases. For the cases with a low quality segmentation, the results can be improved by using ad hoc values for the parameter μ . Furthermore, the results can be improved by applying human correction steps; indeed, the inclusion of the correction steps allows to achieve one hundred percent accuracy, but they were not used in the case studies to better evaluate the contribution for the automation of the process.

As a final remark, it should be observed that methodologies described in this chapter provided automatic alternative to handle the segmentation of myelinated axon in nerve root image, based on combinatorial optimization and graph search algorithms. Considering the cases studies developed to evaluate the methodologies, the second approach seems more promising from a computational effort perspective, because it avoids a time consuming resizing process required by the first approach, in order to handle real images.

However, both alternatives should still be regarded as on-going researches. A more

extensive evaluation of the positive aspects and limitations of the methodologies should be performed by neuro scientists, considering the total number of hours required for the analysis and the quality of the final results.

Unassigned distance geometry problem

This chapter proposes an optimization model for the unassigned distance geometry problem (uDGP) — some of the ideas discussed in the chapter appeared in the paper "A new quadratic relaxation for binary variables applied to the distance geometry problem" ([Bartmeyer & Lyra 2020](#)). The uDGP is a problem in the distance geometry area that seeks for the best assignment of each vertex of a molecule to a three-dimensional Euclidean space; the information available is the number of vertices in a graph and a list of distances between the vertices. In a broader perspective, the uDGP aims to reconstruct the structure of molecules and proteins by defining the spatial position of each atom (vertex) of these structures.

The model includes binary and continuous variables, in addition to linear and quadratic constraints. The intrinsic difficulty of the uDGP makes it a severe testbed to evaluate the ideas presented in Chapter 3. Computational studies illustrate how the relaxation is able to increase the size of solvable instances for the uDGP using the proposed formulation. They also explored the influence of the Lagrangian relaxation parameter in the binary feasibility of the problem.

The literature about the uDGP is still incipient, making it an open area of research ([Liberti & Lavor 2018](#)). Improvements in solutions strategies for the problem can bring benefits to applications that are arising in the different areas, such as robotics ([Porta et al. 2018](#), [Rojas & Thomas 2013](#)), design of structures, nano-technology, and bio-engineering ([Liberti & Lavor 2018](#)).

7.1 Introduction

The distance geometry problem (DGP) seeks for the spatial position of the n vertices of the graph in the Euclidean space \mathbb{R}^k given a list of distance (\mathcal{D}) between the vertices. A solution to the DGP provides the spatial position for the vertices of a given graph.

The DGP is usually classified into two categories. The first category refers to the assigned distance geometry problem (aDGP), for which the Euclidean distance between each pair of vertices is known a priori. The second category deals with the cases for which the relation between each pair of vertices and the distance between them are not known a priori; this information should also be provided by the solution of the uDGP.

Figure 7.1 illustrates the aDGP and the uDGP. Each entry d_{ij} of the matrix on the left-hand side of the figure represents the Euclidean distance between the vertex i and the vertex j . A solution for the aDGP should give the spatial position of the vertices for which the distances between them are represented in the matrix. For instance, a feasible solution (S) for the example are the positions $S = \{(0, 1); (1, 1); (0, 0); (1, 0)\}$.

The list of values represented on the right-hand side of Figure 7.1 illustrates the information available for the solution of the uDGP. A solution to the problem should obtain the best match of the values in the list to the distance between each pair of vertices, and also give the spatial position of the vertices; for instance, providing the matrix of distances between vertices (as represented on the left-hand side) and the solution $S = \{(0, 1); (1, 1); (0, 0); (1, 0)\}$.

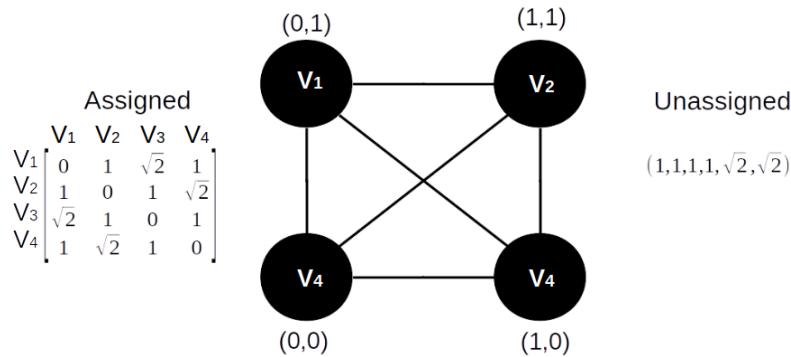


Figure 7.1: Image adapted from the thesis of S.R. Gujarathi

A formal definition for the aDGP can be given, as in Billinge et al. (2016). Given a graph $G = (E, V)$, a set of edges $(i, j) \in E$, a set of vertices $i \in V$, a solution to the aDGP provides the spatial position of the vertices; each solution to the problem is called a realization.

The aDGP can be stated as an optimization problem, that should minimize the objective

function represented in Equation (7.1).

$$\min \sum_{i=1}^n \sum_{j=1}^n (d_{ij}^2 - \|x_i - x_j\|^2)^2 \quad (7.1)$$

where d_{ij} is the distance assigned to the edge $(i, j) \in E$, x_i and x_j are the position of the nodes $i \in V$ and $j \in V$ in the Euclidean space \mathbb{R}^k . For this formulation, an optimal solution is achieved if, and only if, the objective function value is equal to zero.

The first formulation for the uDGP was proposed in the paper "A new quadratic relaxation for binary variables applied to the distance geometry problem" (Bartmeyer & Lyra 2020), and it will be discussed in the next sections. Indeed, the uDGP is has been considered an open-problem in the field of geometry distance (Liberti & Lavor 2018).

Before going on to discuss the uDGP, a word about applications is in order. Both the aDGP and the uDGP problems find applications in protein conformation (Moré & Wu 1999, Liberti et al. 2011), nanoparticle structures (Billinge et al. 2016), and sensor location for wireless networks (Cao et al. 2006, Biswas & Ye 2006).

7.2 The unassigned distance geometry problem

The pieces of information provided for addressing the uDGP are the list of distances \mathcal{D} , the number of vertices n , and the dimension of the Euclidean space k . A solution for the uDGP should provide the assignment of each vertex $i, j \in V$ to a single edge $(i, j) \in E$ of length $d_a \in \mathcal{D}$, and the positions of the vertices in the Euclidean space \mathbb{R}^k .

An exact solution to the uDGP has to respect the condition that each Euclidean distance between the vertices, $\|x_i - x_j\|$, is assigned to one of the entries d_a in the distance list \mathcal{D} . For the cases where there is imprecise information in the distance list, it is possible to soften the previous condition as $\underline{d}_a \leq \|x_i - x_j\| \leq \bar{d}_a$, where \underline{d}_a and \bar{d}_a are the lower and upper bounds for the difference between the distance of vertices x_i, x_j and the assigned distance d_a .

