Nonconvex Quadratic Problems and Games with Separable Constraints

Javier Zazo Ruiz, MSc.

Supervisor: Santiago Zazo Bello, PhD.

Dept. Señales, Sistemas y Radiocomunicaciones
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid

This dissertation is submitted for the degree of
Doctor of Philosophy

July 2017

Abstract

Quadratic programs are generally hard and difficult to solve, where many instances are known to be NP-hard. However, there are some few cases in which the quadratic problem has some hidden convexity and there are methods to solve the problems optimally in polynomial time. When this occurs, we say that strong duality holds, and that the primal problem has zero gap with its dual counterpart. We study this property in nonconvex quadratic problems with separable constraints.

We exploit the disjoint structure of the constraints to analyze when strong duality may occur. We consider problems that may have both inequality or equality quadratic constraints, and we provide sufficient conditions that guarantee zero duality gap. Our results consider any number of constraints as long as they do not overlap between each other.

In addition to these results, we propose two distributed algorithms that aim to solve the quadratic problem optimally whenever the strong duality holds. The first proposal uses a dual formulation and convergence relies on gradient or subgradient methods, depending on the nature of the problem. The second proposal uses a modern decomposition technique from the literature called NOVA and uses successive convex approximations to converge to the optimal solution, which is generally considered faster than first order methods.

We then study two localization problems with ranged measurements. We reformulate them as nonconvex quadratic problems with quadratic constraints and explore their solutions. We refer to the first one as target localization and propose several distributed methods based on primal descent as well as dual ascent methods.

On the one hand, descent methods assume the nonconvexity of the localization problem and may converge to local solutions if the number of measuring nodes is small. They are however fast and require little communication. On the other hand, dual ascent methods aim to solve the localization problem optimally assuming that there is zero duality gap between the primal and dual problem.

The second localization problem we consider is the sensor network localization problem in which a set of nodes try to locate themselves having distance measurements between each other. This is a harder problem, and as before, we propose both distributed methods based on descent techniques as well as based in a dual decomposition.

Finally, we also study the connection of nonconvex quadratic optimization with Nash equilibria in game theory. In order to do so, we propose a robust demand-side management problem in a smart grid, where the agents try to minimize their monetary expenditure. Because there is uncertainty in the real-time energy demand, the cost of energy is prone to variations and a robust model that attends to a worst-case scenario is proposed. This formulation is presented as a min-max game where the maximization game is nonconvex and where the agents have coupled constraints. We study this game and propose a distributed scheme with convergence guarantees to the NE of the game. We also study the game model under a real-time energy market where prices have been set on a day-ahead robust optimization.

Resumen

Los problemas cuadráticos resultan en general difíciles de resolver de manera óptima, y se sabe que muchos de ellos pertenecen incluso a la categoría NP-hard. No obstante, hay algunos casos en los que el problema cuadrático posee una cierta convexidad escondida y existen algoritmos que los resuelven de manera óptima en tiempo polinomial. Cuando esto ocurre, decimos que existe una dualidad fuerte, y que no hay espacio entre las soluciones óptimas del problema primal y del dual. En esta tesis estudiamos cuándo aparece esta propiedad en los problemas cuadráticos no convexos con restricciones separables.

Utilizamos la estructura disjunta para reformular los problemas y analizar cuándo presenta dicha propiedad. Consideramos problemas que pueden tener tanto restricciones de desigualdad como de igualdad cuadráticas, y demostramos condiciones suficientes sobre cuándo existe dualidad cero. Nuestros resultados son válidos para cualquier número finito de restricciones siempre que entre estas restricciones no haya ningún solapamiento.

Adicionalmente, proponemos dos métodos distribuidos que resuelven el problema cuadrático de manera óptima siempre que exista la mencionada dualidad fuerte. El primer método utiliza una formulación dual y la convergencia está condicionada por los métodos de gradiente o subgradiente, que dependen de la naturaleza del problema. El segundo método utiliza una técnica de separación proveniente de la literatura para su resolución distribuida denominada NOVA, y utiliza técnicas de aproximaciones convexas sucesivas. Estos métodos suelen ser más rápidos que los que se basan en direcciones de descenso de primer orden.

También estudiamos dos problemas de localización con medidas de distancia. Reformulamos los problemas a tipo cuadrático con restricciones cuadráticas y estudiamos sus soluciones. Nos referimos al primer tipo de problema como localización de objetivo (TL) y proponemos varios algoritmos distribuidos basados en técnicas de descenso del problema primal, como técnicas de ascenso en el problema dual. Los métodos de descenso primales asumen la no convexidad del problema y podrían converger a soluciones subóptimas cuando el número de nodos es pequeño. Por otra parte, los métodos de ascenso del problema dual intentan resolver el problema de manera óptima suponiendo que hay dualidad fuerte. En principio, dicha propiedad no puede garantizarse de antemano, aunque analizamos y proponemos condiciones necesarias y suficientes sobre cuándo ocurre.

El segundo problema de localización que estudiamos es el problema de localización de sensores en una red, donde un conjunto de los nodos intenta posicionarse utilizando medidas con sus vecinos. Este problema es más difícil que el anterior, aunque también proponemos métodos distribuidos que utilizan técnicas de descenso en el problema primal y de ascenso en el dual.

Finalmente, también estudiamos la relación que existe entre optimización cuadrática no convexa y los equilibrios de Nash en teoría de juegos. Proponemos un modelo de gestión de la demanda energética en redes eléctricas inteligentes, donde los usuarios minimizan sus costes monetarios. Porque hay una inevitable incertidumbre en la demanda energética en tiempo real, el coste de la energía está sujeto a variaciones. Por tanto, proponemos un modelo robusto que estudia un caso peor de dichas fluctuaciones. El juego que proponemos tiene una estructura de min-max, donde el juego que maximiza es no convexo y tiene restricciones acopladas. Analizamos el juego y proponemos un esquema distribuido con garantías de convergencia al equilibrio de Nash. Estudiamos el problema bajo un marco de mercado de energía en tiempo real donde los precios se han fijado un día antes mediante un modelo robusto.

*A mis padres y hermana, que siempre me apoyaron,
y a Melanie, que me acompañó todo el camino.*

Acknowledgements

It has been a long journey, and I would like to thank many people with whom I have enjoyed working all this time and also many others that have accompanied me during this period. First, I would like to thank my supervisor and uncle Santiago Zazo, who has guided me both personally and academically all these years. He has encouraged me and has always provided me with valuable advice, always having my best interest in mind, and for this I am extremely grateful.

I would also like to thank my colleague Sergio Valcarcel, with whom I have learned to be extremely careful in my work. We have shared many lab hours working hard, as well as dead hours talking about life, ourselves and the future, past and present. He also influenced me when I became a vegetarian three years ago, and for all of this I am thankful. I certainly leave a close friend behind.

I have also to thank Juan Parras and Jorge del Val for joining our group in my last two years, for I have learned a lot from them, their presentations and their enthusiasm. Also, our lab mates Adrián Población, Jose Luis Blanco, Virginia Yagüe and Nelson Dopico, for their support, help and advice.

Regarding my visiting research time in Buffalo, I would like to thank Professor Gesualdo Scutari, for his invitation, and further insight and teachings during my whole stay. I send my warmest regards to Peiran Song, Paolo Scarponi, Loris Cannelli, Amir Daneshmand and Shubham Chamadia for their friendship and help during my time in Buffalo.

Special thanks go to Melanie F. Pradier, who has been with me all this time, and to whom this thesis is dedicated. The dedication may be brief, but my gratitude is immense. Also, big thanks to her lab colleagues at Universidad Carlos III, which have kindly accepted me both at leisure and working times: Alfredo Nazabal, Pablo M. Olmos and Ana G. Casado, Víctor Elvira, Francisco Rodríguez and Ana F. Silva, Isabel Valera, Jesús F. Bes, David Ramírez, Tobias Koch and Alicia, Gonzalo Vázquez and Alba, Gonzalo Ríos, Yangfan Liu, Grace Villacrés, Paloma Vaquero and Alberto Valera, Deniz Akyildiz, Alejandro Lancho and many others I have met in all these years. I also have kind regards to Matilde Sánchez, Antonio Artés and Fernando Pérez Cruz for every time in which we have met at congresses and at the university.

From the personal side, I have to thank my parents and sister, for all of the encouragement and understanding that they have given me all these years. For their support and kindness towards everything they do with me since as far as I can remember.

I also have to give many thanks to my closest friends without whom all these years certainly would not have been the same. To Héctor Berlanga and Julie Montes, Álvaro Campesino, Adrián Gómez and Alexis Lamberti. To all the friends from our Friday gatherings, which they know who they are, but are too many to name. To Guillermo Prieto, Berta Herrero, Andrés Pajarón and Elena Álvarez for the nice times, trips and close friendship.

All in all, thanks to all friends, family and colleagues that have been with me at some point in my life.

Table of contents

Nomenclature	xiii
1 Introduction	1
1.1 Nonconvex quadratic problems	2
1.1.1 Strong duality results of quadratic programs	6
1.2 Distributed optimization techniques	10
1.2.1 ADMM decomposition	10
1.2.2 FLEXA decomposition	11
1.2.3 NOVA decomposition	12
1.3 Introduction to algorithmic game theory	14
1.4 Contributions	16
1.4.1 Other contributions	17
1.5 Outline	19
2 Strong duality in nonconvex quadratic problems with separable quadratic constraints	21
2.1 Introduction	21
2.2 Problem formulation	23
2.2.1 Dual formulation	25
2.3 Analysis of the $\mathcal{S}$ -property in QPs	27
2.3.1 F_i simultaneously diagonalizable	30
2.3.2 F_i not simultaneously diagonalizable	31
2.4 Existence of solutions	32
2.5 Algorithms	34
2.5.1 Based on semidefinite programming	34
2.5.2 Projected dual ascent algorithm	35
2.5.3 Distributed formulations	36
2.5.4 Verification of sufficient conditions to satisfy the $\mathcal{S}$ -property	41
2.6 Simulations	42
2.6.1 Robust least squares	44
Appendix 2.A Proofs	48

3	Squared ranged localization problems	57
3.1	Introduction	57
3.2	Target localization	57
3.2.1	ADMM decomposition	62
3.2.2	NEXT decomposition	65
3.2.3	TL decomposition based on dual methods	69
3.3	Sensor Network Localization	75
3.3.1	FLEXA decomposition	77
3.3.2	Centralized proposal based on semidefinite relaxation	80
3.3.3	Distributed proposal based on semidefinite relaxation	82
3.4	Simulations	90
3.4.1	Simulations on the TL problem	90
3.4.2	Simulations of the SNL problem	100
	Appendix 3.A Proofs of TL problem	108
4	Robust worst-case analysis of demand-side managment in smart grids	113
4.1	Introduction	113
4.1.1	Related Work	114
4.2	Smart Grid Model	116
4.2.1	Demand-Side Model	116
4.2.2	Day-ahead Cost Model	118
4.2.3	Real-time Pricing Model	119
4.2.4	Day-ahead Optimization Process	120
4.3	Worst-Case Analysis of the Energy Load Profiles	121
4.3.1	Worst-Case Min-max Game Formulation	123
4.3.2	Distributed Robust Algorithm	128
4.4	Simulation Results	128
4.4.1	Real-time Cost Comparison	130
4.4.2	Comparison of Algorithm 18 with naive users	130
	Appendix 4.A	132
	Appendix 4.B	134
	Appendix 4.C	135
5	Conclusions	139
	References	143

Nomenclature

Mathematical Notation

$\mathbb{R}^p$	Euclidean space of dimension p .
$\mathbb{R}_+^p$	Nonnegative orthant of the Euclidean space of dimension p .
$\mathcal{S}^p$	Space of symmetric matrices of dimension p .
$\mathbf{I}_p$	Identity matrix of dimension p .
$\text{tr}[A]$	Trace of square matrix A .
A^{-1}	Inverse of regular matrix A .
$A^\dagger$	Moore–Penrose pseudoinverse of matrix A .
$(A)_{ij}$	Element in the i th row and j th column of matrix A .
$(A)_{:i}$	i th column of matrix A .
$A \succ 0, A \succeq 0$	Matrix A is positive definite, or positive semidefinite, respectively.
$A \prec 0, A \preceq 0$	Matrix A is negative definite, or negative semidefinite, respectively.
$\triangleq$	Either inequality or equality: $\triangleq \in \{\leq, =\}$.
$\nabla_x f(x)$	Gradient of function $f(x)$ on variable x .
$\nabla_x^2 f(x)$	Hessian matrix of function $f(x)$ on variable x .
$\text{range}[A]$	Range of matrix A .
$\text{null}[A]$	Nullspace of matrix A .
$\bar{\mathbf{E}}$	Square matrix of size $(p+1)$ with single nonzero element $(\bar{\mathbf{E}})_{p+1,p+1} = 1$.

Acronyms / Abbreviations

ADMM	Alternating direction method of multipliers
FLEXA	Flexible parallel algorithm
GNEP	Generalized Nash equilibrium problem
KKT	Karush–Kuhn–Tucker
LS	Least squares
NE	Nash equilibrium
NEP	Nash equilibrium problem
NEXT	In-network nonconvex optimization
NOVA	Inner convex approximation
PCA	Principal component analysis
QCQP	Quadratically constrained quadratic problem
QP	Quadratic problem
RLS	Robust least squares
RMSE	Root mean square error
SDP	Semidefinite programming
SNL	Sensor network localization
SRLS-SNL	Squared ranged least squares sensor network localization
SRLS-TL	Squared ranged least squares target localization
SVD	Singular value decomposition
TL	Target localization
TLS	Total least squares
VI	Variational inequalities

Chapter 1

Introduction

Optimization problems arise in every machine learning approach such as least squares, LASSO, logistic regression, dictionary learning or support vector machines among others. These methods are applied in a wide range of applications, including recommendation systems, compressed sensing, classification problems, and parameter estimation. Many of them present challenges in the amount of data to process (big data), in the location of the data (distribution), in the complexity of the problem (nonconvex), or in the model creation (learning), where all of these aspects are active research areas.

All of these problems are related and affect each other. Learning is difficult in nonconvex problems and high dimensions; distributed and parallel computation is necessary for big data applications; simplicity is also a requirement for huge amounts of data; and, finally, suboptimal solutions may not give any performance guarantees. These considerations emphasize the need to consider all these aspects jointly.

State of the art results using nonconvex methods are noteworthy and very frequently perform better than optimal solutions derived from convex relaxations of the original problems. This evidence motivates that studying algorithms for nonconvex problems is a promising research line, where overcoming the previous difficulties can have a tremendous impact on society. Compressed sensing, sensor decomposition, trust region methods, and methods that use successive convex approximations, are some research areas where quadratic nonconvex formulations are used, and have high impact in many applications which are ubiquitous in our current society. Examples of these are: 3D MRI for diagnosis, seismic imaging for earthquake detection, efficient and inexpensive sensing in biological applications such as DNA micro-array testing, improvement of the resolution in RADAR systems, resource allocation in communication networks, noninvasive cardiac anomaly detection, etc. These involve medical and biological procedures, as well as technological problems, all of which

have a common mathematical representation, and where tailored methods using nonconvex analysis can improve the state of the art of many of our current solutions.

All in all, nonconvex algorithms have high impact in current technology, and quadratic problems (QPs) play an important role in these areas of research. From modeling the mechanisms of a specific problem, to using a quadratic approximation as a method to find solutions to more complex formulations, it is clear that QPs play an essential part in optimization.

In this thesis we study nonconvex quadratic problems with separable constraints, which is a subproblem of a more general formulation. Many problems fall under this category, such as boolean regression, robust least squares, trust-region methods, min-max quadratic problems, etc. These problems are very general and appear in many applications, some of which we analyze in this thesis.

We also consider the use of QPs in a game theory setting regarding a smart grid application. Noncooperative game theory models the interactions of (possibly) many players with local objectives and analyzes the equilibrium points arising from the player's actions and strategies. Therefore, game theory is used to study the expected performance of agents in many applications of interest. Games can be regarded as a multi-objective optimization problems, as both share many tools of analysis and algorithmic capabilities. Using a game theoretical framework, we propose a novel smart grid application with nonconvex quadratic objectives (a min-max game) and analyze the Nash equilibrium points of the game.

We continue this presentation with a detailed analysis of the state of the art in these areas.

1.1 Nonconvex quadratic problems

Quadratic formulations appear in many problems of interest, both convex and nonconvex, and have been studied for more than 70 years. In this thesis we focus in certain types of quadratic problems, namely the ones that have separable constraints, and search for conditions and guarantees that allow us to solve them optimally and efficiently. There are several results in the literature regarding the optimality of nonconvex QPs, which we will review in this introduction, and in the coming chapters we will present our contributions in the topic.

Nonconvex QPs are considered to be very hard problems, where many instances are known to be NP-hard. In most cases, finding optimal solutions to QPs is out of reach as the problem grows in size. Nevertheless, quadratic models appear in all sort of applications and problems, so their interest is high.

The general form of a quadratically constrained quadratic problem (QCQP) is the following:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & \mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i \leq 0 \quad \forall i \in \mathcal{N}, \end{aligned} \quad (1.1)$$

where $A_0, A_i \in \mathcal{S}^p$ are symmetric matrices, $b_0, b_i, \mathbf{x} \in \mathbb{R}^p$ are column vectors and $c_0, c_i \in \mathbb{R}$ are scalar numbers. Also, we denote with $\mathcal{N} = \{1, \dots, N\}$ the set of constraints.

If A_0 and every A_i are positive semidefinite, then, the problem is convex. If in addition to convexity, Slater's condition holds, then the problem has a tractable solution in polynomial time and it is feasible to solve optimally even for a large number of constraints or a high dimension p [15].

As a quick reminder, Slater's condition requires that given a set of constraints $g_i(\mathbf{x}) \leq 0$ for $i \in \mathcal{N}$, there exists some $\hat{\mathbf{x}}$ such that $g_i(\hat{\mathbf{x}}) < 0$ for every $i \in \mathcal{N}$. This requirement is sometimes referred as *strict feasibility*. If some $g_i(\mathbf{x})$ are affine, the condition can be relaxed to just feasibility in the affine constraints.

There is a wide range of algorithms that can solve a convex QP. From algorithms based on descent directions to dual formulations and interior point methods, there is a very mature and solid theory. Furthermore, distributed and parallel algorithms also allow to solve problems both with millions of variables and/or with millions of data, in very efficient ways [38].

However, in the case that some A_i is not positive semidefinite, then the problem becomes nonconvex, and very likely, difficult to solve globally. We will discuss some of these cases here. Consider for instance “partitioning problems”, also called “boolean optimization” which have the following form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} \\ \text{s.t.} \quad & x_i \in \{-1, 1\} \quad \forall i \in \mathcal{N}. \end{aligned} \quad (1.2)$$

The problem is known to be NP-hard, even if $A_0 \succeq 0$, see for e.g. [41]. It can be transformed to a QCQP if the constraints are treated as quadratic functions, i.e., $\|x_i\|^2 = 1$:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} \\ \text{s.t.} \quad & \|x_i\|^2 = 1 \quad \forall i \in \mathcal{N}. \end{aligned} \quad (1.3)$$

We will show however that if A_0 is a Z-matrix, then the problem can be solved optimally. A Z-matrix is a matrix whose off-diagonal elements are nonpositive.

One famous partitioning problem is called the max-cut, and it is a well known NP-hard problem with many algorithmic proposals. Its objective is to find the set of vertices that maximize the weight of the edges such that one extreme point is in the set and the other is

Fig. 1.1 Six-hump camel back function

not. It is usually formulated as (1.3) where A_0 has an empty diagonal and $(A_0)_{ij} \geq 0$ for all $i \neq j$, see for example section 4 in [6] for a general description. Another famous problem is the *minimum cardinality problem*, which aims to find the minimum support of a vector that satisfies a set of given inequalities. A comprehensive description of these problems is given in [25].

Other examples of nonconvex quadratic problems result from polynomial optimization. This kind of problems can be transformed into QPQCs. One has to reduce the order of the polynomial introducing new variables in the objective and adding them as quadratic constraints.

An example of a polynomial minimization problem with multiple minima is the *six-hump camel back function* [29], given by the following formula:

$$f(x_1, x_2) = x_1^2 \left(4 - 2.1x_1^2 + \frac{1}{3}x_1^4 \right) + x_1x_2 + x_2^2(-4 + 4x_2^2), \quad (1.4)$$

which has two dimensions and six minima, two of them optimal. It is represented in figure 1.1.

The equivalent formulation as QCQP is the following:

$$\begin{aligned} \min_{x_1, x_2, u, v, w} \quad & u\left(4 - 2.1u + \frac{1}{3}v\right) + x_1x_2 + w(-4 + 4w) \\ \text{s.t.} \quad & u = x_1^2, \quad v = u^2, \quad w = x_2^2, \end{aligned} \quad (1.5)$$

and one can guess that simple descent algorithms may not be the best choice to optimize these kind of problems, as they would be highly dependent on the initialization point.

One of the most popular methods to try to solve QPs is through semidefinite relaxation. Given a problem in the form of (1.1), an equivalent formulation is given by

$$\begin{aligned} \min_{X, \mathbf{x}} \quad & \text{tr}(A_0X) + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & \text{tr}(A_iX) + 2b_i^T \mathbf{x} + c_i \leq 0 \quad \forall i \in \mathcal{N} \\ & X = \mathbf{x}\mathbf{x}^T \\ & X \in \mathcal{S}^p, \quad \mathbf{x} \in \mathbb{R}^p, \end{aligned} \quad (1.6)$$

which is still a nonconvex problem because of the rank constraint in X . The semidefinite relaxation consists precisely in modifying such constraint to a convex one of the form $X \succeq \mathbf{x}\mathbf{x}^T$. Applying the Schur complement to the previous relaxation (as explained in Appendix A.5.5 in [15]), one can transform problem (1.6) into

$$\begin{aligned} \min_{X, \mathbf{x}} \quad & \text{tr}(A_0X) + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & \text{tr}(A_iX) + 2b_i^T \mathbf{x} + c_i \leq 0 \quad \forall i \in \mathcal{N} \\ & \begin{bmatrix} X & \mathbf{x} \\ \mathbf{x}^T & 1 \end{bmatrix} \succeq 0 \\ & X \in \mathcal{S}^p, \quad \mathbf{x} \in \mathbb{R}^p. \end{aligned} \quad (1.7)$$

Problem (1.7) is already a convex problem and can be solved using convex optimization tools and techniques such as interior point methods. However, note that (1.7) is not easy to solve in a distributed fashion, and also, it does not scale well with the dimension of the problem. We will consider these aspects in the coming chapters when developing our proposals.

Note that since problem (1.7) is a relaxation of the original QP, its solution gives a lower bound on the achievable minimum value. If the gap between the quadratic program and the semidefinite relaxation is zero, then we say that *strong duality* holds in the QP, and that the relaxation is tight. This occurs on seldom QPs, some of which are well understood and known with a solid theoretical foundation, and other formulations where there is no evidence to why some problems have this property and others do not. In chapter 2 we explore and

analyze some results where we can guarantee *strong duality* for QPs with inequality and equality constraints.

For problems that do not present *strong duality*, a popular methodology to obtain feasible suboptimal results of the QP use randomization techniques. The idea comes from the interpretation that the solution X and $\mathbf{x}$ of the SDP formulation (1.7) correspond to a random variable with covariance matrix $X - \mathbf{x}\mathbf{x}^T$ and mean $\mathbf{x}$. Sampling random points from a Gaussian distribution $\mathbf{x} \sim \mathcal{N}(\mathbf{x}, X - \mathbf{x}\mathbf{x}^T)$ will solve the quadratic problem (1.1) *on average*. Specifically, the distribution solves the following QP:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbb{E}[\mathbf{x}^T A_0 \mathbf{x} + 2\mathbf{b}_0^T \mathbf{x} + c_0] \\ \text{s.t.} \quad & \mathbb{E}[\mathbf{x}^T A_i \mathbf{x} + 2\mathbf{b}_i^T \mathbf{x} + c_i] \leq 0 \quad \forall i \in \mathcal{N}, \end{aligned} \quad (1.8)$$

where $\mathbb{E}[\cdot]$ refers to the expectation operator. Some of the samples $\mathbf{x}$ will be feasible and can be taken as candidate solutions. As more samples are obtained, the best point is kept. Furthermore, these samples can be used as initialization points in algorithms based on descent techniques to improve the result.

Very recently, a new technique based on a *semi-Lagrangian dual problem* have been proposed that may improve the semidefinite relaxation lower bound and close the gap with the QP. See [13] for details on this novel technique. In this case, the technique is valid for QPs with inequality quadratic and affine constraints.

1.1.1 Strong duality results of quadratic programs

We introduce here some basic and classic results regarding strong duality for QPs and briefly discuss some applications of interest. We will comment on these and some other results in [chapter 2](#) also, as a literature review prior to presenting our contributions.

Quadratic problem without constraints

Consider the following QP:

$$\min_{\mathbf{x} \in \mathbb{R}^p} \quad f(\mathbf{x}) = \mathbf{x}^T A_0 \mathbf{x} + 2\mathbf{b}_0^T \mathbf{x} + c_0, \quad (1.9)$$

where $A_0 \in \mathcal{S}^p$. The optimal solution $f(\mathbf{x}^*)$ can be expressed as

$$f(\mathbf{x}^*) = \begin{cases} -\mathbf{b}_0^T A_0^\dagger \mathbf{b}_0 + c_0 & \text{if } A_0 \succeq 0 \wedge \mathbf{b}_0 \in \text{range}[A_0] \\ -\infty & \text{otherwise.} \end{cases} \quad (1.10)$$

The argument that attains the minimum value when $f(\mathbf{x}^*)$ is finite is as follows:

$$\mathbf{x}^* \in -A_0^\dagger b_0 + \text{null}[A_0], \quad (1.11)$$

where $\mathbf{x}^* = -A_0^\dagger b_0$ has minimum norm. If A_0 is positive definite, then the QP is strongly convex and the unique solution is given by $\mathbf{x}^* = -A_0^{-1} b_0$.

Quadratic problem with one inequality constraint

A QP with a single inequality quadratic constraint presents strong duality as long as Slater's condition is satisfied. In particular,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \mathbf{x}^T A_1 \mathbf{x} + 2b_1^T \mathbf{x} + c_1 \leq 0 \end{aligned} \quad (1.12)$$

can be solved optimally if there exists some $\hat{\mathbf{x}}$ such that $g_1(\hat{\mathbf{x}}) < 0$. The Lagrangian of the problem becomes

$$L(\mathbf{x}, \lambda_1) = \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 + \lambda_1 (\mathbf{x}^T A_1 \mathbf{x} + 2b_1^T \mathbf{x} + c_1) \quad (1.13)$$

and the dual function is

$$\begin{aligned} q(\lambda_1) = \min_{\mathbf{x}} L(\mathbf{x}, \lambda_1) \\ \begin{cases} -(b_0 + \lambda_1 b_1)^T (A_0 + \lambda_1 A_1)^\dagger (b_0 + \lambda_1 b_1) + c_0 + \lambda_1 c_1 & \text{if } A_0 + \lambda_1 A_1 \succeq 0 \\ & \wedge b_0 + \lambda_1 b_1 \in \text{range}[A_0 + \lambda_1 A_1] \\ -\infty & \text{otherwise,} \end{cases} \end{aligned} \quad (1.14)$$

where we applied the solution of (1.9) to (1.13).

The dual problem consists of maximizing the dual function, and with a Schur complement transformation we get:

$$\begin{aligned} \max_{\gamma, \lambda_1 \in \mathbb{R}} \quad & \gamma \\ \text{s.t.} \quad & \begin{pmatrix} A_0 + \lambda_1 A_1 & b_0 + \lambda_1 b_1 \\ (b_0 + \lambda_1 b_1)^T & c_0 + \lambda_1 c_1 - \gamma \end{pmatrix} \succeq 0 \\ & \lambda_1 \geq 0, \end{aligned} \quad (1.15)$$

which is an SDP with two variables. Alternatively, the dual problem could have been formulated using (1.7), which achieves the same result.

The first publication to establish strong duality between (1.12) and its dual was made in [85]. A modern and comprehensive description of this result with a complete proof is available in Appendix B in [16].

An application that considers this result is used in the development of trust region methods, which in its most simple formulation allows to solve unconstrained problems of the form

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad (1.16)$$

By finding appropriate representations, one can minimize approximated quadratic problems:

$$\begin{aligned} \min_d \quad & d^T B_k d + 2\nabla f(\mathbf{x}^k)^T d \\ \text{s.t.} \quad & \|d\| \leq \Delta_k, \end{aligned} \quad (1.17)$$

where B_k corresponds to an (approximate) Hessian of $f(\mathbf{x})$ evaluated at $\mathbf{x}^k$. The procedure amounts to solving a set of nonconvex QPs with a single inequality constraint in an iterative manner [4].

Quadratic problem with one equality constraint

Consider now a QP with a single quadratic equality constraint: In particular,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \mathbf{x}^T A_1 \mathbf{x} + 2b_1^T \mathbf{x} + c_1 = 0. \end{aligned} \quad (1.18)$$

The previous problem can be solved optimally and presents strong duality if there exists some $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ such that $g_1(\hat{\mathbf{x}}_1) < 0$ and $g_1(\hat{\mathbf{x}}_2) > 0$, and $A_1 \neq 0$. This result was derived by Theorem 3.2 in [56] and establishes the following necessary conditions for optimality:

$$(A_0 + \lambda_1 A_1) \mathbf{x} + b_0 + \lambda_1 b_1 = 0 \quad (1.19)$$

$$A_0 + \lambda_1 A_1 \succeq 0 \quad (1.20)$$

for some $\mathbf{x}$ and λ_1 , primal and dual optimal, respectively.

An efficient solution to problem (1.18) can be derived solving (1.15) eliminating the $\lambda_1 \geq 0$ constraint, or using the (1.7) formulation.

An immediate application of this result corresponds to the principal component analysis problem (PCA), where the purpose of the problem is to find the vector that maximizes the

output of a quadratic form subject to a norm constraint:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} \\ \text{s.t.} \quad & \|\mathbf{x}\| = 1. \end{aligned} \tag{1.21}$$

On many occasions, the way to solve problem (1.21) consists on performing a singular value decomposition (SVD) and choosing the eigenvector associated with the highest singular value. However, an alternative solution would be to solve the SDP problem dual of (1.21) as we have discussed.

Quadratic problem with two inequality constraints

There are also some results about strong duality regarding QPs with two inequality constraints. Given the following problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \mathbf{x}^T A_1 \mathbf{x} + 2b_1^T \mathbf{x} + c_1 \leq 0 \\ & g_2(\mathbf{x}) = \mathbf{x}^T A_2 \mathbf{x} + 2b_2^T \mathbf{x} + c_2 \leq 0 \end{aligned} \tag{1.22}$$

Assume that $p \geq 3$ and that there exists some $\hat{x}$ such that $g_1(\hat{x}) < 0$ and $g_2(\hat{x}) < 0$. Furthermore, there exists some $\mu_1, \mu_2 \in \mathbb{R}$ such that $\mu_1 A_1 + \mu_2 A_2 \succ 0$ and some $\tau_1, \tau_2 \geq 0$ such that $A_0 - \tau_1 A_1 - \tau_2 A_2 \succ 0$. Then, strong duality holds and an optimal solution is attainable from the SDP relaxation. This result was first presented in [63] in Theorem 4.3, and extended to the complex case in [4]. Other results with one equality and a second inequality constraint are also available in [63].

This result has applications in trust region methods, as proposed in [19]. Following the description from [86], we want to minimize

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & g(\mathbf{x}) = 0. \end{aligned} \tag{1.23}$$

The subproblem to solve at each iteration consists of

$$\begin{aligned} \min_d \quad & d^T \nabla f(\mathbf{x}^k) + \frac{1}{2} d^T B_k d \\ \text{s.t.} \quad & \|g(\mathbf{x}^k) + \nabla g(\mathbf{x}^k)^T d\| \leq \varepsilon \\ & \|d\| \leq \Delta_k \end{aligned} \tag{1.24}$$

where $\nabla g(\mathbf{x}^k)$ corresponds to the gradient of $g(\mathbf{x})$ evaluated at $\mathbf{x}^k$ and B_k an (approximate) Hessian of $f(\mathbf{x})$ evaluated at $\mathbf{x}^k$. This transformation allows to derive an iterative procedure to find solutions to (1.23), and where the strong duality results presented in [63] give a theoretical foundation to such methodology.

1.2 Distributed optimization techniques

We briefly discuss several decomposition techniques that we will be using throughout this manuscript.

1.2.1 ADMM decomposition

Alternating direction method of multipliers (ADMM) is a popular technique that has gained importance in the last years to achieve distributed and parallel schemes in optimization problems. It was first proposed by Hestenes and Powell in [44, 65], and later extended and developed by many different authors. A nice and updated survey comes from [14], from which we borrow the following description.

Consider a convex optimization problem with one equality constraint:

$$\begin{aligned} \min_{x,z} \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c, \end{aligned} \tag{1.25}$$

where $x \in \mathbb{R}^p$, $z \in \mathbb{R}^m$, $A \in \mathbb{R}^{N \times p}$, $B \in \mathbb{R}^{N \times m}$. We can form an augmented Lagrangian and aim at a sequential scheme to solve (1.25):

$$L_\rho(x, z, \mu) = f(x) + g(z) + \rho \|Ax + Bz - c\|^2 + \mu^T (Ax + Bz - c), \tag{1.26}$$

where $\mu \in \mathbb{R}^N$ is a dual variable and $\rho > 0$ makes under mild conditions $L(x, z, \mu)$ strongly convex in x and z for fixed z and x , respectively. The Lagrangian is not strongly convex in x and z jointly.

The iterative procedure consists of solving the following problems in sequential order:

1. $x^{k+1} = \arg \min_x L_\rho(x, z^k, \mu^k)$.
2. $z^{k+1} = \arg \min_z L_\rho(x^{k+1}, z, \mu^k)$.
3. $\mu^{k+1} = \mu^k + 2\rho(Ax^{k+1} + Bz^{k+1} - c)$.