Previous approaches for the uDGP are two hybrid heuristics, both using building-up ideas (Duxbury et al. 2016). The TRIBOND approach (Gujarathi et al. 2014) starts with a set of cliques of size $k + 2$ satisfying the distance list. For each clique in the set, named *core*, the vertices are added one by one, always considering the feasibility. The build-up process stopped when all the vertices were added or if there is no feasible choice for the next vertices; for the second case, the process restarts from a new clique of the *core* set.

The heuristic LIGA (Juhás et al. 2006) is a stochastic algorithm able to handle imprecise

or incomplete information on the list of distance. It is a hybrid approach that starts with a *core*, and uses cost functions and optimization to select the next vertex.

Although the heuristic approaches have promising results for the uDGP, the design of exact methods has been investigated during the last two decades. Some of the ideas explored are the spatial branch-and-bound (Liberti & Maculan 2006), neighborhood search (Hansen & Mladenović 2001), successive smoothing (Moré & Wu 1999), and an stochastic approach proposed by Huang & Dokmanić (2020).

7.3 Mathematical model

The proposed mathematical model to address the uDGP search to unveil the best assignment of the distances to the edges of the graph, and to define the position of each vertex of the graph in the \mathbb{R}^k space. The variables used in the model can be split into three sets. The first set comprises the location variables $x_i \in \mathbb{R}^k$ that provide the spatial position of the vertices. The second set of variables contains the variables $y_{aij} \in \{0, 1\}$ that assigns each edge (ij) of the graph to a distance d_a in the list \mathcal{D} . The third set contains the variables $p_{ij} \in \mathbb{R}_+$ and $n_{ij} \in \mathbb{R}_+$, which are, respectively, the positive and negative difference between the distance d_a and the Euclidean distance of the vertices x_i, x_j . The distance list may contain repeated values; therefore, the model includes a parameter z_a that defines the multiplicity of each distance d_a — the use of this parameter is not mandatory for the uDGP, but it conveys information that can enhance computation strategies. The benefits of exploring the multiplicity in the distance list are evaluated in the computational experiments.

The mathematical model is stated in the Equations (7.2)-(7.6).

$$\min \quad \sum_{i=1}^n \sum_{j=1}^n (p_{ij} + n_{ij}) \quad (7.2)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j=1}^n (p_{ij} - n_{ij} + \|x_i - x_j\| - d_a) y_{aij} = 0, \quad \forall a = 1, \dots, m \quad (7.3)$$

$$\sum_{a=1}^m y_{aij} = 1, \quad \forall i, j = 1, \dots, n \quad (7.4)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{aij} = z_a, \quad \forall a = 1, \dots, m \quad (7.5)$$

$$n_{ij} \geq 0, \quad p_{ij} \geq 0, \quad x_i \geq 0, \quad y_{aij} \in \{0, 1\}, \quad n_{ij} \in \mathbb{R}, \quad p_{ij} \in \mathbb{R}, \quad x_i \in \mathbb{R}^k \quad (7.6)$$

The objective function (7.2) represents the total deviation, computed as the sum of the variables p_{ij} and n_{ij} . The constraint set (7.3) define the value of the deviation p_{ij} and n_{ij} ;

the number of constraints in this set is given by the number of distinct distances (m), and not by the total number of distances. The constraint set (7.4) guarantee the assignment of each pair of vertices to a single distance. The constraint set (7.5) assign z_a vertices to the distance d_a distance, where z_a is the multiplicity of the distance d_a .

The mathematical model (7.2)-(7.6) is a nonlinear nonconvex mixed-integer optimization problem. Due to the nonconvexity of the problem, optimal global solutions can only be achieved with global solvers, that usually can not handle large instances of the problem. Also, the binary variables entail high computational effort. However, the binary variables can be relaxed using the ideas presented in Chapter 3.

The relaxation is achieved by adding the term $\sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (y_{aij} - w_{aij})^2 = mn^2$ to the objective function, and using the penalty parameter β to control the binary feasibility of the solution. The formulation (7.7)-(7.11) contains a nonlinear convex objective function and a nonlinear and nonconvex constraint set. A positive aspect of this model is the absence of binary variables; however, as mentioned above, optimal global solutions can only be assured with a global solver.

$$\min \quad \sum_{i=1}^n \sum_{j=1}^n (p_{ij} + n_{ij}) + \beta(mn^2 - \sum_{a=1}^m \sum_{i=1}^n \sum_{j=1}^n (y_{aij} - w_{aij})^2) \quad (7.7)$$

$$\text{s.t.} \quad \sum_{i=1}^n \sum_{j=1}^n (p_{ij} - n_{ij} + \|x_i - x_j\| - d_a) y_{aij} = 0, \quad \forall a = 1, \dots, m+1 \quad (7.8)$$

$$\sum_{a=1}^m y_{aij} \leq 1, \quad \forall i, j = 1, \dots, n \quad (7.9)$$

$$\sum_{i=1}^n \sum_{j=1}^n y_{aij} = z_a, \quad \forall a = 1, \dots, m \quad (7.10)$$

$$n_{ij} \geq 0, p_{ij} \geq 0, x_i \geq 0, y_{aij} \in [0, 1], w_{aij} \in [0, 1], n_{ij} \in \mathbb{R}, p_{ij} \in \mathbb{R}, x_i \in \mathbb{R}^k \quad (7.11)$$

The performance of the formulation (7.7)-(7.11) was evaluated with the computational studies described in the next section.

7.4 Computational studies

The computational studies evaluate the behavior of the relaxed formulation with respect to the parameter β , and the improvements in the computational performance provided by the information about the multiplicity of the distances. These evaluations consider a single instance from the literature, and a set of instances generated with the procedure described in Lavor (2006). Computational tests were performed in a desktop PC (Ubuntu 18.04 op-

erational system, i7 and 32GB of memory) using the AMPL language ([Fourer et al. 2003](#)) with the solver Knitro version 7.0 ([Waltz & Plantenga 2011](#)).

The quality of solutions are evaluated with respect to the total deviation and to the average deviation, defined, respectively, by Equation 7.12 and by Equation 7.13.

$$Total = \sum_{i=1}^{n-1} \sum_{j=i}^n (p_{ij} + n_{ij}) \quad (7.12)$$

$$Average = \frac{\sum_{i=1}^{n-1} \sum_{j=i}^n (p_{ij} + n_{ij})}{(n - 1) \times n} \quad (7.13)$$

Note that a sufficient condition for a solution to be a global optimal is that the value of *Total* is equal to zero.

7.4.1 Influence of the penalty parameter β

The following computational experiments aim to evaluate the influence of the distance multiplicity on the behavior of the solution. The main aspect to be evaluated is the effect of the value β for the binary feasibility of the solution; it is expected that the bigger the value β , the closer to $\{0, 1\}$ is the solution. However, large penalty parameters entail numerical instabilities during the solution approach. For this reason, successive approximations methods can be more stable and provide more accurate results ([Mehanna et al. 2014](#)).