It can be shown that if $f(x)$ and $g(z)$ are closed, proper and convex, and there exists a saddle point for the unaugmented Lagrangian, i.e., when $\rho = 0$, then, there is residual convergence (variables x and z become feasible), objective convergence (the objective function of the iterates approach the optimal value) and dual variable convergence (dual variable converges to its optimal point). Under additional assumptions, x^k and z^k also converge to their optimal values.

The framework is very versatile, allowing for many different problem types, such as constrained optimization, L1-normed problems, consensus or sharing, or distributed model fitting among some others. In this thesis we are interested in the consensus approach which applies to the following type of problem

$$\min_x f(x) = \sum_{i=1}^N f_i(x) \quad (1.27)$$

where each $f_i(x)$ is convex. The framework introduces a slack variable z and forces equality among local variables x_i :

$$\begin{aligned} \min_{x_i \forall i, z} \quad & \sum_{i=1}^N f_i(x_i) \\ \text{s.t.} \quad & x_i = z \quad \forall i \in \{1, \dots, N\}, \end{aligned} \quad (1.28)$$

where $x_i \in \mathbb{R}^p$ and $z \in \mathbb{R}^p$.

Forming the augmented Lagrangian as in (1.26) and following the standard steps we arrive at the following update scheme:

1. $\min_{x_i} f_i(x_i) + (\mu_i^k)^T (x_i - z^k) + \rho \|x_i - z^k\|^2$ for all i (can be solved in parallel).
2. $z^{k+1} = \frac{1}{N} \sum_{i=1}^N \left(x_i^{k+1} + \frac{1}{2\rho} \mu_i^k \right)$.
3. $\mu_i^{k+1} = \mu_i^k + 2\rho (x_i^{k+1} - z^{k+1})$ for all i .

If every f_i is closed, proper and convex, we arrive at same convergence results as in the standard problem (1.25).

This framework is useful to decompose centralized problems and reach global solutions where agents have their own local private function and only need to exchange the consensus variable z between all of them.

1.2.2 FLEXA decomposition

FLEXA is a novel framework introduced in [70] as part of a general decomposition framework with the aim to solve a sum of nonconvex problems in multiagents systems. It was later

extended in [38] to incorporate non-smooth terms as well as a flexible block update scheme with particular emphasis in big data applications. The name is an acronym for “flexible parallel algorithm”. A later extension also considered asynchronous update rules [18].

The starting problem for FLEXA is

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \sum_{i=1}^N f_i(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{K} \end{aligned} \quad (1.29)$$

and where $f_i(\mathbf{x})$ need not be convex. The framework also assumes the Cartesian separability of the constraints, i.e., $\mathcal{K} = \prod_{i=1}^N \mathcal{K}_i$ for some partition. Other assumptions on (1.29) are described in [70].

The framework solves a series of convex surrogate problems that approximate the non-convex original problem at every iteration. Consider a series of surrogate functions $\tilde{f}_i(x_i; \mathbf{x}^k)$ such that their sum approximates the derivative of $f(\mathbf{x})$ at every $\mathbf{x}^k$ and corresponds also to an overestimator at such point. Then, the following surrogate functions decompose the original problem (1.29):

$$\tilde{f}_i(x_i; \mathbf{x}^k) = f_i(x_i) + \pi_i(\mathbf{x}^k)^T (x_i - x_i^k) + \tau_i \|x_i - x_i^k\|^2, \quad \forall i, \quad (1.30)$$

where $\mathbf{x}^k = (x_i^k)_i$, and $\tau_i \geq 0$ enforces strong convexity over $\tilde{f}_i(x_i; \mathbf{x}^k)$. We also define

$$\pi_i(\mathbf{x}^k) = \sum_{j \neq i} \nabla_{x_i} f_j(\mathbf{x}) \big|_{\mathbf{x}=\mathbf{x}^k}, \quad (1.31)$$

which refers to the other agents derivatives that influence the i 'th own local problem. For simplicity we assumed that $f_i(x_i)$ are convex in (1.30), but the nonconvex parts could be linearized and the surrogate function would be similar and still satisfy the required properties.

The types of surrogate functions that can be used are many, and we refer the interested reader to the original articles [70, 38] for more details, examples and applications, which also deal with several types of nonconvexity.

To conclude the introduction to the FLEXA framework, we present a simplified version that computes exact solutions in [algorithm 1](#). The convergence results are described in Theorem 3 in [70].

1.2.3 NOVA decomposition

The NOVA framework is an extension of the FLEXA algorithm that includes nonconvex constraints that can be coupled among several agents. It was presented in [69], and it is named

Algorithm 1 Exact Jacobi FLEXA implementation

-
- 1: Initialize $\tau_i \geq 0$, $\mathbf{x}^0 = (x_i)_i \in \mathcal{K}$ and $\{\alpha^k\} > 0$. Set $k \leftarrow 0$.
 - 2: **while** $\mathbf{x}^k$ is not a stationary point **do**
 - 3: For all i , solve $\hat{x}_i(x^k) = \arg \min \tilde{f}_i(x_i; \mathbf{x}^k)$ **in parallel**.
 - 4: $x_i^{k+1} = x_i^k + \alpha^k(\hat{x}_i(x^k) - x_i^k)$
 - 5: Set $k \leftarrow k + 1$.
 - 6: **end while**
-

after “inner convex approximation”. As a particular case of the original NOVA decomposition problem, we consider a sum of functions in the objective and shared constraints:

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \sum_{i=1}^N f_i(\mathbf{x}) \\
 \text{s.t.} \quad & g_j(\mathbf{x}) \leq 0 \quad \forall j \in \{1, \dots, M\}, \\
 & \mathbf{x} \in \mathcal{K}
 \end{aligned} \tag{1.32}$$

where the problem requires a Cartesian decomposition of $\mathcal{K} = \prod_{i=1}^N \mathcal{K}_i$. Note that N indicates the number of local functions and M the number of constraints.

The NOVA framework aims to solve a series of strongly convex surrogate problems that approximate the original nonconvex formulation:

$$\begin{aligned}
 (P^k) : \quad & \min_{\mathbf{x}} \quad \sum_{i=1}^N \tilde{f}_i(x_i; \mathbf{x}^k) \\
 \text{s.t.} \quad & \tilde{g}_j(\mathbf{x}; \mathbf{x}^k) \leq 0 \quad \forall j \in \{1, \dots, M\}, \\
 & x_i \in \mathcal{K}_i \quad \forall i \in \{1, \dots, N\}
 \end{aligned} \tag{1.33}$$

with $\mathbf{x} = (x_i)_{i=1}^N$. Function $\tilde{f}_i(x_i; \mathbf{x}^k)$ is a strongly convex surrogate function that has to satisfy Assumption 2 in [69]). One possibility is

$$\tilde{f}_i(x_i; \mathbf{x}^k) = f_i(x_i) + \pi_i(\mathbf{x}^k)^T (x_i - x_i^k) + \tau_i \|x_i - x_i^k\|^2, \tag{1.34}$$

where

$$\pi_i(\mathbf{x}^k) = \sum_{r \neq i} \nabla_{x_i} f_r(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}^k}. \tag{1.35}$$

For the purposes of this introduction it suffices to consider that f_i are convex, so we did not decompose (1.34) any further to include nonconvexities.

Function $\tilde{g}_j(\mathbf{x}; \mathbf{x}^k)$ is a convex surrogate of $g_j(\mathbf{x})$ that has to satisfy requirements specified in Assumption 3 in [69]. If $g_j(\mathbf{x})$ is convex, we can have $g_j(\mathbf{x}) = \tilde{g}_j(\mathbf{x}; \mathbf{x}^k)$. Additionally, if

$\tilde{g}_j(\mathbf{x}; \mathbf{x}^k)$ is separable, i.e., $\tilde{g}_j(\mathbf{x}; \mathbf{x}^k) = \sum_i \tilde{g}_j^i(x_i; \mathbf{x}^k)$, then the problem can be fully decomposed into a distributed formulation.

Consider the dual problem of (P^k) whenever $\tilde{g}_j(\mathbf{x}; \mathbf{x}^k)$ is separable:

$$\max_{\boldsymbol{\lambda} \geq 0} q(\boldsymbol{\lambda}; \mathbf{x}^k) \quad (1.36)$$

where $\boldsymbol{\lambda} = (\lambda_i)_{i=1}^N$ and

$$q(\boldsymbol{\lambda}; \mathbf{x}^k) = \min_{\mathbf{x} \in \mathcal{K}} \left\{ \sum_{i=1}^N \tilde{f}_i(x_i; \mathbf{x}^k) + \sum_{j=1}^M \lambda_j \tilde{g}_j^i(x_i; \mathbf{x}^k) \right\}. \quad (1.37)$$

Denoting the solution of (1.37) with $\hat{x}(\boldsymbol{\lambda}; \mathbf{x}^k) = (\hat{x}_i(\boldsymbol{\lambda}; \mathbf{x}^k))_{i=1}^N$, and

$$\hat{x}_i(\boldsymbol{\lambda}; \mathbf{x}^k) = \arg \min_{x_i \in \mathcal{K}_i} \tilde{f}_i(x_i; \mathbf{x}^k) + \sum_{j=1}^M (\lambda_j^i)^T (\tilde{g}_j^i(x_i)), \quad (1.38)$$

the original problem is now decomposed into N independent problems. The aim is to solve the dual problem in a distributed fashion and use FLEXA to solve (1.33) while solving (1.36) in a distributed fashion. The specific description is presented in [algorithm 2](#).

The convergence results are presented in Theorems 2 and 4 from [69].

Algorithm 2 Distributed NOVA implementation

- 1: Initialize $\tau_i \geq 0$, $\mathbf{x}^0 = (x_i)_{i \in \mathcal{K}}$ and $\{\alpha^k\} > 0$. Set $k \leftarrow 0$.
 - 2: **while** $\mathbf{x}^k$ is not a stationary point **do**
 - 3: Initialize λ_j^0 for all j , $\{\alpha^n\} > 0$; set $n \leftarrow 0$.
 - 4: **while** $\boldsymbol{\lambda}^n$ does not satisfy a termination criterion **do**
 - 5: Solve $\hat{x}_i(\boldsymbol{\lambda}^n; \mathbf{x}^k)$ **in parallel** for all i .
 - 6: Update $\lambda_j^{n+1} = \lambda_j^n + \alpha^n \sum_{i=1}^N \tilde{g}_j^i(\hat{x}_i(\boldsymbol{\lambda}^n; \mathbf{x}^k); \mathbf{x}^k)$ for all $j \in \{1, \dots, M\}$.
 - 7: Set $n \leftarrow n + 1$.
 - 8: **end while**
 - 9: $x_i^{k+1} = x_i^k + \alpha^k (\hat{x}_i(\mathbf{x}^k) - x_i^k)$
 - 10: Set $k \leftarrow k + 1$.
 - 11: **end while**
-

1.3 Introduction to algorithmic game theory

Many authors regard *algorithmic game theory* as a new branch of *game theory*, in which the computability and the examination of the equilibria of interest play a central role [37, 60]. In

this introduction we will focus on some standard methods recently proposed to find Nash equilibria (NE) in distributed settings. We will give a very short description of the main concepts presented in [71] that we will later use in chapter 4.

Consider a game where each player $i \in \mathcal{N}$ has to solve a minimization problem coupled with other player's interests. We define the NEP (Nash Equilibrium Problem) as follows:

$$\mathcal{G}_{\text{NEP}} : \begin{cases} \min_{x_i} & f_i(x_i, \mathbf{x}_{-i}) \\ \text{s.t.} & x_i \in \mathcal{K}_i, \end{cases} \quad \forall i \in \mathcal{N}, \quad (1.39)$$

where $\mathbf{x} = (x_i)_{i=1}^N \in \mathbb{R}^p$ and $\mathbf{x}_{-i} = (x_j)_{j \neq i}$ refer to the global set of strategies, and to the other player's strategies, respectively. We define the set $\mathcal{K} = \prod_{i=1}^N \mathcal{K}_i$. The solution concept of interest is defined as follows:

Definition 1. A pure strategy NE, is a feasible point x^* that satisfies

$$f_i(x_i^*, \mathbf{x}_{-i}^*) \leq f_i(x_i, \mathbf{x}_{-i}^*), \quad \forall x_i \in \mathcal{K}_i \quad (1.40)$$

for every player $i \in \mathcal{N}$.

Alternatively, we can have a generalized Nash equilibrium problem (GNEP) defined as follows:

$$\mathcal{G}_{\text{GNEP}} : \begin{cases} \min_{x_i} & f_i(x_i, \mathbf{x}_{-i}) \\ \text{s.t.} & x_i \in \mathcal{K}_i(\mathbf{x}_{-i}), \end{cases} \quad \forall i \in \mathcal{N}, \quad (1.41)$$

where the feasible space of each player i depends on the strategies of other players, i.e., $x_i \in \mathcal{K}_i(\mathbf{x}_{-i})$. The definition of NE remains the same, only that now the feasibility constraint changes with the other player's strategies.

In chapter 4 we will have both kind of games, as we will consider a min-max game where the inner game consists of a NEP, but the outer game has coupled constraints.

GNEPs are in general much more difficult problems than NEPs. However, under jointly convex shared constraints for every $\mathbf{x}$, that is,

$$\mathcal{K}(\mathbf{x}) = \prod_{i \in \mathcal{N}} \mathcal{K}_i(\mathbf{x}_{-i}) \quad (1.42)$$

be a convex set, then there is some short of accessible theory to analyze GNEPs. In such case, the existence and algorithmic theory to attain a solution can be analyzed within the theory of variational inequalities (VIs).

A $\text{VI}(\mathcal{K}, \mathbf{F})$ consists of a set $\mathcal{K}$, which we will assume to be convex, and a mapping $F : \mathbb{R}^p \rightarrow \mathbb{R}^p$. The objective of a VI problem, is to find a point $\mathbf{x}^* \in \mathcal{K}$ that satisfies:

$$(\mathbf{x} - \mathbf{x}^*)^T \mathbf{F}(\mathbf{x}^*) \geq 0, \quad \forall \mathbf{x} \in \mathcal{K}. \quad (1.43)$$

Many problems in optimization, game theory or complementarity problems have a VI reformulation. In particular, if a NEP $\mathcal{G}_{\text{NEP}}$ has every $\mathcal{K}_i$ closed and convex, and every $f_i(x_i, \mathbf{x}_{-i})$ is convex and continuously differentiable in $x_i \in \Omega_i \supset \mathcal{K}_i$, then, the game $\mathcal{G}$ is equivalent to the $\text{VI}(\mathcal{K}, \mathbf{F})$, where $\mathbf{F} = (\nabla_{x_i} f_i(x_i, \mathbf{x}_{-i}))_{i=1}^N$. This result comes from Proposition 4.1 in [71].

Transforming a NEP into a VI is beneficial because the theory of VI's is very mature and helps not only to establish conditions for the existence of solutions, but also provides and algorithmic framework to find the solution of interest.

We can state that if the VI is monotone and $\mathcal{K}$ is convex, then the VI (and the NEP $\mathcal{G}$) has a convex solution set. Furthermore, there are distributed algorithms available with convergence guarantees to such solution, if it exists. These results are described in Theorem 4.1 and the distributed scheme is presented in Algorithm 4.1, both in [71].

The monotonicity property is a desirable property for VIs (or games) that works with similar implications as the role convexity plays in optimization problems. We can determine if a VI is monotone checking the following condition:

$$\mathbf{F}(\mathbf{x}) \text{ is monotone on } \mathcal{K} \iff \nabla \mathbf{F}(\mathbf{x}) \succeq 0, \quad \forall \mathbf{x} \in \mathcal{K}, \quad (1.44)$$

where $\nabla \mathbf{F}(\mathbf{x})$ corresponds to the Jacobian of $\mathbf{F}(\mathbf{x})$.

As a concluding remark, VIs and NEPs are well understood and we have efficient methods to determine their solutions if they are monotone. Similarly, GNEPs with jointly convex shared constraints also have specific algorithms to find their solutions optimally. On the other hand, the lack of monotonicity constitutes a challenging task where each problem may have to be studied in its own setting.

1.4 Contributions

Some of our contributions in this thesis have been published in journals of high impact, or international conferences. Others contributions are planned to be published but have not been submitted yet.

An early work from [chapter 2](#) was published in the following conference:

- Javier Zazo, Santiago Zazo, and Sergio Valcarcel Macua, “Non-monotone quadratic potential games with single quadratic constraints,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4373–4377, doi:[10.1109/ICASSP.2016.7472503](https://doi.org/10.1109/ICASSP.2016.7472503).

One of the algorithmic proposals we describe in [chapter 3](#) was published in:

- Javier Zazo, Sergio Valcarcel Macua, Santiago Zazo, Marina Pérez, Iván Pérez-Álvarez, Eugenio Jiménez, Laura Cardona, Joaquín Hernández Brito, and Eduardo Quevedo, “Underwater electromagnetic sensor networks, part II: Localization and network simulations,” *Sensors*, vol. 16, no. 12, pp. 2176, 2016, doi:[10.3390/s16122176](https://doi.org/10.3390/s16122176).

A complete version of [chapter 4](#) was published in:

- Javier Zazo, Santiago Zazo, and Sergio Valcarcel Macua, “Robust worst-case analysis of demand-side management in smart grids,” *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 662–673, March 2017, doi:[10.1109/TSG.2016.2559583](https://doi.org/10.1109/TSG.2016.2559583).

1.4.1 Other contributions

We have also other publications made in collaboration with our colleagues.

Dynamic game theory

- Santiago Zazo, Sergio Valcarcel Macua, Matilde Sánchez-Fernández, and Javier Zazo, “Dynamic potential games with constraints: Fundamentals and applications in communications,” *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3806–3821, July 2016, doi:[10.1109/TSP.2016.2551693](https://doi.org/10.1109/TSP.2016.2551693).
- Santiago Zazo, Javier Zazo, and Matilde Sánchez-Fernández, “A control theoretic approach to solve a constrained uplink power dynamic game,” in *22nd European Signal Processing Conference (EUSIPCO)*, Sept 2014, pp. 401–405, <http://ieeexplore.ieee.org/document/6952099/>.
- Santiago Zazo, Sergio Valcarcel Macua, Matilde Sánchez-Fernández, and Javier Zazo, “A new framework for solving dynamic scheduling games,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 2071–2075, doi:[10.1109/ICASSP.2015.7178335](https://doi.org/10.1109/ICASSP.2015.7178335).

- Sergio Valcarcel Macua, Santiago Zazo, and Javier Zazo, “Learning in constrained stochastic dynamic potential games,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 4568–4572, doi:10.1109/ICASSP.2016.7472542.
- Sergio Valcarcel Macua, Javier Zazo, and Santiago Zazo, “Closed-loop stochastic dynamic potential games with parametric policies and constraints,” in *Learning, Inference and Control of Multi-Agent Systems workshop in NIPS*, 2016.

Algorithmic game theory

- Javier Zazo, Santiago Zazo, and Sergio Valcarcel Macua, “Distributed cognitive radio systems with temperature-interference constraints and overlay scheme,” in *22nd European Signal Processing Conference (EUSIPCO)*, Sept 2014, pp. 855–859, <http://ieeexplore.ieee.org/document/6952270/>.

Distributed optimization

- Sergio Valcarcel Macua, Santiago Zazo, and Javier Zazo, “Distributed black-box optimization of nonconvex functions,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 3591–3595, doi:10.1109/ICASSP.2015.7178640.
- Sergio Valcarcel Macua, Carlos Moreno Leon, J. S. Romero, S. S. Pereira, Javier Zazo, Alba Pagés-Zamora, R. López-Valcarce, and Santiago Zazo, “How to implement doubly-stochastic matrices for consensus-based distributed algorithms,” in *IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 333–336, doi:10.1109/SAM.2014.6882409.

Differential and mean field games

- Juan Parras, Santiago Zazo, Jorge del Val, Javier Zazo, and Sergio Valcarcel Macua, “Pursuit-evasion games: a tractable framework for antijamming games in aerial attacks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 69, 2017 doi:10.1186/s13638-017-0857-8.
- Juan Parras, Jorge del Val, Santiago Zazo, Javier Zazo, and Sergio Valcarcel Macua, “A new approach for solving anti-jamming games in stochastic scenarios as pursuit-evasion games,” in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2016, pp. 1–5, doi:10.1109/SSP.2016.7551804.

- Jorge del Val, Santiago Zazo, Sergio Valcarcel Macua, Javier Zazo, and Juan Parras, “Optimal attack and defence of large scale networks using mean field theory,” in *24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 973–977, [doi:10.1109/EUSIPCO.2016.7760393](https://doi.org/10.1109/EUSIPCO.2016.7760393).

1.5 Outline

The thesis is structured as follows:

In [chapter 2](#) we introduce quadratic programs with inequality and equality constraints and give sufficient conditions for the QP to have the strong duality property. Then, we analyze the existence of solutions and propose different algorithms to solve the nonconvex problems. We finish the chapter with an appendix containing all the proofs from our theoretical claims.

In [chapter 3](#) we introduce two problems: the target localization and the sensor network localization problems. Both can have a nonconvex quadratic reformulation. We propose algorithms with guarantees of convergence to suboptimal stationary solutions, as well as distributed algorithms based on dual methods with convergence to a global solution provided that a certain constraint qualification is satisfied. We conclude the chapter with several simulations illustrating the behavior of our algorithms.

In [chapter 4](#) we describe our robust model in a demand-side-management framework, considering a day-ahead cost function as well as a real-time energy market. We formulate the problem as a min-max game, where the maximization game is non-monotone (and nonconvex). We analyze the game and propose a distributed algorithm with convergence guarantees. We provide several simulations illustrating our results.

Finally, in [chapter 5](#) we present our conclusions and discuss future lines of work.

Chapter 2

Strong duality in nonconvex quadratic problems with separable quadratic constraints

We study nonconvex quadratic problems with quadratic constraints (QPs) which do not overlap with each other. These constraints can be defined both as inequalities or equalities. We derive sufficient conditions for these types of problems in order to present the $\mathcal{S}$ -property, which ultimately guarantees strong duality between the primal and dual problems of the QP. We study the existence of solutions of these QPs and propose several algorithms to solve the problems optimally when the $\mathcal{S}$ -property is satisfied, both centralized and in parallel.

2.1 Introduction

Nonconvex quadratic problems with quadratic constraints have numerous applications and have been extensively studied for more than 70 years [39]. They are normally considered to be NP-hard problems, or at least, as hard as a large number of other problems which are also assumed to have a high degree of complexity [41]. In this chapter we consider a subclass of QPs where the constraints do not overlap with each other and we give sufficient conditions on the problem formulation that guarantee strong duality between the primal problem and its dual. From our results, we are able to identify specific cases in which these problems can be solved optimally, and provide some algorithmic results for these cases.

We already discussed in [chapter 1](#) some of the few results in the literature regarding nonconvex quadratic problems that present strong duality. The first one was established in [85] with the proposal of the $\mathcal{S}$ -Lemma. Combined with previous results regarding the

field of values of two matrices [28, 17, 43], strong duality was characterized for nonconvex quadratic problems with a single quadratic constraint. An accessible proof of such a result is available in [15, Appx. B] and an extension that considers a single equality constraint rather than an inequality is analyzed in [56]. Strong duality for two quadratic constraints was analyzed in [63] for the real case and it was extended in [4] to the complex case.

However, results regarding an arbitrary number of constraints are very limited. One relevant result is [7], which establishes strong duality in a beamforming application where the objective is convex and the constraints consist of differences of convex functions with a given structure. In this case, the application is very specific, but the approach for the proof is novel. Further results regarding the existence of rank 1 solutions in similar beamforming settings have also been proposed in [49].

Another result that has an arbitrary number of constraints is established when the Hessian matrices corresponding to the objective and constraints are simultaneously diagonalizable via congruence [50]. In this circumstance, the problem can be transformed into a diagonalized version and solved efficiently as a linear program. However, such a requirement is extremely restrictive, and there is currently no known method for characterizing diagonalizable matrices via congruence simultaneously. Proposition 3.5 in [62] presents another result on strong duality which requires that the Hessian matrices of the quadratic objective and constraints are a linear combination of only two of those matrices. Two survey papers describing these and some additional results can be found in [46] and [62].

Reference [51] presented new results regarding strong duality in QPs with inequality constraints whenever the objective and constraints transformed to matrix form had a Z-matrix structure (all the off-diagonal elements are nonpositive). Our results consider a similar problem applied to separable constraints, but also considering quadratic equality constraints (as long as they do not overlap). To the best of our knowledge, the consideration of quadratic equality constraints is novel in the literature, and our results cannot be derived from [51].

Other interesting results for nonconvex QPs are provided in [52], which analyzes necessary and/or sufficient conditions on some quadratic problems for optimality. Specifically, they consider a weighted nonconvex least squares problem with convex quadratic constraints, and a partitioning problem (referred to as bivalent nonconvex problem), but do not establish strong duality requirements in these cases. Similarly, [6] considers global optimality conditions for quadratic problems with binary constraints. Both references provide conditions to establish optimality for candidate points, but do not provide any methods nor guarantees to find such solutions. Finally, [13] analyzes the use of copositive relaxations to obtain better lower bounds for general nonconvex QPs than semidefinite relaxations. This result brings closer the non-zero duality gap in hard problems.

Regarding our contribution, we establish strong duality results for problems that have any number of constraints, where the variables of the optimization problem appear in a single constraint but in no other one, i.e., they are non-overlapping constraints. At most, we can only have as many constraints in the problem as the dimension of the unknown variable. Many applications fall into this category, namely, Boolean least squares, partitioning problems, the max-cut, robust least squares, etc. We consider both equality constraints and inequality, providing a new analysis methodology, as well as specific results where strong duality occurs.

The chapter is structured as follows: in [section 2.2](#) we introduce the QPs with inequality and equality constraints. In [section 2.2.1](#) we present the dual problem derived from the QP and define the constraint qualification known as the $\mathcal{S}$ -property necessary to establish strong duality between the primal and dual problems. In [section 2.3](#) we derive sufficient conditions for a QP with non-overlapping constraints to satisfy the $\mathcal{S}$ -property and in [section 2.4](#) we derive sufficient conditions for the existence of solutions. In [section 2.5](#) we discuss different methods to find an optimal solution of the QP whenever the $\mathcal{S}$ -property is satisfied and, finally, in [section 2.6](#) we illustrate our results. Some of the proofs have been moved to [appendix 2.A](#) in order to improve readability.

2.2 Problem formulation

We consider the following QP with inequality constraints:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & x_i^T \tilde{A}_i x_i + 2\tilde{b}_i^T x_i + c_i \leq 0 \quad \forall i \in \mathcal{N}, \end{aligned} \tag{QPI}$$

where $\mathbf{x} \in \mathbb{R}^p$ is a column vector of size p ; $x_i \in \mathbb{R}^{n_i}$ is the i th subblock of vector $\mathbf{x}$ and has size n_i ; we suppose that $\mathbf{x}$ has N blocks, and writing $\mathcal{N} = \{1, \dots, N\}$, we have $\mathbf{x} = (x_i)_{i \in \mathcal{N}}$ and $p = \sum_{i \in \mathcal{N}} n_i$. Matrix $A_0 \in \mathcal{S}^p$ is a symmetric matrix of size $p \times p$; $b_0 \in \mathbb{R}^p$ is a column vector; c_i for all i and $c_0 \in \mathbb{R}$ are scalar numbers; $\tilde{A}_i \in \mathcal{S}^{n_i}$ for all i are symmetric matrices; and finally, $\tilde{b}_i \in \mathbb{R}^{n_i}$ for all i are column vectors. Note that we did not assume that A_0 and $\tilde{A}_i$ are positive semidefinite matrices, and, therefore, the problem is in general nonconvex.

In addition to problem [QPI](#), we also consider the following QP:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & x_i^T \tilde{A}_i x_i + 2\tilde{b}_i^T x_i + c_i = 0 \quad \forall i \in \mathcal{N}, \end{aligned} \tag{QPE}$$

where we substituted the inequality constraints with equality constraints. This problem is novel in the literature with no existing strong duality results for more than a single equality constraint.

We introduce some notation that will be useful for our presentation

$$b_i = [0_{n_1 \times 1}^T, \dots, \tilde{b}_i^T, \dots, 0_{n_N \times 1}^T]^T, \quad (2.1)$$

$$A_i = \text{diag}[0_{n_1 \times n_1}, \dots, \tilde{A}_i, \dots, 0_{n_N \times n_N}], \quad (2.2)$$

where the previous expressions have nonzero elements in i th positions, $b_i \in \mathbb{R}^p$ and $A_i \in \mathcal{S}^p$. Furthermore, we also introduce matrices

$$\bar{A}_0 = \begin{pmatrix} A_0 & b_0 \\ b_0^T & c_0 \end{pmatrix}, \quad \bar{A}_i = \begin{pmatrix} A_i & b_i \\ b_i^T & c_i \end{pmatrix} \quad \forall i \in \mathcal{N} \quad (2.3)$$

that combine all the elements that appear in the quadratic problem. Note that $\bar{A}_0$ and $\bar{A}_i \in \mathcal{S}^{p+1}$.

We define the following functions:

$$f(\mathbf{x}) = \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0, \quad (2.4)$$

which is the objective function of the QP; and, with a slight abuse of notation,

$$g_i(x_i) = x_i^T \tilde{A}_i x_i + 2\tilde{b}_i^T x_i + c_i = \mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i = g_i(\mathbf{x}), \quad (2.5)$$

which denotes the i th constraint.

We make the following assumption:

Assumption 1 (Slater's condition). *Assume the following:*

*A: For problem **QPI**, there $\exists \hat{x}_i \in \mathbb{R}^{n_i} \mid g_i(\hat{x}_i) < 0$ for all $i \in \mathcal{N}$.*

*B: For problem **QPE**, there $\exists \hat{x}_i^1, \hat{x}_i^2 \in \mathbb{R}^{n_i} \mid g_i(\hat{x}_i^1) < 0 \wedge g_i(\hat{x}_i^2) > 0$ for all $i \in \mathcal{N}$.*

The previous assumption guarantees some regularity condition that will be necessary in the upcoming derivations. We remark that **item B**: was introduced as a constraint qualification for a QP with a single quadratic equality constraint in [56].

2.2.1 Dual formulation

The Lagrangian of the QPs has the following form

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 + \sum_{i \in \mathcal{N}} \lambda_i (\mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i), \quad (2.6)$$

where we introduced dual variables $\boldsymbol{\lambda} = (\lambda_i)_{i \in \mathcal{N}}$ and $\lambda_i \in \mathbb{R}$. The dual function $q(\boldsymbol{\lambda}) = \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$ is given by

$$q(\boldsymbol{\lambda}) = \begin{cases} -(b_0 + \sum_i \lambda_i b_i^T)(A_0 + \sum_i \lambda_i A_i)^\dagger (b_0 + \sum_i \lambda_i b_i^T) + c_0 + \sum_i \lambda_i c_i & \text{if } A_0 + \sum_i \lambda_i A_i \succeq 0 \text{ and } (b_0 + \sum_i \lambda_i b_i) \in \text{range}(A_0 + \sum_i \lambda_i A_i) \\ -\infty & \text{otherwise.} \end{cases} \quad (2.7)$$

The dual problems of **QPI** and **QPE** can be rewritten using the Schur complement as

$$\begin{aligned} \max_{\gamma, \boldsymbol{\lambda}} \quad & \gamma \\ \text{s.t.} \quad & \begin{pmatrix} A_0 + \sum_i \lambda_i A_i & b_0 + \sum_i \lambda_i b_i \\ (b_0 + \sum_i \lambda_i b_i)^T & c_0 + \sum_i \lambda_i c_i - \gamma \end{pmatrix} \succeq 0 \\ & \gamma \in \mathbb{R}, \quad \boldsymbol{\lambda} \in \Gamma \end{aligned} \quad (2.8)$$

where we introduced the set Γ as a general way to constrain $\boldsymbol{\lambda}$ in problem **QPI** and **QPE** accordingly, i.e., $\Gamma = \mathbb{R}_+^N$ in **QPI**, and $\Gamma = \mathbb{R}^N$ in **QPE**. Furthermore, $\Gamma = \Gamma_1 \times \cdots \times \Gamma_N$, where $\Gamma_i = \mathbb{R}_+$ in **QPI** or $\Gamma_i = \mathbb{R}$ in **QPE**.

The dual problem given by (2.8) is an underestimator of the optimal values of **QPI** and **QPE**. Under the strong duality property the solution of the primal problem and the dual problem coincide, and can be efficiently determined.

In order to derive the conditions that guarantee strong duality, we will be using results from [52]. Before introducing the theorem that will motivate our proof, we first define the constraint qualification required for our QPs:

Definition 2 (*S-property* [52]). *A QP satisfies the S-property, if and only if, the following system of equations are equivalent $\forall \alpha \in \mathbb{R}$:*

$$g_i(x_i) \leq 0, \forall i \in \mathcal{N} \implies f(\mathbf{x}) \geq \alpha, \quad \iff \quad (2.9a)$$

$$\exists \boldsymbol{\lambda} \in \Gamma \mid f(\mathbf{x}) + \sum_{i \in \mathcal{N}} \lambda_i g_i(x_i) \geq \alpha, \forall \mathbf{x} \in \mathbb{R}^p. \quad (2.9b)$$

We use the symbol ‘ $\trianglelefteq$ ’ to refer in a general way to either an equality or an inequality, when considering the constraints of an optimization problem. More specifically, $\trianglelefteq \in \{\leq, =\}$.

It is immediate to see that (2.9b) $\Rightarrow$ (2.9a): suppose (2.9b) is satisfied and $g_i(x_i) \trianglelefteq 0$. Then $\sum_{i \in \mathcal{N}} \lambda_i g_i(x_i) \leq 0$ and we necessarily have that $f(\mathbf{x}) \geq \alpha$. The other implication is not always true and is difficult to know when it is satisfied. The main result of this chapter is dedicated to establish the requirements of when such implication holds.