The evaluation of the parameter β uses the instance presented on Figure 7.2. This is a nanostructure instance that contains 60 vertices and 21 different distances, with multiplicity (z_a) between 30 and 120. The histogram with the frequency of the distance (y-axis) and the length of the distance (x-axis) is on the left-hand side. The 3D structure of the problem is represented on the left-hand side.

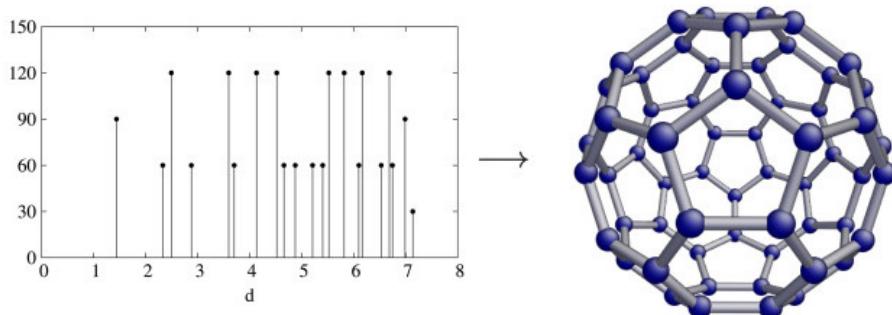


Figure 7.2: Image adapted from [Gujarathi et al. \(2014\)](#).

Table 7.1 describes the value of the parameter β used in each test and the worst feasibility value, calculated as $Feas = \max_y(\min(|y - 1|, |y|))$; the column "Time(s)" contains the computational time in seconds; column "Total" gives the sum of all deviations; column "Average" contains the average deviation for each edge; and column "Std Deviation" contains the standard deviation of the values p_{ij} and n_{ij} .

β	Feas.	Deviation (p_{ij}, n_{ij})			Time (s)
		Total	Average	Std Deviation	
5	10^2	**	**	**	**
10	10^{-7}	8.1×10^{-4}	4.7×10^{-7}	1.04×10^{-3}	327
10^2	10^{-8}	0.001	6.0×10^{-7}	7.1×10^{-10}	648
10^4	10^{-5}	0.10	5.7×10^{-5}	6.6×10^{-15}	619
10^6	10^{-7}	0.23	1.3×10^{-6}	3.4×10^{-5}	511
10^8	10^{-7}	0.19	1.1×10^{-6}	1.1×10^{-5}	608

Table 7.1: Influence of the penalty parameter β in the convergence of the solution approach.

7.4.2 Influence of the multiplicity

These computational studies explored the same instance used in the previous section. Four randomly generated instances were also used, for which there are no repetitions in the distance list; for these instances, the proposed model cannot take advantage of the repetitions to reduce the number of constraints in the set (7.3).

Table 7.2 presents the results for the randomly generated instances — these instances are available online¹. The column "Vertex" gives the number of vertices for each instance; "Dist." gives the number of entries in the distance list; the columns "Total" and "Average" in the field "Distance" give, respectively, the total sum of the distances and the average distance value for each instance. The instances were generated to have a similar average distance between the vertices. The penalty value β was set to 100, based on the results of the previous experiments. This choice aims to provide feasibility values under 10^{-8} ($Feas \leq 10^{-8}$), and without entailing numerical instability.

Number of Vertex	Dist.	Distance		Feas.	Deviation (p_{ij}, n_{ij})			Time (s)
		Total	Average		Total	Average	Std Deviation	
4	7	34.47	5.74	10^{-5}	2.54	0.42	0.64	0.1
6	15	76.19	5.07	10^{-9}	0.07	0.01	0.001	2.3
8	28	179.99	6.42	10^{-8}	1.50	0.05	0.06	5.4
20	190	1181.53	6.21	10^{-9}	1.88	0.01	0.01	870

Table 7.2: Computational results for the randomly generated instances.

¹<https://github.com/petrabartmeyer/SBPO2019>

The desired feasibility condition was achieved for almost all the instances, with a single exception for the instances with four vertices. The four vertices instance also presented the most significant deviations; these difficulties in the optimization were probably due to a poor choice of the initial point — because the solver used is not a global optimization solver, the convergence of the solution depends on the initial point.

Note that the number of distances in the instances is given by the combination of the number of vertices in the instance. Therefore, the number of distances quadratically increases as the number of vertices increases, and the computational effort should increase similarly.

The largest instance solved in this computational experiments has 20 vertices and 190 distances, leading to 190 quadratic nonconvex constraints (because there are no duplicated entries in the distance list), and to a total solution time around 800 seconds. To give a perspective of the benefits provided by the repetitions of distances the nanostructure instance has 60 vertices and 1770 distances, and the number of quadratic nonconvex constraints is only 21, due to the distance multiplicity; this characteristic allows solving the instance in less than 700 seconds, with $\beta = 100$.

7.4.3 Protein conformation

The protein conformation is the problem of defining the 3D structure, or backbone, of a protein. In general, this problem is interpreted as an assigned distance geometry problem, since the connection between two consecutive atoms is known. However, without this chemical information, the general case of the problem protein conformation problems is an unassigned distance geometry problem.

Figure 7.3 illustrates the protein conformation solutions for instances with 100 vertices, on the left-hand side, and with 20 vertices, on the right-hand side. The backbone instances present the same distance between two consecutive vertices; however, the distance list contains a range of values larger than the previously generated instances.

For these computational studies, a third group of instances using the proposal of [Lavor \(2006\)](#) was generated; these instances simulate protein backbones in a 3D space. The instances can present multiplicity, but not as much as in the instance represented in Figure 7.2. Four classes of instances were generated, using different number of vertices; each class contains 10 instances, for 5, 7, 10, and 20 vertices.

The model was coded with the AMPL language ([Fourer. et al. 2003](#)) and solved with Knitro ([Waltz & Plantenga 2011](#)). The maximum execution time was set as 3600 seconds.

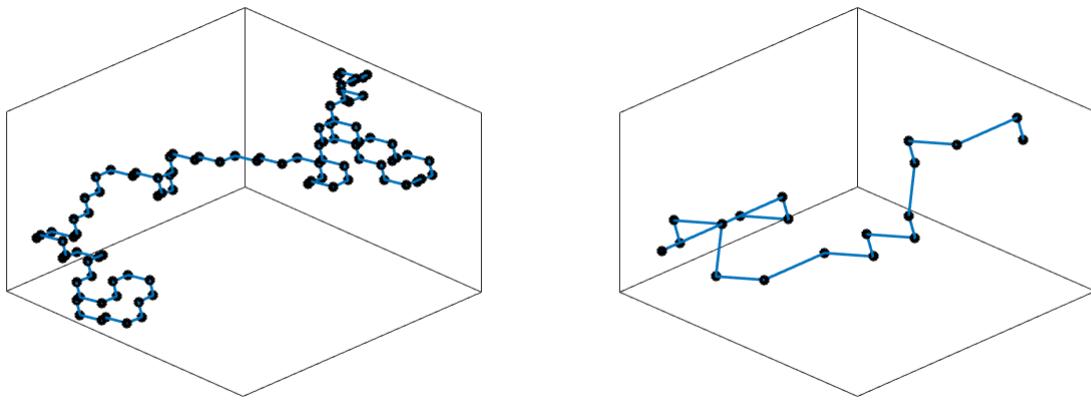


Figure 7.3: Example of two protein conformation instances with 20 vertices (right) and 100 vertices (left), generated using the procedure described in (Lavor 2006).