We can express the system of equations from the definition of the $\mathcal{S}$ -property as a system of alternatives rather than a system of equivalencies. This is helpful because we will be able to derive a theorem of alternative systems based on the new formulation. A description of the theory of alternatives can be found in section 5.8 in [15]. We reformulate the $\mathcal{S}$ -property with the following lemma:

Lemma 1 ($\mathcal{S}$ -property reformulation). *A QP possesses the $\mathcal{S}$ -property if and only if, the following systems are strong alternatives for every $\alpha \in \mathbb{R}$:*

$$\exists \mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < \alpha \wedge g_i(x_i) \trianglelefteq 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.10a)$$

$$\nexists \boldsymbol{\lambda} \in \Gamma \mid \bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{\mathbf{E}} \succeq 0. \quad (2.10b)$$

We introduced $\bar{\mathbf{E}}$ to denote a square matrix of size $(p+1)$ with single nonzero element $(\bar{\mathbf{E}})_{p+1,p+1} = 1$.

Proof. See [appendix 2.A](#). □

The following theorem reflects the importance of the $\mathcal{S}$ -property:

Theorem 1 (Necessary and sufficient conditions, Theorem 3.1 in [52]). *Let $\mathbf{x}^*$ be a feasible point of the QP. Suppose that the problem satisfies the $\mathcal{S}$ -property and $\boldsymbol{\lambda} \in \Gamma$. Then, $\mathbf{x}^*$ is a global minimizer of the QP if and only if*

$$\nabla_{\mathbf{x}} f(\mathbf{x}^*) + \sum_{i \in \mathcal{N}} \lambda_i^* \nabla_{\mathbf{x}} g_i(\mathbf{x}^*) = 0, \quad (2.11a)$$

$$A_0 + \sum_{i \in \mathcal{N}} \lambda_i^* A_i \succeq 0, \quad \lambda_i^* g_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{N}. \quad (2.11b)$$

Theorem 1 states that if the $\mathcal{S}$ -property holds in the QP, then, the KKT conditions of the problem plus $A_0 + \sum_{i \in \mathcal{N}} \lambda_i^* A_i \succeq 0$ are sufficient and necessary to establish optimality of the solution. The next proposition explicitly relates the $\mathcal{S}$ -property with zero gap between the primal and dual problems:

Proposition 1. *Strong duality holds between QPI, QPE and problem (2.8), if and only if the $\mathcal{S}$ -property is satisfied.*

Sufficient conditions to satisfy the $\mathcal{S}$ -property will be presented in Theorem 3 and Theorem 5. We remark that the $\mathcal{S}$ -property is a constraint qualification. It has the interesting consequence that it only occurs whenever there is zero duality gap between the primal and dual problems of the QP. Therefore, there is an immediate equivalence between strong duality and the $\mathcal{S}$ -property.

Proof of Proposition 1. Suppose strong duality holds. Then, the KKT conditions derived from (2.6) and $A_0 + \sum_{i \in \mathcal{N}} \lambda_i^* A_i \succeq 0$ from problem (2.8) are satisfied for optimal points $\mathbf{x}^*$ and $\boldsymbol{\lambda}^*$. Because of Theorem 1 the $\mathcal{S}$ -property holds.

Suppose now that the $\mathcal{S}$ -property holds. Assume that QPI or QPE has a minimum value which we denote with γ :

$$\forall \mathbf{x} \in \mathbb{R}^p \mid \mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i \leq 0 \quad (\forall i \in \mathcal{N}) \implies \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 \geq \gamma \quad (2.12)$$

Because of the $\mathcal{S}$ -property, there exists a $\boldsymbol{\lambda} \in \Gamma$ such that $\bar{A}_0 - \gamma \bar{\mathbf{E}} + \sum_i \bar{\lambda}_i \bar{A}_i \succeq 0$ which makes problem (2.8) feasible. Because of this, strong duality holds. $\square$

2.3 Analysis of the $\mathcal{S}$ -property in QPs

Lemma 1 establishes necessary and sufficient conditions for the $\mathcal{S}$ -property to hold. In this section we derive sufficient conditions that guarantee that these requirements are satisfied.

The road map is as follows: we first analyze a system of strong alternatives in the context of semidefinite programming (SDP), whose variable Z can be a matrix of any rank. Then, we reformulate the system to an equivalent one with easier to analyze properties, and derive the conditions that guarantee the existence of a rank 1 solution in such system. Finally, we will relate the existence of a rank 1 solution with Lemma 1 and state our main result over the $\mathcal{S}$ -property. The importance of establishing a rank 1 solution resides in being able to recover the QP from the SDP formulation. These steps are summarized in figure 2.1.

Lemma 2. *Under Assumption 1, the following systems are strong alternatives for all $\alpha \in \mathbb{R}$:*

$$\exists Z \succeq 0 \mid \text{tr}[(\bar{A}_0 - \alpha \bar{\mathbf{E}})Z] < 0 \wedge \text{tr}[\bar{A}_i Z] \leq 0 \quad \forall i \in \mathcal{N}, \quad \iff \quad (2.13a)$$

$$\nexists \boldsymbol{\lambda} \in \Gamma \mid \bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{\mathbf{E}} \succeq 0. \quad (2.13b)$$

Fig. 2.1 Block diagram of equivalences to prove that the $\mathcal{S}$ -property holds.

Proof. See [appendix 2.A](#). □

We perform a transformation of the system in [Lemma 2](#) so that the matrices in the constraints become diagonal or *almost* diagonal. The reformulations are helpful because the new forms are easier to study. Before analyzing them, we introduce some definitions regarding the theory of matrix congruency:

Definition 3. Two matrices A and B over a field are called congruent if there exists an invertible matrix P over the same field such that

$$P^T A P = B. \quad (2.14)$$

Definition 4. A set of matrices $\{A_1, A_2, \dots, A_N\}$ is said to be simultaneously diagonalizable via congruence, if there exists a nonsingular matrix P such that $P^T A_i P$ is diagonal for every matrix A_i .

In general, finding sets of matrices that are simultaneously diagonalizable is a difficult task and it remains an open question [\[46\]](#). Nonetheless, we will study one simple case of simultaneously diagonalizable matrices. Finally, the following remark emphasizes an important property of congruent systems:

Remark. Sylvester's law of inertia ([Theorem 4.5.8 in \[48\]](#)): Two congruent matrices A and B have the same number of positive and negative eigenvalues. This implies that the definiteness of matrix A is preserved onto B .

Consider the following matrix P of dimension $(p+1) \times (p+1)$

$$P = \begin{pmatrix} P_1 & 0_{n_1 \times n_2} & \cdots & p_1 \\ 0_{n_2 \times n_1} & P_2 & \cdots & p_2 \\ \vdots & \ddots & \ddots & \vdots \\ 0_{n_N \times n_1} & \cdots & P_N & p_N \\ 0_{1 \times n_1} & 0_{1 \times n_2} & \cdots & 1 \end{pmatrix}, \quad (2.15)$$

where P_i has dimension $n_i \times n_i$ and $p_i \in \mathbb{R}^{n_i}$. We obtain the following congruent matrices:

$$F_i = P^T \bar{A}_i P = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & P_i^T \tilde{A}_i P_i & 0 & P_i^T (\tilde{A}_i p_i + \tilde{b}_i) \\ \vdots & \cdots & \ddots & 0 \\ 0 & (p_i^T \tilde{A}_i^T + \tilde{b}_i^T) P_i & 0 & p_i^T \tilde{A}_i p_i + 2\tilde{b}_i^T p_i + c_i \end{pmatrix}. \quad (2.16)$$

The first case assumes that $\tilde{b}_i \in \text{range}[\tilde{A}_i]$ for every $i \in \mathcal{N}$. Then, since every $\tilde{A}_i$ are real symmetric matrices, they allow a spectral decomposition of the form $\tilde{A}_i = Q_i \Delta_i Q_i^T$ where Q_i are orthogonal matrices, i.e., $Q_i Q_i^T = Q_i^T Q_i = I_{n_i}$, and Δ_i are diagonal. Choosing values $P_i = Q_i(I - \Delta_i \Delta_i^\dagger + \Delta_i^{\dagger/2})$ and $p_i = -\tilde{A}_i^\dagger \tilde{b}_i$ satisfy that every $F_i = P^T \bar{A}_i P$ for all $i \in \mathcal{N}$ are simultaneously diagonal. Moreover, this choice of values ensures that P is invertible. Finally, we note that $F_0 = P^T (\bar{A}_0 - \alpha \bar{E}) P$ is not diagonal in general.

The second case assumes that for some $i \in \mathcal{N}$, $\tilde{b}_i \notin \text{range}[\tilde{A}_i]$. In this case, F_i will not become diagonal since it is not possible that $P_i^T (\tilde{A}_i p_i + \tilde{b}_i) = 0$ for any P_i which is also full rank. Nonetheless, the previous choice of matrix P ensures that some components will be *almost* diagonalized, and some elements will remain in the last column and row. However, if for some j index $(F_i)_{p+1,j} = (F_i)_{j,p+1} \neq 0$, we necessarily have that $(F_i)_{jj} = 0$. This structure of F_i can still be exploited to obtain strong duality results.

We can reformulate [Lemma 2](#) to an equivalent system through the following lemma:

Lemma 3. *The following systems are equivalent:*

$$\exists Z \succeq 0 \mid \text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] < 0 \wedge \text{tr}[\bar{A}_i Z] \leq 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.17a)$$

$$\exists Y \succeq 0 \mid \text{tr}[F_0 Y] < 0 \wedge \text{tr}[F_i Y] \leq 0 \quad \forall i \in \mathcal{N}. \quad (2.17b)$$

Proof. We consider the change of variable $Y = P^{-1} Z P^{-T}$ and assume there exists some $Z \succeq 0$ satisfying (2.17a). Then,

$$\begin{aligned} \text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] &= \text{tr}[P^{-T} F_0 P^{-1} Z] = \text{tr}[F_0 P^{-1} Z P^{-T}] = \text{tr}[F_0 Y] < 0 \\ \text{tr}[\bar{A}_i Z] &= \text{tr}[P^{-T} F_i P^{-1} Z] = \text{tr}[F_i P^{-1} Z P^{-T}] = \text{tr}[F_i Y] \leq 0, \quad \forall i \in \mathcal{N}. \end{aligned}$$

Because of Sylvester's law of inertia, Y is a congruent matrix of Z and we have that $Y \succeq 0$. This implies that (2.17a) and (2.17b) are equivalent. $\square$

The next step is to find when there exists a $Y \succeq 0$ that satisfies (2.17b) and has rank 1. This is the most important step to prove strong duality in a QP, and will introduce several limitations to when this property can be guaranteed. We analyze two cases separately.

2.3.1 F_i simultaneously diagonalizable

This subsection assumes that $\tilde{b}_i \in \text{range}[\tilde{A}_i]$ and that we can find F_i diagonal matrix and congruent to $\tilde{A}_i$ for all $i \in \mathcal{N}$ simultaneously. We state the following theorem.

Theorem 2. Assume $\exists Y \succeq 0$ such that Y satisfies $\text{tr}[F_0 Y] < 0$ and a system of constraints given by $\text{tr}[F_i Y] = \psi_i$, where $\psi_i \in \mathbb{R}$ for all $i \in \mathcal{N}$. A sufficient condition for the previous system to have a rank 1 solution is that there exists a diagonal matrix D with entries ± 1 such that $DF_0 D$ is a Z -matrix.

Proof. See appendix 2.A. □

Because of Theorem 2, assume now there exists a matrix D such that $DF_0 D$ is a Z -matrix. This implies, that there exists a vector $y = (y_j)_{j=1}^{p+1}$ and $y_j = (D)_{jj} \sqrt{\sum_{r=1}^R v_{rj}^2}$ such that

$$y^T F_0 y \leq \text{tr}[F_0 Y] < 0 \quad (2.18a)$$

$$y^T F_i y = \text{tr}[F_i Y] = \psi_i, \quad \forall i \in \mathcal{N}. \quad (2.18b)$$

The rank 1 solution in (2.18) implies that the system of strong alternatives from Lemma 2 is satisfied because of the relation established in Lemma 3. We can now combine these results with the $\mathcal{S}$ -property and show that (2.18) is equivalent to Lemma 1. We complete this argument with the following two lemmas:

Lemma 4. The following systems are equivalent:

$$\exists y \in \mathbb{R}^{p+1} \mid y^T F_0 y < 0 \wedge y^T F_i y \leq 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.19a)$$

$$\exists \mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < \alpha \wedge g_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{N}. \quad (2.19b)$$

Proof. See appendix 2.A. □

Problem QPE requires an alternative lemma:

Lemma 5. The following systems are equivalent if $\tilde{A}_i \neq 0 \quad \forall i \in \mathcal{N}$:

$$\exists y \in \mathbb{R}^{p+1} \mid y^T F_0 y < 0 \wedge y^T F_i y = 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.20a)$$

$$\exists \mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < \alpha \wedge g_i(\mathbf{x}) = 0, \forall i \in \mathcal{N}. \quad (2.20b)$$

Proof. See [appendix 2.A](#). □

We now state our main result.

Theorem 3. *Given a QP in the form of [QPI](#) or [QPE](#), suppose [Assumption 1](#) is satisfied and that $\tilde{b}_i \in \text{range}[\tilde{A}_i]$ for every $i \in \mathcal{N}$. Furthermore, assume there exists a diagonal matrix D whose elements are ± 1 such that DF_0D is a Z -matrix. Then, the $\mathcal{S}$ -property holds.*

An immediate consequence of [Theorem 3](#) is that we now have sufficient conditions to apply [Theorem 1](#) and [Proposition 1](#). To conclude, a search algorithm that verifies whether such matrix D exists is proposed in [section 2.5.4](#).

Proof. [Lemma 2](#) introduces a system of strong alternatives between semidefinite programs. [Lemma 3](#) reformulates one of the systems as an equivalent problem with diagonal matrices. [Theorem 2](#) introduces a sufficient condition for the reformulated system to have a solution of rank 1. [Lemmas 4](#) and [5](#) show that if there exists a rank 1 solution in the system of strong alternatives, then the $\mathcal{S}$ -property defined in [Lemma 1](#) is satisfied. □

Remark. *We note that if DF_0D becomes diagonal, then the QP can be formulated as an equivalent linear program, as presented in [\[50\]](#).*

2.3.2 F_i not simultaneously diagonalizable

In this subsection we assume that for some $i \in \mathcal{N}$, $\tilde{b}_i \notin \text{range}[\tilde{A}_i]$. Denote with M the set of indexes such that for $m \in M$ we have $(F_i)_{p+1,m} = (F_i)_{m,p+1} \neq 0$ for any $i \in \mathcal{N}$. Furthermore, we necessarily have that $(F_i)_{mm} = 0$ for $m \in M$. We can invoke the following theorem for this case:

Theorem 4. *Suppose $\exists Y \succeq 0$ such that Y satisfies $\text{tr}[F_0Y] < 0$ and $\text{tr}[F_iY] = \psi_i$, where $\psi_i \in \mathbb{R}$ for all $i \in \mathcal{N}$. A sufficient condition for the previous system to have a rank 1 solution is that the diagonal elements of A_0 are nonnegative, $(A_0)_{mj} = (A_0)_{jm} = 0$ for any $m \neq j$ ($m \in M$), and there exists a diagonal matrix D whose terms satisfy*

$$(D)_{ii}(D)_{jj}(F_0)_{ij} \leq 0 \quad \forall i, j \notin M. \quad (2.21)$$

[Theorem 4](#) is more demanding than [Theorem 2](#) because of the lack of simultaneous diagonalization of the constraints. Nevertheless, both theorems state unique results regarding the existence of rank 1 solutions in SDP programs.

Proof. See [appendix 2.A](#). □

The final step is to relate the existence of $y \in \mathbb{R}^{p+1}$ such that

$$y^T F_0 y \leq \text{tr}[F_0 Y] < 0 \quad (2.22)$$

$$y^T F_i y = \text{tr}[F_i Y] \leq 0 \quad \forall i \in \mathcal{N}, \quad (2.23)$$

to the $\mathcal{S}$ -property reformulation described in [Lemma 1](#). We present the following result:

Lemma 6. *Assume that $A_0 \succeq 0$. Then, the following systems are equivalent:*

$$\exists y \in \mathbb{R}^{p+1} \mid y^T F_0 y < 0 \wedge y^T F_i y \leq 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.24a)$$

$$\exists \mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < \alpha \wedge g_i(\mathbf{x}) \leq 0, \forall i \in \mathcal{N}. \quad (2.24b)$$

Proof. See [appendix 2.A](#). □

The previous lemma directly relates the existence of rank 1 solution in the SDP program with the $\mathcal{S}$ -property. This allows us to summarize this result in the following theorem.