The instances in Table 7.3 are also available online².

The computational studies compare the mathematical models for the uDGP, using the formulation with binary variables and the formulation with the relaxation of binary variables, given by (7.7)-(7.11). Table 7.3 presents the computational results for the model with binary variables, (7.2)-(7.6), named "Integer", and for the model (7.7)-(7.11), called "Relaxed". The column "Bin Var" presents the number of binary variables for the "Integer" formulation; column " $|Solved|$ " is the cardinality of the set of instances solved within 3600 seconds; the column "Dev Total" presents the total sum of deviation, for all the instances solved in each class. The computation of the "Dev Total" is presented in Equation 7.14.

$$Dev\ Total = \sum_{a \in Solved} \sum_{i=1}^{n-1} \sum_{j=i}^n (p_{ij} + n_{ij}) \quad (7.14)$$

where $Solved$ is the set of instance solved for each class of instance. An instance is considered solved if it achieves an optimal solution within 3600, and with $Feas \leq 10^{-8}$.

Vertex	Number of		Integer		Relaxed	
	Distances	Bin Var	$ Solved $	Dev Total	$ Solved $	Dev Total
5	10	70	0	—	10	7.5
7	21	315	1	1.57	10	38.7
10	45	1440	0	—	8	59.7
20	190	25270	0	—	3	443.3

Table 7.3: Data about instances and solutions. Table from Bartmeyer & Lyra (2020).

The results in Table 7.3 show that the proposed relaxation allow to solve all the instances within the maximum execution time; however, some of the instances fail to achieve $Feas \leq 10^{-8}$.

²<https://github.com/petrabartmeyer/uDGP>

7.5 Discussion

This chapter proposed a new mathematical formulation for the unassigned distance geometry problem (uDGP) and presented computational tests to evaluate the quality of the proposed modeling and the benefits of using the binary relaxation developed in Chapter 3. The model is nonlinear, nonconvex, and includes binary variables. Therefore, it is a severe testbed for the binary relaxation proposed in this research.

The computational tests showed that the binary relaxation allowed to solve more instances and with higher dimensions when compared to the formulation using binary variables. The penalty parameter β did not need to be set to high values in order to provide binary feasibility; indeed, the computational tests showed that $\beta = 10$ was enough to provide good approximations for the binary variables. This is a positive aspect because large values for β could lead to numerical instabilities.

The constraint set (7.3), designed to explore the distance multiplicity, proved to be effective; the high multiplicity instance with 60 vertices was solved in less computational time than the instance with 20 vertices and low multiplicity. This feature can be helpful to applications in nanostructure because these applications present a more symmetric structure, which entails more distance multiplicity.

The explorations of the protein conformation problem showed that the distance multiplicity is low, making the constraint set (7.3) less efficient when compared with the nanostructure applications. However, the use of the proposed relaxation allowed solving instances with up to 20 vertices and more than 25 thousand binary variables.

Part III

Conclusion and Future works

CHAPTER 8

Conclusions

The core of this thesis is a proposal of a new quadratic relaxation for binary variables. A set of theoretical results proved that the original binary problem can be converted into a new quadratic continuous optimization problem without losing the integrality properties of the problem. The theoretical aspects were extended to the quadratic unconstrained binary optimization problem, and the computational benefits of the proposed relaxation were explored in three applications. The first application addressed the reliability of electrical distribution networks. The second application explored strategies to automate the image segmentation process for myelinated axon measurements in nerve root images. The third application addressed the unassigned distance geometry problem.

The theoretical results for the quadratic unconstrained binary optimization problem proved that the proposed relaxation is able to provide a convex objective function, using a suitable value for a Lagrangian multiplier. Computational experiments showed that the proposed relaxation presented the best overall performance when compared with the previous relaxation approaches in the literature.

The first application considered the placement of fault sensors to reduce the effect of temporary faults in power distribution networks. A mathematical model was proposed to find the best locations for fault sensors in distribution networks, with the objective of reducing the detection time of reclosers. The best solutions of this deterministic location problem are inputs to a local search strategy that inserts stochastic aspects into the model, and further improves the sensor locations guided by Monte Carlo simulations. Computational studies showed that the hybrid optimization-simulation approach can reduce the amount of

time that short-circuit currents travel in the network, even when using a small number of sensors. The next step in this study is the analysis of the benefits of fault sensors placement in order to avoid that the temporary faults lead to permanent interruptions — permanent interruptions occur when the recloser operations are not fast enough to save fuses. Another exploration is the impact of distributed generation in the sensor placement problem.

The second application developed strategies for the automation of myelinated axons measurements in nerve root images. A first approach relied on combinatorial optimization concepts that interpret the image as a graph, and the myelinated axons as Hamiltonian cycles in the sub-graphs. The computational studies for the Hamiltonian cycle provided evidence that the proposed relaxation increases the size of the instances solvable via mathematical programming — it also indicates that the proposed relaxation can be useful to other formulations of the traveling salesman problem. The second approach uses a graph search based on the flood-fill algorithm. This algorithm is a simple coloring strategy with linear computational complexity, a property that allows to address the segmentation problem without the resizing of the original images; however, it requires a preprocessing phase that may introduce noises that interfere in the segmentation process. The computational studies showed that the approach was able to automate the segmentation process with good overall accuracy of results. Both alternatives for the automation of the nerve root image segmentation should still be regarded as on-going researches; a more extensive evaluation of the positive aspects and limitations of the methodologies should be performed by neuroscientists.

The third application addressed the unassigned Distance Geometry Problem (uDGP). The uDGP is a class of distance geometry problems with a wide spectrum of applications, including the design of structures, nanotechnology, and bio-engineering. The research developed a mathematical model for the uDGP, which is an incipient area of research. The model relied on quadratic constraints to represent the Euclidean distance between vertices and used binary variables for the assignment of distances to edges of 3D graphs. The proposed relaxation enhanced the optimization process allowing to solve instances of the uDGP with up to twenty thousand binary variables. The next step in this research is the development of solution strategies based on meta-heuristics in order to provide a framework for comparisons with the mathematical programming approach developed here.

To conclude, the binary relaxation ideas developed in this research motivated innovations in the quadratic unconstrained optimization problem and contributions in three new applications in combinatorial optimization. The innovative aspects in the first application are the mathematical model and the solution strategy composed of two phases: a deterministic

solution using the proposed mathematical model, and a local search that improves the deterministic solutions with the evaluation of the stochastic aspects of the problem. The second application presents two new methodologies for the automation of nerve root segmentation measurements. Innovative aspects of both methodologies are a graph interpretation of the image measurement problems that avoids the use of segmentation methods based on pattern recognition; other innovative aspects are the development of a combinatorial optimization approach and a graph search strategy for the automation of the myelin sheath measurements. The third application addresses the unassigned distance geometry problem; the innovations are the proposition of the mathematical model for the problem and the solution strategy based on the proposed relaxation.

The appendix describes the talks and papers that exposed the ideas developed in this thesis.

Bibliography

- Acosta, J. S., López, J. C. & Rider, M. J. (2018), ‘Optimal multi-scenario, multi-objective allocation of fault indicators in electrical distribution systems using a mixed-integer linear programming model’, *IEEE Transactions on Smart Grid* **10**(4), 4508–4519.
- Aiex, R. M., Resende, M. G. & Ribeiro, C. C. (2007), ‘TTT plots: a perl program to create time-to-target plots’, *Optimization Letters* **1**(4), 355–366.
- Alam, A., Alam, M. N., Pant, V. & Das, B. (2018), ‘Placement of protective devices in distribution system considering uncertainties in loads, temporary and permanent failure rates and repair rates’, *IET Generation, Transmission Distribution* **12**(7), 1474–1485.
- Albers, C. J., Critchley, F. & Gower, J. C. (2011), ‘Quadratic minimisation problems in statistics’, *Journal of Multivariate Analysis* **102**(3), 698–713.
- Anstreicher, K. M. (2012), ‘On convex relaxations for quadratically constrained quadratic programming’, *Mathematical programming* **136**(2), 233–251.
- Anstreicher, K. & Wolkowicz, H. (2000), ‘On lagrangian relaxation of quadratic matrix constraints’, *SIAM Journal on Matrix Analysis and Applications* **22**(1), 41–55.
- Anthony, M., Boros, E., Crama, Y. & Gruber, A. (2017), ‘Quadratic reformulations of nonlinear binary optimization problems’, *Mathematical Programming* **162**(1-2), 115–144.
- Applegate, D. L., Bixby, R. E., Chvatal, V. & Cook, W. J. (2006), *The traveling salesman problem: a computational study*, Princeton university press.
- Audet, C., Hansen, P., Jaumard, B. & Savard, G. (2000), ‘A branch and cut algorithm for nonconvex quadratically constrained quadratic programming’, *Mathematical Programming* **87**(1), 131–152.
- Bai, S., Yang, X., Zhu, G., Jiang, D. & Huang, J. (2010), ‘Generalized knight’s tour on 3d chessboards’, *Discrete applied mathematics* **158**(16), 1727–1731.
- Bartmeyer, P. M. & Lyra, C. (2020), ‘A new quadratic relaxation for binary variables applied to the distance geometry problem’, *Structural and Multidisciplinary Optimization* pp. 1–5.
- Bazaraa, M. S. & Shetty, C. M. (2012), *Foundations of optimization*, Vol. 122, Springer Science & Business Media.
- Beasley, J. E. (1998), ‘Heuristic algorithms for the unconstrained binary quadratic programming problem’, *London, England* pp. 1–16.
- Bektaş, T. & Gouveia, L. (2014), ‘Requiem for the Miller–Tucker–Zemlin subtour elimination constraints?’, *European Journal of Operational Research* **236**(3), 820–832.
- Bendsøe, M. P. (1989), ‘Optimal shape design as a material distribution problem’, *Structural optimization* **1**(4), 193–202.

Bertsekas, D. P. (1997), ‘Nonlinear programming’, *Journal of the Operational Research Society* **48**(3), 334–334.

Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. (2017), ‘Julia: A fresh approach to numerical computing’, *SIAM review* **59**(1), 65–98.

Billinge, S. J., Duxbury, P. M., Gonçalves, D. S., Lavor, C. & Mucherino, A. (2016), ‘Assigned and unassigned distance geometry: applications to biological molecules and nanostructures’, *4OR* **14**(4), 337–376.

Billinton, R. & Allan, R. N. (1992), *Reliability Evaluation of Engineering Systems: Concepts and Techniques*, 2 edn, Plenum Press and Pitman Publishing Limited.

Billionnet, A. & Elloumi, S. (2007), ‘Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem’, *Mathematical Programming* **109**(1), 55–68.

Biscola, N. P., Bartmeyer, P. M., Zhang, N. & Havton, L. A. (2020), ‘Strategies for autonomic nervous system mapping of fiber composition and neural circuitry using transmission electron microscopy in support of sparc projects’, *The FASEB Journal* **34**(S1), 1–1.

Biswas, P. & Ye, Y. (2006), A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization, in ‘Multiscale optimization methods and applications’, Springer, pp. 69–84.

Bixby, B. (2014), ‘Inc.,“Gurobi optimizer reference manual,” 2015’.

Boros, E., Hammer, P. L. & Tavares, G. (2007), ‘Local search heuristics for quadratic unconstrained binary optimization (QUBO)’, *Journal of Heuristics* **13**(2), 99–132.

Brown, R. E. (2017), *Electric power distribution reliability*, CRC press.

Burer, S. & Letchford, A. N. (2012), ‘Non-convex mixed-integer nonlinear programming: A survey’, *Surveys in Operations Research and Management Science* **17**(2), 97–106.

Calude, C. S., Dinneen, M. J. & Hua, R. (2017), ‘QUBO formulations for the graph isomorphism problem and related problems’, *Theoretical Computer Science* **701**, 54–69.

Cao, M., Anderson, B. D. & Morse, A. S. (2006), ‘Sensor network localization with imprecise distances’, *Systems & control letters* **55**(11), 887–893.

Chapuis, G., Djidjev, H., Hahn, G. & Rizk, G. (2019), ‘Finding maximum cliques on the d-wave quantum annealer’, *Journal of Signal Processing Systems* **91**(3-4), 363–377.

Chowdhury, A. & Koval, D. (2011), *Power distribution system reliability: practical methods and applications*, Vol. 48, John Wiley & Sons.

Clifford, R., Jalsenius, M., Montanaro, A. & Sach, B. (2012), ‘The complexity of flood filling games’, *Theory of Computing Systems* **50**(1), 72–92.

Conrad, A., Hindrichs, T., Morsy, H. & Wegener, I. (1994), ‘Solution of the knight’s hamiltonian path problem on chessboards’, *Discrete Applied Mathematics* **50**(2), 125–134.

Cplex, I. I. (2009), ‘V12. 1: User’s manual for CPLEX’, *International Business Machines Corporation* **46**(53), 157.

Crama, Y. & Rodríguez-Heck, E. (2017), ‘A class of valid inequalities for multilinear 0-1 optimization problems’, *Discrete Optimization* **25**, 28–47.