Theorem 5. *Given a QP in the form of [QPI](#) or [QPE](#), suppose [Assumption 1](#) is satisfied and that for some $i \in \mathcal{N}$, $\tilde{b}_i \notin \text{range}[\tilde{A}_i]$. Assume that $A_0 \succeq 0$, $(A_0)_{mj} = (A_0)_{jm} = 0$ for any $m \neq j$, $m \in M$, and that there exists a diagonal matrix D whose terms satisfy*

$$(D)_{ii} D_{jj} (F_0)_{ij} \leq 0 \quad \forall i, j \notin M. \quad (2.25)$$

Then, the $\mathcal{S}$ -property holds.

2.4 Existence of solutions

We introduce the following set of dual variables:

$$W = \{ \boldsymbol{\lambda} \in \Gamma \mid A_0 + \sum_i \lambda_i A_i \succeq 0 \}. \quad (2.26)$$

We have the following proposition:

Proposition 2. *Function $q(\boldsymbol{\lambda})$ is coercive if [Assumption 1](#) is satisfied.*

Proof. In order to prove the coercivity of the function, we need to show that if $\|\boldsymbol{\lambda}\| \rightarrow \infty$, then $q(\boldsymbol{\lambda}) \rightarrow -\infty$. Let's indicate $M = \{ i \in \mathcal{N} \mid \lambda_i \rightarrow \infty \}$ and $\overline{M} = \{ i \in \mathcal{N} \mid \lambda_i \nrightarrow \infty \}$. We require that M is nonempty.

We distinguish two cases, one where there $\exists i \in M | A_i \not\geq 0$, and another where $\forall i \in M, A_i \geq 0$. In the first case, $q(\boldsymbol{\lambda}) \rightarrow -\infty$ because $A_0 + \sum_i \lambda_i A_i \not\geq 0$ as $\lambda_i \rightarrow \infty$.

In the second case, we take $\lambda_i = \lambda$ for every $i \in M$ and we get

$$\begin{aligned}
\lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} q(\boldsymbol{\lambda}) &= \lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} \sum_i \lambda_i c_i - (b_0 + \sum_i \lambda_i b_i)^T (A_0 + \sum_i \lambda_i A_i)^\dagger (b_0 + \sum_i \lambda_i b_i) \\
&= \lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} \lambda \left(\frac{1}{\lambda} \sum_i \lambda_i c_i - \frac{1}{\lambda^2} (b_0 + \sum_i \lambda_i b_i)^T \left(\frac{1}{\lambda} A_0 + \frac{1}{\lambda} \sum_i \lambda_i A_i \right)^\dagger (b_0 + \sum_i \lambda_i b_i) \right) \\
&= \lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} \sum_{i \in M} \lambda \left(c_i - b_i^T \left(\frac{1}{\lambda} A_0 + \frac{1}{\lambda} \sum_{j \in \bar{M}} \lambda_j \tilde{A}_j + \sum_{j \in M} \tilde{A}_j \right)^\dagger b_i \right) \\
&= \lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} \sum_{i \in M} \lambda (c_i - b_i^T A_i^\dagger b_i). \tag{2.27}
\end{aligned}$$

Note that we used the relations

$$\left(\sum_{i \in M} A_i \right)^\dagger = \sum_{i \in M} A_i^\dagger \quad \text{and} \quad b_i A_i^\dagger \tilde{b}_j = \tilde{b}_j A_i^\dagger \tilde{b}_j = 0 \quad \forall i \neq j \tag{2.28}$$

because of the non-overlapping property of the A_i 's and b_i 's.

Considering the case in which $\tilde{A}_i \geq 0$, we can express $g_i(x_i)$ in the following form:

$$g_i(x_i) = \|(\tilde{A}_i)^{1/2} x_i + \tilde{b}_i^{\text{sq}}\|^2 - c_i^{\text{sq}} = x_i^T \tilde{A}_i x_i + 2\tilde{b}_i^T x_i + c_i, \tag{2.29}$$

where $\tilde{A}_i = (\tilde{A}_i)^{T/2} (\tilde{A}_i)^{1/2}$, $\tilde{b}_i^T = (\tilde{b}_i^{\text{sq}})^T (\tilde{A}_i)^{1/2}$, $c_i = (\tilde{b}_i^{\text{sq}})^T \tilde{b}_i^{\text{sq}} - c_i^{\text{sq}}$. Because of Slater's condition there exists an $\hat{x}_i$ such that $g_i(\hat{x}_i) < 0$ and we necessarily have $c_i^{\text{sq}} > 0$. Determining the value of (2.27) we get:

$$c_i - \tilde{b}_i^T \tilde{A}_i^\dagger \tilde{b}_i = (\tilde{b}_i^{\text{sq}})^T \tilde{b}_i^{\text{sq}} - c_i^{\text{sq}} - (\tilde{b}_i^{\text{sq}})^T (\tilde{A}_i)^{1/2} (\tilde{A}_i)^\dagger (\tilde{A}_i)^{1/2} (\tilde{b}_i^{\text{sq}}) = -c_i^{\text{sq}} < 0$$

and conclude that $\lim_{\|\boldsymbol{\lambda}\| \rightarrow \infty} q(\boldsymbol{\lambda}) = -\infty$. The case in which $\lambda_i \rightarrow -\infty$ is analyzed in a similar way and we omit it. $\square$

Corollary 1. Suppose *Assumption 1* is satisfied and there exists $\boldsymbol{\lambda}$ such that $A_0 + \sum_i \lambda_i A_i \succ 0$. Then, $q(\boldsymbol{\lambda})$ has a nonempty and compact solution set. Furthermore, *QPI* and *QPE* have a nonempty solution set.

Proof. Using the Weirstrass theorem [9, Prop.3.2.1], $q(\boldsymbol{\lambda})$ is coercive and has non-empty interior, which guarantees existence of a non-empty and compact solution set. Because of weak duality, *QPI* and *QPE* must have a solution. $\square$

2.5 Algorithms

2.5.1 Based on semidefinite programming

QPI and **QPE** can be rewritten as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \text{tr}[A_0 X] + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & \text{tr}[A_i X] + 2b_i^T \mathbf{x} + c_i \leq 0, \quad \forall i \in \mathcal{N}, \\ & X = \mathbf{x}\mathbf{x}^T. \end{aligned} \quad (2.30)$$

Relaxing the rank constraint on X to $X \succeq \mathbf{x}\mathbf{x}^T$, and formulating the constraint using the Schur complement (see A.5.5 in [15]), we get

$$\begin{aligned} \min_{\mathbf{x}} \quad & \text{tr}[A_0 X] + 2b_0^T \mathbf{x} + c_0 \\ \text{s.t.} \quad & \text{tr}[A_i X] + 2b_i^T \mathbf{x} + c_i \leq 0, \quad \forall i \in \mathcal{N}, \\ & \begin{pmatrix} X & \mathbf{x} \\ \mathbf{x}^T & 1 \end{pmatrix} \succeq 0, \end{aligned} \quad (2.31)$$

which is a convex problem that can be solved using a semidefinite program solver. Problem [equation \(2.31\)](#) yields an optimal solution of problems **QPI** or **QPE**, if conditions of [Theorem 3](#) are satisfied.

Likewise, solving [\(2.8\)](#) as the dual formulation also achieves an optimal solution. The primal values are then recovered determining

$$\min_{\mathbf{x}} \quad \mathbf{x}^T (A_0 + \sum_i \lambda_i^* A_i) \mathbf{x} + (b_0 + \sum_i \lambda_i^* b_i)^T \mathbf{x}, \quad (2.32)$$

where $\boldsymbol{\lambda}^*$ corresponds to the solution of [equation \(2.8\)](#). In particular, the solution belongs to the following subspace

$$\mathbf{x}^* \in -\left(A_0 + \sum_i \lambda_i^* A_i\right)^\dagger \left(b_0 + \sum_i \lambda_i^* b_i\right)^T + \text{null} \left[A_0 + \sum_i \lambda_i^* A_i\right], \quad (2.33)$$

which has to be feasible and satisfy the complementarity conditions:

$$\lambda_i^* g_i(\mathbf{x}_i^*) = 0 \quad \forall i \in \mathcal{N}. \quad (2.34)$$

Note that if $A_0 + \sum_i \lambda_i^* A_i \succ 0$ the solution is unique.

2.5.2 Projected dual ascent algorithm

In this part we introduce some new notation, and we denote $\mathbf{g}(\mathbf{x}) = (g_i(\mathbf{x}))_{i \in \mathcal{N}}$ as a column operator that collects all of the constraints. The ascent algorithm is based on the following problem

$$\max_{\boldsymbol{\lambda}} \quad \boldsymbol{\lambda}^T \mathbf{g}(\tilde{\mathbf{x}}(\boldsymbol{\lambda}^k, \mathbf{x}^k)) \quad (2.35)$$

$$\text{s.t.} \quad \boldsymbol{\lambda} \in W \quad (2.36)$$

where W is defined in (2.26) and $\tilde{\mathbf{x}}(\boldsymbol{\lambda}^k, \mathbf{x}^k)$ is an operator defined as:

$$\tilde{\mathbf{x}}(\boldsymbol{\lambda}, \mathbf{x}^k) = \arg \min_{\mathbf{x} \in \mathbb{R}^p} L(\mathbf{x}, \boldsymbol{\lambda}) \quad (2.37)$$

The projected dual ascent method is described in [algorithm 3](#). Convergence analysis and performance corresponds to that of subgradient methods, see [16]. In our formulation we introduced Π_W as a projection operator onto set W and α^k as the step size at iteration k . Notice also that the problem in (2.37) is strongly convex.

Algorithm 3 Projected dual subgradient method

- 1: Initialize $\mathbf{x}^0$ and $\boldsymbol{\lambda}^0$. Set $k \leftarrow 0$.
 - 2: **while** $\boldsymbol{\lambda}^k$ is not a stationary point **do**
 - 3: $\mathbf{x}^{k+1} = \tilde{\mathbf{x}}(\boldsymbol{\lambda}^k, \mathbf{x}^k)$.
 - 4: $\boldsymbol{\lambda}^{\text{temp}} = \boldsymbol{\lambda}^k + \alpha^k \mathbf{g}(\mathbf{x}^{k+1})$.
 - 5: $\boldsymbol{\lambda}^{k+1} = \Pi_W[\boldsymbol{\lambda}^{\text{temp}}]$.
 - 6: Set $k \leftarrow k + 1$
 - 7: **end while**
-

The projection onto W can be done using an L2 norm:

$$\Pi_W(\boldsymbol{\lambda}^{\text{temp}}) = \arg \min_{\boldsymbol{\lambda} \in W} \sum_{i \in \mathcal{N}} \|\lambda_i - \lambda_i^{\text{temp}}\|^2. \quad (2.38)$$

This results into a convex problem formed by a quadratic function and semidefinite constraints. Alternatively, we can obtain a standard SDP program if we choose a distance metric like the L1 norm. The new form has an easier structure for semidefinite programming optimizers and can be solved more efficiently. We have:

$$\Pi_W(\boldsymbol{\lambda}^{\text{temp}}) = \arg \min_{\boldsymbol{\lambda} \in W} \sum_{i \in \mathcal{N}} |\lambda_i - \lambda_i^{\text{temp}}| \quad (2.39)$$

Because the separability of the constraints, we can establish the following conditions:

1. If $\tilde{A}_i \succeq 0$, then we have that $\lambda_i \geq \lambda_i^{\text{temp}}$.
2. If $\tilde{A}_i \preceq 0$, then we have that $\lambda_i \leq \lambda_i^{\text{temp}}$.
3. If $\tilde{A}_i$ is indefinite, then $\lambda_i \leq \lambda_i^{\text{temp}}$ if $\lambda_i^{\text{temp}} > 0$ and $\lambda_i \geq \lambda_i^{\text{temp}}$ otherwise.

If we introduce these new requirements as constraints in (2.39) and omit the absolute values in the objective, we obtain an equivalent SDP problem of dimension N , which can be solved efficiently.

Augmented Lagrangian

In order to infuse stability to dynamical systems derived from subgradient methods, it is frequent to use an augmented Lagrangian formulation [14]. By doing so, the augmented Lagrangian becomes differentiable (if the problem becomes strictly convex) and the subgradient becomes a gradient at every iteration point. In such case, problem (2.37) is substituted by:

$$\tilde{\mathbf{x}}(\boldsymbol{\lambda}, \mathbf{x}^k) = \arg \min_{\mathbf{x} \in \mathbb{R}^p} L(\mathbf{x}, \boldsymbol{\lambda}) + \sum_{i \in \mathcal{N}} \rho c(g_i(x_i)). \quad (2.40)$$

where $c(\cdot)$ corresponds to a strictly convex function that does not alter the cost of the problem at the optimal solution. Assuming that $g_i(x_i)$ is part of an equality constraint, $c(\cdot) = \|\cdot\|^2$ suffices to enforce stability to the subgradient method. Alternatively, if $g_i(x_i)$ is part of an inequality constraint, then

$$c(x) = \begin{cases} \|x\|^2 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.41)$$

achieves the same purpose.

It is easy to verify that the squared term added to problem (2.37) does not alter the solution points nor the optimality requirements of the original quadratic problem.

2.5.3 Distributed formulations

Centralized methods based on SDP such as equation (2.31) and (2.8), or dual ascent methods such as section 2.5.2, are not suitable for large problems with many variables. For this reason, distributed and parallel methods that can handle problems with large number of variables, or distributed datasets, are of high interest. To the best of our knowledge, there exists no framework for nonconvex QPs that use a distributed or parallel scheme. In order to propose one, we extend the setting of section 2.5.2 to include parallel updates using the FLEXA decomposition theory presented in [70]. Since the solution to the QP still relies on

(sub)gradient methods, which may not be very efficient, we also present a second formulation that solves the QP and the restrictions over the dual variables jointly, using the NOVA framework from [69].

Dual ascent method with FLEXA decomposition

The first idea consists of a two-loop procedure that emerges from the dual ascent method presented in section 2.5.2. Specifically, the joint optimization update (2.37) is decoupled and solved iteratively using the FLEXA framework from [70], while the dual variables are updated in an outer loop. The framework approximates at every step the original problem with a convex surrogate that can be decomposed in parallel problems and solved efficiently. The approximation is formed fixing the non-local group of variables to a point from a previous iteration, and optimizing only in the local variables. If this procedure is repeated and the conditions of Theorem 3 in [70] are fulfilled, then the iterates will converge to the global minimum point of the Lagrangian (2.37). Note that the Lagrangian is convex on $\mathbf{x}$, even though it is non-separable, and that is the reason why FLEXA converges to the global minimum.

The surrogate problem can be derived as follows:

$$U(\mathbf{x}, \boldsymbol{\lambda}; \mathbf{x}^{k,q}, \mathbf{x}^k) = \sum_{i \in \mathcal{N}} U_i(x_i; \mathbf{x}^{k,q}) + \rho_i \|x_i - x_i^k\|^2, \quad (2.42)$$

where

$$U_i(x_i; \mathbf{x}^{k,q}) = x_i^T \tilde{A}_0^{ii} x_i + 2 \sum_{j \neq i} (x_j^{k,q})^T \tilde{A}_0^{ij} x_i + 2 \tilde{b}_0^T x_i + \lambda_i (x_i^T \tilde{A}_i x_i + 2 \tilde{b}_i^T x_i + c_i), \quad (2.43)$$

$\mathbf{x}^{k,q} = (x_j^{k,q})_{j \in \mathcal{N}}$ is a global vector from a previous inner iteration q , x_i^k is a pivotal point in the outer loop k , and

$$A_0 = \begin{pmatrix} A_0^{11} & \cdots & A_0^{1N} \\ \vdots & \ddots & \vdots \\ A_0^{N1} & \cdots & A_0^{NN} \end{pmatrix} \quad (2.44)$$

is obtained from the quadratic objective function. Parameter $\rho_i \geq 0$ can be set to zero if $U(\mathbf{x}, \boldsymbol{\lambda}; \mathbf{x}^{k,q}, \mathbf{x}^k)$ is strongly convex in x_i .

Note that problem (2.42) is separable and convex in x_i . It is a quadratic problem with no constraints, whose solution can be computed as follows, see Appendix A in [15].

$$\begin{aligned} \min_{x_i \in \mathbb{R}^{n_i}} \quad & U_i(x_i; \mathbf{x}^k) + \rho_i \|x_i - x_i^k\|^2 \\ = \quad & \begin{cases} -r(\mathbf{x}^k)^T X_i(\lambda_i) r(\mathbf{x}^k) + c_0 + \lambda_i c_i & \text{if } X_i(\lambda_i) \succeq 0 \wedge r(\mathbf{x}^k) \in \text{range}[X_i(\lambda_i)] \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \quad (2.45)$$

where $X_i(\lambda_i) = \tilde{A}_0^{ii} + \lambda_i \tilde{A}_i + \rho_i \mathbf{I}_{n_i}$ and $r(\mathbf{x}^{k,q}, x_i^k) = \tilde{b}_{0i} + \lambda_i \tilde{b}_i + \sum_{j \neq i} (\tilde{A}_0^{ij})^T x_j^{k,q} + \rho_i x_i^k$. The solution comes as

$$\tilde{x}_i(\lambda_i; \mathbf{x}^{k,q}, x_i^k) \in \arg \min_{x_i \in \mathbb{R}^{n_i}} U_i(x_i; \mathbf{x}^k) + \rho_i \|x_i - x_i^k\|^2 = -X(\lambda_i)^\dagger r(\mathbf{x}^k) + \text{null}[X(\lambda_i)]. \quad (2.46)$$

If the quadratic problem is strongly convex then the solution is unique and $\tilde{x}_i(\lambda_i; \mathbf{x}^{k,q}, x_i^k) = -X(\lambda_i)^{-1} r(\mathbf{x}^k)$. This occurs automatically if $\rho_i > 0$. All steps are described in [algorithm 4](#).