Crum, W. R., Camara, O. & Hill, D. L. (2006), ‘Generalized overlap measures for evaluation and validation in medical image analysis’, *IEEE transactions on medical imaging* **25**(11), 1451–1461.

- Delei, J., Sen, B. & Wenming, D. (2009), An ant colony optimization algorithm for knight's tour problem on the chessboard with holes, in '2009 First International Workshop on Education Technology and Computer Science', Vol. 1, IEEE, pp. 292–296.
- Dolan, E. D. & Moré, J. J. (2002), 'Benchmarking optimization software with performance profiles', *Mathematical programming* **91**(2), 201–213.
- dos Santos, C., Cavalheiro, E., Bartmeyer, P. & Lyra, C. (2020), A minlp model to optimize battery placement and operation in smart grids, in '2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)', IEEE, pp. 1–5.
- Dugan, R. C. (2012), 'Reference guide: The open distribution system simulator (OpenDSS)', *Electric Power Research Institute, Inc* **7**, 29.
- Dunning, I., Huchette, J. & Lubin, M. (2017), 'Jump: A modeling language for mathematical optimization', *SIAM Review* **59**(2), 295–320.
- Duxbury, P. M., Granlund, L., Gujarathi, S., Juhas, P. & Billinge, S. J. (2016), 'The unassigned distance geometry problem', *Discrete Applied Mathematics* **204**, 117–132.
- Farajollahi, M., Fotuhi-Firuzabad, M. & Safdarian, A. (2016), 'Deployment of fault indicator in distribution networks: A mip-based approach', *IEEE Transactions on Smart Grid* **9**(3), 2259–2267.
- Fourer., R., Gay, D. & Kernighan, B. W. (2003), *AMPL: A modeling language for mathematical programming*, 2 edn, Pacific Grove: Thomson/Brooks/Cole.
- Garcia, M. L., Lobsiger, C. S., Shah, S. B., Deerinck, T. J., Crum, J., Young, D., Ward, C. M., Crawford, T. O., Gotow, T., Uchiyama, Y. et al. (2003), 'NF-M is an essential target for the myelin-directed "outside-in" signaling cascade that mediates radial axonal growth', *The Journal of cell biology* **163**(5), 1011–1020.
- Garey, M. R. & Johnson, D. S. (1979), *Computers and intractability: A guide to the theory of NP-completeness*, WH Freeman San Francisco.
- Gerber, D., Ghidinelli, M., Tinelli, E., Somandin, C., Gerber, J., Pereira, J. A., Ommer, A., Figlia, G., Miehe, M., Nägeli, L. G. et al. (2019), 'Schwann cells, but not oligodendrocytes, depend strictly on dynamin 2 function', *eLife* **8**, e42404.
- Glover, F., Kochenberger, G. A. & Alidaee, B. (1998), 'Adaptive memory tabu search for binary quadratic programs', *Management Science* **44**(3), 336–345.
- Glover, F., Lewis, M. & Kochenberger, G. (2018), 'Logical and inequality implications for reducing the size and difficulty of quadratic unconstrained binary optimization problems', *European Journal of Operational Research* **265**(3), 829–842.
- Gondzio, J. (2012), 'Interior point methods 25 years later', *European Journal of Operational Research* **218**(3), 587–601.
- Gordon, V. S. & Slocum, T. J. (2004), The knight's tour-evolutionary vs. depth-first search, in 'Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)', Vol. 2, IEEE, pp. 1435–1440.
- Gueye, S. & Michelon, P. (2009), 'A linearization framework for unconstrained quadratic (0-1) problems', *Discrete Applied Mathematics* **157**(6), 1255–1266.
- Gujarathi, S., Farrow, C., Glosser, C., Granlund, L. & Duxbury, P. (2014), 'Ab-initio reconstruction of complex euclidean networks in two dimensions', *Physical Review E* **89**(5), 053311.
- Hammer, P. L. & Shlifer, E. (1971), 'Applications of pseudo-boolean methods to economic problems', *Theory and decision* **1**(3), 296–308.

- Hansen, P. & Meyer, C. (2009), ‘Improved compact linearizations for the unconstrained quadratic 0-1 minimization problem’, *Discrete Applied Mathematics* **157**(6), 1267–1290.
- Hansen, P. & Mladenović, N. (2001), ‘Variable neighborhood search: Principles and applications’, *European journal of operational research* **130**(3), 449–467.
- Helberg, C. & Rendl, F. (1998), ‘Solving quadratic (0, 1)- problems by semidefinite programs and cutting planes’, *Mathematical programming* **82**(3), 291–315.
- Ho, C., Lee, T. & Lin, C. (2010), ‘Optimal placement of fault indicators using the immune algorithm’, *IEEE Transactions on Power Systems* **26**(1), 38–45.
- Horst, R. & Thoai, N. V. (1999), ‘DC programming: overview’, *Journal of Optimization Theory and Applications* **103**(1), 1–43.
- Huang, S. & Dokmanić, I. (2020), ‘Reconstructing point sets from distance distributions’, *arXiv preprint arXiv:1804.02465*.
- Juhás, P., Cherba, D., Duxbury, P., Punch, W. & Billinge, S. (2006), ‘Ab initio determination of solid-state nanostructure’, *Nature* **440**(7084), 655.
- Kochenberger, G. A., Hao, J., Lü, Z., Wang, H. & Glover, F. (2013), ‘Solving large scale max cut problems via tabu search’, *Journal of Heuristics* **19**(4), 565–571.
- Kochenberger, G., Hao, J., Glover, F., Lewis, M., Lü, Z., Wang, H. & Wang, Y. (2014), ‘The unconstrained binary quadratic programming problem: a survey’, *Journal of Combinatorial Optimization* **28**(1), 58–81.
- Köppe, M. (2012), On the complexity of nonlinear mixed-integer optimization, in ‘Mixed Integer Nonlinear Programming’, Springer, pp. 533–557.
- Lasdon, L. S. (2011), *Optimization theory for large systems*, Dover.
- Lavor, C. (2006), On generating instances for the molecular distance geometry problem, in ‘Global optimization’, Springer, pp. 405–414.
- Lee, J., Jun, S., Cho, Y., Lee, H., Kim, G. B., Seo, J. B. & Kim, N. (2017), ‘Deep learning in medical imaging: general overview’, *Korean journal of radiology* **18**(4), 570–584.
- Leo, M., Medioni, G., Trivedi, M., Kanade, T. & Farinella, G. M. (2017), ‘Computer vision for assistive technologies’, *Computer Vision and Image Understanding* **154**, 1–15.
- Lewis, M. & Glover, F. (2017), ‘Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis’, *Networks* **70**(2), 79–97.
- Liberti, L. (2007), ‘Compact linearization for binary quadratic problems’, *4OR* **5**(3), 231–245.
- Liberti, L. & Lavor, C. (2018), Open research areas in distance geometry, in ‘Open Problems in Optimization and Data Analysis’, Springer, pp. 183–223.
- Liberti, L., Lavor, C., Mucherino, A. & Maculan, N. (2011), ‘Molecular distance geometry methods: from continuous to discrete’, *International Transactions in Operational Research* **18**(1), 33–51.
- Liberti, L. & Maculan, N. (2006), *Global optimization: from theory to implementation*, Vol. 84, Springer Science & Business Media.
- Linderoth, J. (2005), ‘A simplicial branch-and-bound algorithm for solving quadratically constrained quadratic programs’, *Mathematical programming* **103**(2), 251–282.
- Luenberger, D. G. & Ye, Y. (2003), *Linear and Nonlinear Programming*, Springer US.

- Mahapatro, A. & Khilar, P. M. (2013), ‘Fault diagnosis in wireless sensor networks: A survey’, *IEEE Communications Surveys & Tutorials* **15**(4), 2000–2026.
- Mallach, S. (2018), ‘Compact linearization for binary quadratic problems subject to assignment constraints’, *4OR* **16**(3), 295–309.
- Martinez, J. (2005), ‘A note on the theoretical convergence properties of the simp method’, *Structural and Multidisciplinary Optimization* **29**(4), 319–323.
- Mauri, G. R. & Lorena, L. A. N. (2012), ‘Improving a Lagrangian decomposition for the unconstrained binary quadratic programming problem’, *Computers & Operations Research* **39**(7), 1577–1581.
- McCormick, G. P. (1976), ‘Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems’, *Mathematical programming* **10**(1), 147–175.
- Mehanna, O., Huang, K., Gopalakrishnan, B., Konar, A. & Sidiropoulos, N. D. (2014), ‘Feasible point pursuit and successive approximation of non-convex QCQPs’, *IEEE Signal Processing Letters* **22**(7), 804–808.
- Merikoski, J. K. & Kumar, R. (2004), ‘Inequalities for spreads of matrix sums and products’, *Applied Mathematics E-Notes* **4**, 150–159.
- Merz, P. & Freisleben, B. (1999), Genetic algorithms for binary quadratic programming, in ‘Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation—Volume 1’, Morgan Kaufmann Publishers Inc., pp. 417–424.
- Merz, P. & Katayama, K. (2004), ‘Memetic algorithms for the unconstrained binary quadratic programming problem’, *BioSystems* **78**(1-3), 99–118.
- Miettinen, K. (2012), *Nonlinear multiobjective optimization*, Vol. 12, Springer Science & Business Media.
- Miller, C. E., Tucker, A. W. & Zemlin, R. A. (1960), ‘Integer programming formulation of traveling salesman problems’, *Journal of the ACM (JACM)* **7**(4), 326–329.
- Minnema, J., van Eijnatten, M., Kouw, W., Diblen, F., Mendrik, A. & Wolff, J. (2018), ‘Ct image segmentation of bone for medical additive manufacturing using a convolutional neural network’, *Computers in biology and medicine* **103**, 130–139.
- Moré, J. J. & Wu, Z. (1999), ‘Distance geometry optimization for protein structures’, *Journal of Global Optimization* **15**(3), 219–234.
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J. & Sewell, E. C. (2016), ‘Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning’, *Discrete Optimization* **19**, 79–102.
- Nesterov, Y., Wolkowicz, H. & Ye, Y. (2000), Semidefinite programming relaxations of nonconvex quadratic optimization, in ‘Handbook of semidefinite programming’, Springer, pp. 361–419.
- Palubeckis, G. (2004), ‘Multistart tabu search strategies for the unconstrained binary quadratic optimization problem’, *Annals of Operations Research* **131**(1), 259–282.
- Pardalos, P. M., Prokopyev, O. A., Shylo, O. V. & Shylo, V. P. (2008), ‘Global equilibrium search applied to the unconstrained binary quadratic optimization problem’, *Optimisation Methods and Software* **23**(1), 129–140.
- Pavlidis, T. (1979), ‘Filling algorithms for raster graphics’, *Computer graphics and image processing* **10**(2), 126–141.

- Perdomo-Ortiz, A., Flueguemann, J., Narasimhan, S., Smelyanskiy, V. N. & Biswas, R. (2014), A quantum approach to diagnosis of multiple faults in electrical power systems, in 'Space Mission Challenges for Information Technology (SMC-IT), 2014 IEEE International Conference on', IEEE, pp. 46–53.
- Pohl, I. (1967), 'A method for finding hamilton paths and knight's tours', *Communications of the ACM* **10**(7), 446–449.
- Pörn, R., Nissfolk, O., Skjäl, A. & Westerlund, T. (2017), 'Solving 0-1 quadratic programs by reformulation techniques', *Industrial & Engineering Chemistry Research* **56**(45), 13444–13453.
- Porta, J. M., Rojas, N. & Thomas, F. (2018), Distance geometry in active structures, in 'Mechatronics for Cultural Heritage and Civil Engineering', Springer, pp. 115–136.
- Rodriguez Heck, E. & Crama, Y. (2018), 'Linear and quadratic reformulation techniques for nonlinear 0-1 optimization problems', *Welcome to ORBEL 32!* p. 76.
- Rojas, N. & Thomas, F. (2013), 'Application of distance geometry to tracing coupler curves of pin-jointed linkages', *Journal of mechanisms and robotics* **5**(2).
- Saha, M. M., Izykowski, J. J. & Rosolowski, E. (2009), *Fault location on power networks*, Springer Science & Business Media.
- Santos, C., Bartmeyer, P. M. & Lyra, C. (2019), Allocation of fault indicators for adaptive protection schemes, in '2019 IEEE PES Innovative Smart Grid Technologies Conference-Latin America (ISGT Latin America)', IEEE, pp. 1–6.
- Schneider, C. A., Rasband, W. S. & Eliceiri, K. W. (2012), 'NIH image to ImageJ: 25 years of image analysis', *Nature methods* **9**(7), 671.
- Schwenk, A. J. (1991), 'Which rectangular chessboards have a knight's tour?', *Mathematics Magazine* **64**(5), 325–332.
- Sherali, H. D. & Smith, J. C. (2007), 'An improved linearization strategy for zero-one quadratic programming problems', *Optimization Letters* **1**(1), 33–47.
- Sing, F. (1976), 'Some results on matrices with prescribed diagonal elements and singular values', *Canadian Mathematical Bulletin* **19**(1), 89–92.
- Stolpe, M. & Sandal, K. (2018), 'Structural optimization with several discrete design variables per part by outer approximation', *Structural and Multidisciplinary Optimization* **57**(5), 2061–2073.
- Takefuji, Y. (1992), Knight's tour problems, in 'Neural Network Parallel Computing', Springer, pp. 111–118.
- Thompson, R. C. (1977), 'Singular values, diagonal elements, and convexity', *SIAM Journal on Applied Mathematics* **32**(1), 39–63.
- Wächter, A. & Biegler, L. (2009), 'Ipopt-an interior point optimizer'.
- Waltz, R. & Plantenga, T. (2011), 'Knitro documentation release 8.0', *Ziena Optimization LLC* **7**, 33–34.
- Wang, Y., Hao, J., Glover, F. & Lü, Z. (2014), 'A tabu search based memetic algorithm for the maximum diversity problem', *Engineering Applications of Artificial Intelligence* **27**, 103–114.
- Wang, Y., Lü, Z., Glover, F. & Hao, J. (2012), 'Path relinking for unconstrained binary quadratic programming', *European Journal of Operational Research* **223**(3), 595–604.