Algorithm 4 Dual gradient ascent method with distributed FLEXA decomposition.

- 1: Initialize $\mathbf{x}^0$ and $\boldsymbol{\lambda}^0$. Set $k \leftarrow 0$.
 - 2: **while** $\boldsymbol{\lambda}^k$ is not a stationary point **do**
 - 3: Set $q \leftarrow 0$.
 - 4: **while** $\mathbf{x}^{k,q}$ is not a stationary point **do**
 - 5: Compute $\tilde{x}_i(\lambda_i, \mathbf{x}^{k,q}, x_i^k)$ for all $i \in \mathcal{N}$ **in parallel**.
 - 6: Update $x_i^{k,q+1} = x_i^{k,q} + \alpha_q (\tilde{x}_i(\lambda_i^k; \mathbf{x}^{k,q}, x_i^k) - x_i^{k,q})$ for all $i \in \mathcal{N}$.
 - 7: Set $q \leftarrow q + 1$.
 - 8: **end while**
 - 9: Set $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^{k,q}$.
 - 10: $\boldsymbol{\lambda}^{\text{temp}} = \boldsymbol{\lambda} + \mu^k \mathbf{g}(\mathbf{x}^{k+1})$.
 - 11: $\boldsymbol{\lambda}^{k+1} = \Pi_W[\boldsymbol{\lambda}^{\text{temp}}]$.
 - 12: Set $k \leftarrow k + 1$
 - 13: **end while**
-

Convergence of [algorithm 4](#) is guaranteed if conditions of Theorem 3 in [70] are fulfilled and μ^k are chosen small enough so that the (sub)gradient method converges. Combining the convergence results from [70] and diminishing step sizes from [16] for subgradient methods, we can establish the following theorem:

Theorem 6. *Given problem [QPI](#) or [QPE](#). Assume that the $\mathcal{S}$ -property holds. Suppose*

$$0 < \inf_q \alpha_q \leq \sup_q \alpha_q \leq \alpha_{\max} \leq 1, \quad \rho_i > 0 \quad \forall i \in \mathcal{N}$$

and μ^k is small enough. Then, *algorithm 4* converges to an optimal solution of the QP.

An extension of this decomposition using the augmented Lagrangian formulation from *section 2.5.2* is possible, although in this case problem (2.45) is no longer quadratic and must be solved using other convex optimization tools.

Distributed method based on NOVA decomposition

The previous subsection presented a distributed method with guarantees to a global minimum point using a two loop iterative procedure. It is based on first order methods, which are simple, but may be slow to converge. In this section we focus also on a two loop procedure with better convergence properties. We still assume that the $\mathcal{S}$ -property holds in the QP.

We introduce the following complementarity problem

$$0 \preceq Z \perp A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i \succeq 0, \quad (2.47)$$

where in this case, $A \perp B$ denotes the inner matrix product $\text{tr}[AB] = 0$, where $Z \in \mathcal{S}^p$ is a dual variable. Problem (2.47) is a requirement for optimality as indicated by *Theorem 1*.

We can reformulate (2.47) as a minimization problem:

$$\min_Z \quad \text{tr} \left[Z(A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i) \right], \quad (2.48)$$

which has minimum value zero. From (2.48), we formulate a gradient descent algorithm

$$Z^{k+1} = \left[Z^k - \mu^k (A_0 + \sum_{i \in \mathcal{N}} \lambda_i^k A_i) \right]_+, \quad (2.49)$$

where λ_i^k corresponds to the result of a coupled quadratic problem, μ^k is a step size, and $[Z]_+$ denotes the projection onto the positive semidefinite cone of symmetric matrices.

We can form a Lagrangian that includes all constraints:

$$L(\mathbf{x}, \boldsymbol{\lambda}, Z) = \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 + \sum_{i \in \mathcal{N}} \lambda_i \left(\mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i \right) + \text{tr} \left[Z(A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i) \right]. \quad (2.50)$$

And from it, we can deduce the following complementarity problems:

$$\Gamma_i \ni \lambda_i \perp \mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i + \text{tr}[Z A_i] \quad \forall i \in \mathcal{N}, \quad (2.51)$$

altogether with complementarity problem (2.47). Term $\text{tr}[Z A_i] \neq 0$ prevents values of $\lambda_i \notin W$, enforcing optimality. Note that if $\text{tr}[Z A_i] = 0$, the constraint is unaffected.

Problem (2.51) is coupled in $\mathbf{x}$, but we can use the NOVA decomposition framework presented in [69] to find a stationary solution using a parallel update scheme. The procedure is similar to that of the FLEXA decomposition: we propose a convex surrogate problem that can be decoupled and solved in parallel. Because NOVA only guarantees convergence to a stationary solution, our nonconvex QP given by (2.50) may fail to converge to the global solution. This gives rise to a two loop algorithm: in the outer loop we update Z^k to enforce that $\lambda \in W$, and in the inner loop the procedure iterates over the coupled problem using the NOVA framework. If the stationary solution does not satisfy the optimality conditions, we update Z and repeat the procedure. Eventually, Z will enforce that $\lambda \in W$ and the stationary solution from NOVA will be optimal.

The centralized surrogate problem is presented as

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^p} \quad & \sum_{i \in \mathcal{N}} U_i(x_i, \mathbf{x}^q) + \rho_i \|x_i - x_i^q\|^2 + \lambda_i^T (g_i(x_i) + \text{tr}[Z^k A_i]) \\ \text{s.t.} \quad & \Gamma_i \ni \lambda_i \perp g_i(x_i) + \text{tr}[Z^k A_i] \quad \forall i \in \mathcal{N}, \end{aligned} \quad (2.52)$$

where we added a quadratic term to enforce strong convexity in the local iteration and retained the complementarity condition (2.51) as a local constraint. Index q refers to an inner loop iteration number and k to an outer loop. In this case

$$U_i(x_i; \mathbf{x}^q) = x_i^T \tilde{A}_0^{ii} x_i + 2 \sum_{j \neq i} (x_j^q)^T \tilde{A}_0^{ij} x_i + 2 \tilde{b}_{0i}^T x_i \quad (2.53)$$

and $\mathbf{x}^q = (x_j^q)_{j \in \mathcal{N}}$. Problem (2.52) is decomposed into N subproblems

$$\begin{aligned} \min_{x_i \in \mathbb{R}^{n_i}} \quad & U_i(x_i, \mathbf{x}^q) + \rho_i \|x_i - x_i^q\|^2 + \lambda_i^T (g_i(x_i) + \text{tr}[Z^k A_i]) \\ \text{s.t.} \quad & \Gamma_i \ni \lambda_i \perp g_i(x_i) + \text{tr}[Z^k A_i], \end{aligned} \quad (2.54)$$

where each subproblem is a nonconvex QP with a single complementarity constraint, which is also quadratic. These problems can be solved efficiently for any $\rho_i \geq 0$ as we have already discussed in chapter 1, see Appendix B in [15]. The dual function becomes

$$\Upsilon_i(\lambda_i, \mathbf{x}^q, Z^k) = \begin{cases} -(r_i + \lambda_i s_i)^T (\tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i)^\dagger (r_i + \lambda_i s_i) + \lambda_i c_i & \text{if } \tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i \succeq 0 \text{ and } (r_i + \lambda_i s_i) \in \text{range}(\tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i) \\ -\infty & \text{otherwise} \end{cases} \quad (2.55)$$

where $r_i = \tilde{b}_{0i} + \sum_{j \neq i} (\tilde{A}_0^{ij})^T x_j^q + \rho_i x_i^q$ and $s_i = \tilde{b}_i + \text{tr}[Z^k A_i]$.

The maximum of $\Upsilon_i(\lambda_i, \mathbf{x}^q, Z^k)$ characterizes the solution of (2.54) for given strategies $\mathbf{x}^k$ and Z^k . The previous formulation allows us to establish lower and upper bounds on λ_i ,

namely

$$\begin{aligned} \{\lambda_i^{\inf} | \lambda_i^{\sup}\} &= \arg\{\inf | \sup\} \lambda_i \\ \text{s.t. } \tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i &\succeq 0, \end{aligned} \quad (2.56)$$

which defines the convex region

$$\Lambda_i = \{\lambda_i \in \Gamma_i | \lambda_i^{\inf} \leq \lambda_i \leq \lambda_i^{\sup}\}.$$

Note that the values at the boundary may not satisfy $\tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i \succeq 0$ and should be excluded accordingly. We can now present the following proposition.

Proposition 3. Assume that $g_i(x_i) + \text{tr}[ZA_i]$ satisfies [Assumption 1](#). Then, $\Upsilon_i(\lambda_i; \mathbf{x}^q, Z^k)$ is coercive, it has a compact solution set, and (2.42) has unique solution for some $\rho_i > 0$.

Proof. See [appendix 2.A](#). □

If Slater condition is not satisfied in [Proposition 3](#), then the problem is unfeasible and $\lambda_i \rightarrow \pm\infty$ depending on the structure of the QP. Equation (2.49) should be updated having in mind that the projection will solve the discontinuity. Step size μ^k might be too large.

The calculation of $\Upsilon_i(\lambda_i; \mathbf{x}^q, Z^k)$ can be done solving the semidefinite program

$$\begin{aligned} \max_{\gamma_i, \lambda_i} \quad & \gamma_i \\ \text{s.t.} \quad & \begin{pmatrix} \tilde{A}_0^{ii} + \lambda_i \tilde{A}_i + \rho \mathbf{I}_{n_i} & r(\mathbf{x}^k) + \lambda_i s_i \\ r(\mathbf{x}^k)^T + \lambda_i s_i^T & \lambda_i c_i - \gamma_i \end{pmatrix} \succeq 0 \\ & \gamma_i \in \mathbb{R}, \lambda_i \in \Lambda_i \end{aligned} \quad (2.57)$$

or, more efficiently, using a bisection algorithm on the constraint: find $\lambda_i \in \Lambda_i$ such that $\lambda_i(g_i(x_i) + \text{tr}[Z^k A_i]) = 0$, and

$$\tilde{x}_i(\lambda_i; \mathbf{x}^q, Z^k) = -(\tilde{A}_0^{ii} + \lambda_i \tilde{A}_i + \rho \mathbf{I}_{n_i})^{-1}(r + \lambda_i s_i). \quad (2.58)$$

Finally, existence of a fixed point is guaranteed if [Assumption 1](#) is satisfied. [Algorithm 5](#) indicates the specific steps to find a stationary solution. Note that step 7 is performed in parallel, whereas step 8 requires communication between the optimization nodes. Convergence of the algorithm is detailed in Theorem 2 in [69].

2.5.4 Verification of sufficient conditions to satisfy the $\mathcal{S}$ -property

Finding a matrix D that satisfies the sufficient conditions enunciated in [Theorem 3](#) can be automated with a simple algorithm. We propose an efficient search that finds matrix D if it

Algorithm 5 Optimal distributed algorithm for QPs using NOVA decomposition (see [69]).

- 1: Choose step size $\mu^k > 0$, and $\alpha^k \in (0, 1]$. Set $\mathbf{x}^0$ feasible in QP. Set $k \leftarrow 1$.
 - 2: Initialize $Z^0 = 0$ and $\lambda_i^{0,0} = 0$ for all i .
 - 3: **while** Z^k is not a stationary point **do**
 - 4: Initialize $\lambda_i^{k,0} \leftarrow \lambda_i^{k-1,q}$. Set $q \leftarrow 0$.
 - 5: **while** $\lambda^{k,q}$ is not a stationary point **do**
 - 6: Set $\lambda_i^{k,q+1} \leftarrow \max_{\lambda_i} \Upsilon_i(\lambda_i, \mathbf{x}^q, Z^k)$.
 - 7: Compute $x_i^{q+1} = x_i^q + \alpha^k(\tilde{x}_i(\lambda_i^{k,q+1}) - x_i^q)$ for all $i \in \mathcal{N}$.
 - 8: Set $\mathbf{x}^{q+1} = (x_i^{q+1})_{i \in \mathcal{N}}$. Communicate $\mathbf{x}^{q+1}$ to neighbors.
 - 9: Set $q \leftarrow q + 1$.
 - 10: **end while**
 - 11: Update $Z^{k+1} = [Z^k + \mu^k(A_0 + \sum_{i \in \mathcal{N}} \lambda_i^{k,q} A_i)]_+$.
 - 12: Set $k \leftarrow k + 1$.
 - 13: **end while**
-

exists, and returns a failure if it does not exist. The steps are described in [algorithm 6](#). To simplify the description, we introduced the logical operator $\text{vfnp}[DF_0D]$ to indicate when the off-diagonal elements of DF_0D are all nonpositive: it returns `true` if the condition is satisfied and `false` otherwise.

[Algorithm 6](#) is based on a search to find matrix D such that it fulfills the requirements. The algorithm can exit in two ways: it finds the matrix D that satisfies the constraints, or verifies the impossibility to satisfy them. From steps [4](#) to [18](#) and [5](#) to [15](#), the algorithm iterates over the upper triangular elements of F_0 evaluating every column in sequential order. On step [6](#) it identifies the first positive element in column j and sets $(D)_{jj} \leftarrow -1$ if it was not already set. Then, from step [7](#) to [12](#) it explores the rest of the column j for other positive elements and adjusts the corresponding values of D accordingly. If at any point the algorithm finds that there is a positive element in $(DF_0D)_{ij}$ and, $(D)_{jj}$ and $(D)_{ii}$ have already been set to -1, then, a matrix D that satisfies the requirements cannot exist. These conditions are verified on steps [9-10](#) and [13](#). Finally, at the end of every column iterate j , the algorithm checks if DF_0D satisfies the requirements to have a prompt exit.

2.6 Simulations

In this section we verify with simulations that problems satisfying [Theorem 3](#) indeed present the strong duality property and illustrate the performance of [algorithm 5](#). The proposed

Algorithm 6 Check if conditions from [Theorem 3](#) are satisfied for given F_0

```

1: Initialize  $D = I_{p+1}$ .
2: if  $\text{vfnp}[DF_0D]$  then return  $D$ .
3: end if
4: for  $j=1:p+1$  do
5:   for  $i=1:j-1$  do
6:     if  $(DF_0D)_{i,j} > 0$  and  $(D)_{jj} == 1$  then set  $(D)_{jj} \leftarrow -1$ .
7:     for  $r=i+1:p+1$  and  $r \neq j$  do
8:       if  $(DF_0D)_{i,j} > 0$  and  $(D)_{rr} == 1$  then set  $(D)_{rr} \leftarrow -1$ .
9:       else if  $(DF_0D)_{i,j} > 0$  and  $(D)_{rr} == -1$  then
10:        return  $D$  does not exist.
11:       end if
12:     end for
13:     else if  $(F_0)_{i,j} > 0$  and  $(D)_{jj} = -1$  then return  $D$  does not exist.
14:     end if
15:   end for
16:   if  $\text{vfnp}[DF_0D]$  then return  $D$ .
17:   end if
18: end for

```

problem to be solved is a nonconvex partitioning QP with the following form:

$$\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} \\
\text{s.t.} \quad & x_i^2 = 1 \quad \forall i \in \{1, \dots, p\},
\end{aligned} \tag{2.59}$$

where A_0 varies from being positive, negative semidefinite, or an indefinite matrix. Problem (2.59) is an instance of [QPE](#) which makes all problems nonconvex.

We present our results in [section 2.6](#). We averaged the different curves over 100 simulations and specified $p = 150$. The optimal red curves were obtained solving instances of [equation \(2.31\)](#), whereas the individual curves were obtained using [algorithm 5](#). Both methods yield the same result, satisfy the constraints and verify the optimality criteria, as expected. We used $\rho_i = 0.1$ in all cases.

Matrix A_0 and b_0 were simulated as follows: values were randomly chosen with a Gaussian distribution, and then the matrix was made symmetric. Then, we constructed a temporary matrix F_{temp1} with the previously drawn values. We also sampled a random matrix D whose elements were ± 1 . We forced that the off-diagonal elements of $F_{\text{temp2}} = DF_{\text{temp1}}D$ be nonpositive and recovered $F_0 = DF_{\text{temp2}}D$ to finally characterize (2.59). The diagonal elements of F_0 were then altered to force that A_0 was positive or negative semidefinite, or remained indefinite.

Fig. 2.2 Convergence speed of **algorithm 5** to the optimal cost value.

We simulated our results in MATLAB R2016a using an Intel Core i5 M560 @2.67GHz. The average simulation times of **equation (2.31)** using CVX [42] were around 18 seconds, whereas the simulation times of **algorithm 5** were around 0.18 seconds.