- Wolsey, L. A. & Nemhauser, G. L. (1999), *Integer and combinatorial optimization*, Vol. 55, John Wiley & Sons.
- Zaimi, A., Duval, T., Gasecka, A., Côté, D., Stikov, N. & Cohen-Adad, J. (2016), ‘Axonseg: open source software for axon and myelin segmentation and morphometric analysis’, *Frontiers in neuroinformatics* **10**, 37.
- Zaimi, A., Wabbartha, M., Herman, V., Antonsanti, P., Perone, C. S. & Cohen-Adad, J. (2018), ‘Axondeepseg: automatic axon and myelin segmentation from microscopy data using convolutional neural networks’, *Scientific reports* **8**(1), 3816.
- Zheng, X., Sun, X. & Li, D. (2011), ‘Nonconvex quadratically constrained quadratic programming: best dc decompositions and their SDP representations’, *Journal of Global Optimization* **50**(4), 695–712.

APPENDIX A

Talks and papers

A.1 Conferences participation

Part of this research has been presented in conferences, workshops and journal papers, such as the Brazilian Symposium on Operation Research (SBPO) in 2018 and 2019; the Brazilian Workshop on Cutting Stock and related problems (ONPCE) in 2017 and 2019; and the Regional Workshop in Operation Research (ERPO) in 2018.

It was also presented in two international events during the year of 2018, the International Symposium on Mathematical Programming (ISMP) in Bordeaux - France and the International Workshop in Lot Sizing (IWLS) in Ubatuba - Brazil.

A.2 Abstracts

- Santos, Caio; Cavalheiro, Ellen; Bartmeyer, Petra; Lima, Rodrigo; Lyra, Christiano. "A Lot Sizing Perspective for the Battery Storage Coordination in Power Distribution Systems". International Workshop in Lot Sizing (IWLS) 2018.
- Bartmeyer, Petra; Lyra, Christiano. "A new approach to relax the binary variables on quadratic unconstrained binary problems". International Symposium on Mathematical Programming (ISMP) 2018.
- Biscola, Natalia; Bartmeyer, Petra; Zhang, Nianhui; Havton, Leif. "Strategies for Autonomic Nervous System Mapping of Fiber Composition and Neural Circuitry Using Transmission Electron Microscopy in Support of SPARC Projects". Experimental

Biology Meeting, 2020. San Diego, USA ([Biscola et al. 2020](#)).

A.3 Conference papers

- Bartmeyer, Petra; Lyra, Christiano. "A quadratic reformulation of binary variables applied to unconstrained binary quadratic problem". Brazilian Symposium on Operation Research (SBPO), 2018. Rio de Janeiro, Brazil.
- Bartmeyer, Petra; Lyra, Christiano. "A new mixed-integer nonlinear optimization model for the unassigned distance geometry problem considering inaccuracy in the distance list". Presented in the Brazilian Symposium on Operation Research (SBPO), 2019. Limeira, Brazil.
- Santos, Caio; Bartmeyer, Petra; Lyra, Christiano. "Allocation of Fault Indicators for Adaptive Protection Schemes". IEEE Innovative Smart Grid Technologies (ISGT-LA), 2019. Gramado, Brazil ([Santos et al. 2019](#)).
- Santos, Caio; Bartmeyer, Petra; Cavelheiro, Ellen; Lyra, Christiano. "An MINLP Model to Optimize Battery Placement and Operation in Smart Grids". Innovative Smart Grid Technologies (ISGT-NA), 2020. Washington DC, USA ([dos Santos et al. 2020](#)).

A.4 Journal papers - published

- Bartmeyer, P.M., Lyra, C. "A new quadratic relaxation for binary variables applied to the distance geometry problem" Structural and Multidisciplinary Optimization (2020). <https://doi.org/10.1007/s00158-020-02567-7>

The article summarizes the theoretical results concerning the proposed relaxation and the mathematical model developed for the protein conformation problem.

- Bartmeyer, P; Bocanegra, S; Oliveira, A "Switching preconditioners using a hybrid approach for linear systems arising from interior point methods for linear programming" Numerical Algorithm (2020). <https://doi.org/10.1007/s11075-020-00893-x>

The results of this paper were mainly developed during my master thesis in collaboration with my supervisor Silvana Bocanegra and the co-supervisor Aurelio R. L. Oliveira.

- Amorim, F; Lopes, D; Bartmeyer, P; Ospina, M. "Productivity and profitability of the sugarcane production in the State of São Paulo, Brazil". Sugar Tech (2020). <https://doi.org/10.1007/s12355-020-00813-2>

This paper proposed a mixed-integer optimization model to analyze aspects related to sugarcane production and costs. The research was developed in partnership with Prof Marco Ospina and Fernando Amorim from the School of Agricultural Engineering, University of Campinas (UNICAMP), and Prof David Lopes from the Department of Rural Economy, São Paulo State University (UNESP).

A.5 Journal papers submitted and in preparation

- Jones, D; Treloar, R; Ouelhadj, D; Glampedakis A, Bartmeyer, P. "Incorporation of Poverty Principles into Goal Programming". Submitted to the European Journal of Operations Research.

This research were developed during a short-term visit to the Centre of Operation Research at the University of Portsmouth - UK. The article explores the aspects of goal programming to find the best budget distribution considering aspects related to fairness. The theoretical developments are applied to a dataset of public schools in the UK.

- Bartmeyer, P; Biscola, N; Havton, L. "A new mathematical approach for axon populations detection in L6-S4 spinal cord levels of Rhesus macaques". In preparation.

This article is result of a sandwich-period at the Brain Research Institute (BRI) of the University of California - Los Angeles (UCLA). Under the supervision of Prof Leif Havton, it was proposed a new mathematical tool to analyze general nerve root images. The theoretical results in Chapter 6 gave support for the ideas in the paper.

A.6 Talks

- Regional Workshop in Operation Research (ERPO) 2018 – "A mathematical programming approach to model the tradeoff between cost and productivity in sugarcane plantation".
- International Symposium on Mathematical Programming (ISMP) 2018 – "A new approach to relax the binary variables on quadratic unconstrained binary problems".

- Brazilian Symposium on Operation Research (SBPO) 2018 – "A quadratic reformulation of binary variables applied to unconstrained binary quadratic problem".
- Brazilian Symposium on Operation Research (SBPO) 2019 – "A new mixed-integer nonlinear optimization model for the unassigned distance geometry problem considering inaccuracy in the distance list".