2.6.1 Robust least squares

Least squares (LS) is a popular method that finds linear dependencies between output data y and a list of input observations given by A and b , such that, in the absence of noise, $Ax - b = y$. We assumed $x \in \mathbb{R}^p$, A is matrix of dimensions $N \times p$ and $b \in \mathbb{R}^N$. In the presence of noise, a squared error measure is proposed and minimized, such that $\|Ax - b\|^2$ has the smallest value possible. The total least squares (TLS) problem aims to minimize such error, allowing changes in matrix A and vector b , such that $(A + \Delta A)x - (b + \Delta b) = 0$ with minimal energy of $\|(\Delta A, \Delta b)\|_{\mathcal{F}}^2$.

Both problem are widely used in the literature, but they present high sensitivity to perturbations in the input parameters. For this reason, regularization alternatives have been proposed that aim to minimize the output variance under noisy input parameters. An alternative to such methods proposed in [31] consists of solving a robust least squares (RLS) version as follows:

$$\begin{aligned} \min_{x \in \mathbb{R}^p} \max_{(\Delta A, \Delta b)} \quad & \|(A + \Delta A)x - (b + \Delta b)\|^2 \\ \text{s.t.} \quad & \|(\Delta A, \Delta b)\|^2 \leq \rho, \end{aligned} \quad (2.60)$$

where the output x of such problem corresponds to the robust solution under deterministic (but finite) perturbations on values A , b .

The reason behind different error values may be due to a distributed source of data, such as in [21]. Our RLS proposal becomes:

$$\begin{aligned} \min_{x \in \mathbb{R}^p} \max_{(\Delta A, \Delta b)} \quad & \| (A + \Delta A)x - (b + \Delta b) \|^2 \\ \text{s.t.} \quad & \| (\Delta A)_{:i} \|^2 = \rho_i \quad \forall i \in \{1, \dots, p\} \\ & \| \Delta b \|^2 \leq \rho_{p+1}, \end{aligned} \quad (2.61)$$

when grouping errors in columns. We denoted the i th column of matrix ΔA with $(\Delta A)_{:i}$.

We remark that the methods proposed in [31] are only valid for a single quadratic constraint, but we can use the analysis we developed to solve the problem with multiple constraints as long as the $\mathcal{S}$ -property is satisfied.

Centralized approach

Our algorithmic proposal consists of using a gradient descent method on variable x , while solving the maximization problem either in a centralized manner or using the decentralized methods from [section 2.5.3](#).

The minimization step on variable $\bar{x}$ takes the following form

$$x^{k+1} = x^k + \alpha_x^k (A + \Delta A)^T ((A + \Delta A)x^k - (b + \Delta b)) \quad (2.62)$$

where α_x^k corresponds to the gradient's step size and k denotes the iteration step.

Consider the following renaming of variables: $\Delta = (\Delta A, \Delta b)$, $H = (A, b)$ and $\bar{x} = (x^T, -1)^T$. The maximization problem can be written as

$$\begin{aligned} \max_{\Delta} \quad & \| (H + \Delta)\bar{x} \|^2 \\ \text{s.t.} \quad & \| (\Delta)_{:i} \|^2 = \rho_i \quad \forall i \in \{1, \dots, p+1\}. \end{aligned} \quad (2.63)$$

By further denoting $U = \Delta^T \Delta$ and some transformations, an equivalent formulation becomes:

$$\begin{aligned} \max_{U, \Delta} \quad & \text{tr}[U\bar{x}\bar{x}^T] + \text{tr}[(H^T \Delta + \Delta^T H)\bar{x}\bar{x}^T] + \text{tr}[H^T H\bar{x}\bar{x}^T] \\ \text{s.t.} \quad & U_{ii} = \rho_i \quad \forall i \in \{1, \dots, p+1\} \\ & U = \Delta^T \Delta. \end{aligned} \quad (2.64)$$

Easily, the SDP relaxation is obtained as follows when $U \succeq \Delta^T \Delta$, i.e.,

$$\begin{aligned} \max_{U, \Delta} \quad & \text{tr}[U \bar{x} \bar{x}^T] + \text{tr}[(H^T \Delta + \Delta^T H) \bar{x} \bar{x}^T] + \text{tr}[H^T H \bar{x} \bar{x}^T] \\ \text{s.t.} \quad & U_{ii} = \rho_i \quad \forall i \in \{1, \dots, p+1\} \\ & \begin{pmatrix} U & \Delta^T \\ \Delta & I_N \end{pmatrix} \succeq 0. \end{aligned} \quad (2.65)$$

The centralized proposal based on solving problem (2.65) is described in [algorithm 7](#).

Algorithm 7 Centralized RLS algorithm

- 1: Initialize x^0, Δ^0 and $\{\alpha^k\}$.
 - 2: **while** STOP criteria is not satisfied on x^k **do**
 - 3: Solve problem (2.65). Assign result to $M^k = (\Delta A^k, \Delta b^k)$.
 - 4: Set $x^{k+1} \leftarrow x^k + \alpha_x^k 2(A + \Delta A^k)^T ((A + \Delta A^k)x^k - (b + \Delta b^k))$.
 - 5: Set $k \leftarrow k + 1$.
 - 6: **end while**
-

Distributed approach

We consider a distributed scheme based on the NOVA decomposition we developed in [section 2.5.3](#) to solve the maximization problem. The constraints are all separable, and the objective function is coupled. Provided that the $\mathcal{S}$ -property holds, we need to verify that

$$0 \preceq Z \perp (-\bar{x} \bar{x}^T + \text{diag}[\lambda_1, \dots, \lambda_{p+1}]) \succeq 0. \quad (2.66)$$

The previous result is derived from the first order conditions of (2.61).

The local problem we aim to solve according to [section 2.5.3](#) is as follows:

$$\begin{aligned} \min_{(M)_{:,i} \in \mathbb{R}^{p+1}} \quad & \|H \bar{x}^k + (\bar{x}^k)_i (M)_{:,i} + \sum_{j \neq i} (\bar{x}^k)_j (M^r)_{:,j}\|^2 \\ \text{s.t.} \quad & \|(M)_{:,i}\|^2 + (Z)_{ii} = \rho_i, \end{aligned} \quad (2.67)$$

where $\bar{x}^k$ corresponds to an inner value of $\bar{x}$ at iteration k and M^r to an intermediate estimate at inner iteration r . Problem (2.67) has the following Lagrangian:

$$L_i((M)_{:,i}, \lambda_i) = \|H \bar{x}^k + (\bar{x}^k)_i (M)_{:,i} + \sum_{j \neq i} (\bar{x}^k)_j (M^r)_{:,j}\|^2 + \lambda_i (\|(M)_{:,i}\|^2 + (Z)_{ii} - \rho_i). \quad (2.68)$$

We can obtain an analytical solution from the first order condition:

$$\frac{\partial L_i}{\partial (M)_{:i}} = -(\bar{x})_i(H\bar{x}^k + (\bar{x})_i(M)_{:i} + \sum_{j \neq i} (\bar{x})_j(M^r)_{:j}) + 2\lambda_i(M)_{:i} = 0 \quad (2.69)$$

$$(M)_{:i} = \frac{(\bar{x}^k)_i}{(\bar{x}^k)_i^2 + \lambda_i} \left(-H\bar{x}^k + \sum_{j \neq i} (\bar{x}^k)_j(M^r)_{:j} \right), \quad (2.70)$$

where (2.70) corresponds to the value that minimizes the local Lagrangian (2.68). We can solve problem (2.67) efficiently using a bisection algorithm on λ_i , until $\|(M)_{:i}\|^2 = \rho_i$, where $(M)_{:i}$ is determined with (2.70).

The proposal is described in [algorithm 8](#)

Algorithm 8 Distributed RLS algorithm.

- 1: Initialize $\mathbf{x}^0$, M^0 and $\{\alpha^k\}$. Set $k \leftarrow 0$.
 - 2: **while** STOP criteria is not satisfied on x^k **do**
 - 3: Set $q \leftarrow 0$
 - 4: **while** STOP criteria is not satisfied on Z^q **do**
 - 5: Set $r \leftarrow 0$
 - 6: **while** STOP criteria is not satisfied on M^r **do**
 - 7: Solve for every $i \in \{1, \dots, p+1\}$ problems (2.67) for given M^r , Z^q and $\bar{x}^k$.
 - 8: Set $r \leftarrow r + 1$.
 - 9: **end while**
 - 10: Set $Z^{q+1} \leftarrow [Z^q - \alpha_Z^q(-\bar{x}\bar{x}^T + \text{diag}[\lambda_1, \dots, \lambda_{p+1}])]_+$.
 - 11: Set $q \leftarrow q + 1$.
 - 12: **end while**
 - 13: Set $x^{k+1} \leftarrow x^k + \alpha_x^k 2(A + \Delta A^k)^T ((A + \Delta A^k)x^k - (b + \Delta b^k))$.
 - 14: Set $k \leftarrow k + 1$.
 - 15: **end while**
-

Simulations

We considered the RLS with Gaussian zero mean random matrix A and x of deviation one, and $b = Ax$. Then, we incorporated noise to the values A and b to perturb the system, again with zero mean Gaussian noise of deviation one. This is performed in order to use our robust estimator (2.61). The size of the problem was set to $p = 4$ and $N = 10$ samples. We performed 100 simulations of the problem and illustrated the average behavior of [algorithm 7](#) in [figure 2.3](#). The figure shows the worst-case LS error of the linear adjustment compared to the number of inner loop iterations. The first peak shown in the graph is because $(\Delta A, \Delta b)$ are initialized to zero, and then their values are calculated to maximize the cost. The algorithm converged in every simulation and shows a good performance.

Fig. 2.3 Centralized RLS problem

The distributed version [algorithm 8](#) for a single simulation is illustrated in [figure 2.4](#), where we showed the number of packet exchanges in the algorithm (total number of iterations in the most inner loop). The staircase behavior is due to the multiple loop scheme, which may be diminished with laxer convergence criteria.

Appendix 2.A Proofs

Proof of [Lemma 1](#). From [definition 2](#), the $\mathcal{S}$ -property holds if the following systems are strong alternatives:

$$\exists \mathbf{x} \in \mathbb{R}^p \mid f(\mathbf{x}) < \alpha \wedge g_i(x_i) \leq 0 \quad \forall i \in \mathcal{N}, \quad \Longleftrightarrow \quad (2.71a)$$

$$\nexists \boldsymbol{\lambda} \in \Gamma \mid f(\mathbf{x}) + \sum_{i \in \mathcal{N}} \lambda_i g_i(x_i) \geq \alpha \quad \forall \mathbf{x} \in \mathbb{R}^p. \quad (2.71b)$$

We need to prove the equivalence of [\(2.10b\)](#) and [\(2.71b\)](#). The implication [\(2.10b\)](#) $\Rightarrow$ [\(2.71b\)](#) is immediate from

$$[x, 1] \left(\bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{E} \right) [x, 1]^T = f(\mathbf{x}) + \sum_{i \in \mathcal{N}} \lambda_i g_i(x_i) - \alpha \geq 0. \quad (2.72)$$

Fig. 2.4 Distributed RLS problem

The implication (2.10b) $\Leftarrow$ (2.71b) is shown as follows. First, we define $h(\mathbf{x}, \boldsymbol{\lambda})$:

$$\begin{aligned} h(\mathbf{x}, \boldsymbol{\lambda}) &= f(\mathbf{x}) + \sum_{i \in \mathcal{N}} \lambda_i g_i(x_i) - \alpha \\ &= \mathbf{x}^T A_0 \mathbf{x} + 2b_0^T \mathbf{x} + c_0 - \alpha + \sum_{i \in \mathcal{N}} \lambda_i (\mathbf{x}^T A_i \mathbf{x} + 2b_i^T \mathbf{x} + c_i) \geq 0. \end{aligned} \quad (2.73)$$

Function $h(\mathbf{x}, \boldsymbol{\lambda})$ is nonnegative $\forall \mathbf{x} \in \mathbb{R}^p$ if and only if a minimum exists and it is nonnegative, and if the function is convex. This occurs if and only if

$$A_0 + \sum_i \lambda_i A_i \succeq 0 \quad (2.73a)$$

$$c_0 + \sum_i \lambda_i c_i - \alpha - (b_0 + \sum_i \lambda_i \tilde{b}_i)^T (A_0 + \sum_i \lambda_i A_i)^\dagger (b_0 + \sum_i \lambda_i \tilde{b}_i) \geq 0 \quad (2.73b)$$

$$b_0 + \lambda_i b_i \in \text{range}[A_0 + \sum_i \lambda_i A_i]. \quad (2.73c)$$

are all satisfied. Note that (2.73a) corresponds to the Hessian of $h(\mathbf{x}, \boldsymbol{\lambda})$, (2.73b) corresponds to $h(\mathbf{x}_{\min}, \boldsymbol{\lambda})$, where $\mathbf{x}_{\min} = -(A_0 + \sum_i \lambda_i A_i)^\dagger (b_0 + \sum_i \lambda_i \tilde{b}_i)$ is the minimum value of $h(\mathbf{x}, \boldsymbol{\lambda})$, and (2.73c) guarantees there is no direction where the problem is unbounded. See Appdx. A.5.4 in [15] for the complete description of the solution of a QP. Given (2.73a) and using the relation of the Schur complement of a matrix and its positive semidefiniteness as presented in page 651 in [15], we conclude that $\bar{A}_0 + \sum_i \lambda_i \bar{A}_i - \alpha \bar{E} \succeq 0$. This ends the proof of the equivalence. $\square$

Proof of Lemma 2. The proof is a standard result in semidefinite programming and can be derived in a very similar way to Example 5.14 in [15]. We reproduce it here for completeness.

Consider the following problem:

$$\begin{aligned} \min_{s \in \mathbb{R}, \boldsymbol{\lambda} \in \Gamma} \quad & s \\ \text{s.t.} \quad & \bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{E} \succeq -s \mathbf{I}_{p+1}. \end{aligned} \quad (2.74)$$

The Lagrangian has the following form:

$$\begin{aligned} L(s, \boldsymbol{\lambda}, Z) &= s - \text{tr}[(\bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{E} + s \mathbf{I}_{p+1})Z] \\ &= s - \text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] - \sum_{i \in \mathcal{N}} \lambda_i \text{tr}[\bar{A}_i Z] - s \text{tr}[Z] \end{aligned} \quad (2.75)$$

and the dual function can take the following values:

$$\min_{s \in \mathbb{R}, \boldsymbol{\lambda} \in \Gamma} L(s, \boldsymbol{\lambda}, Z) = \begin{cases} -\text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] & \text{if } \text{tr}[\bar{A}_i Z] \leq 0 \forall i, \text{tr}[Z] = 1 \\ -\infty & \text{otherwise.} \end{cases} \quad (2.76)$$

Maximizing the dual function given by (2.76) we obtain the dual problem:

$$\begin{aligned} \max_Z \quad & -\text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] \\ \text{s.t.} \quad & \text{tr}[\bar{A}_i Z] \leq 0 \forall i \\ & Z \succeq 0, \quad \text{tr}[Z] = 1. \end{aligned} \quad (2.77)$$

Primal problem (2.74) is convex and has non-empty interior (it satisfies Slater's condition) and, therefore, strong duality holds with its dual formulation given by (2.77).

A technical condition needs to be satisfied to establish the strong alternative property between the systems described in (2.13): problem (2.74) has to attain its infimum. A sufficient condition is given by the following implication (see Example 5.14 in [15]):

$$\mathbf{v}_i \in \Gamma, \sum_{i \in \mathcal{N}} \mathbf{v}_i \bar{A}_i \succeq 0 \implies \mathbf{v}_i = 0 \quad (\forall i \in \mathcal{N}) \implies \sum_{i \in \mathcal{N}} \mathbf{v}_i \bar{A}_i = 0 \quad (2.78)$$

which in our case is satisfied because of **Assumption 1**. In particular, every matrix $\bar{A}_i$ has at least a negative eigenvalue in the case of **QPI**, and positive and negative eigenvalues in the case of **QPE**, because of Slater's condition. Moreover, matrices $\bar{A}_i$ do not overlap between each other except for the matrix elements $(p+1, p+1)$. As a consequence, any positive linear combination of these matrices will have at least a negative eigenvalue, which fulfills the condition described in (2.78).

It is now clear that primal problem (2.74) and dual problem (2.77) attain the same objective value. Therefore, if $s > 0$, then $\text{tr}[(\bar{A}_0 - \alpha \bar{E})Z] < 0$ and there exists no $\lambda \in \Gamma$ such that $\bar{A}_0 + \sum_{i \in \mathcal{N}} \lambda_i \bar{A}_i - \alpha \bar{E} \succeq 0$. This shows that both conditions are strong alternatives and it concludes the proof of the lemma. $\square$

Proof of Lemma 4. Implication (2.19a) $\Leftarrow$ (2.19b) is verified, since we can introduce $z = (\mathbf{x}^T, 1)^T = Py$ and satisfy (2.19a):

$$f(\mathbf{x}) - \alpha = z^T (\bar{A}_0 - \alpha \bar{E}) z = y^T P^T (\bar{A}_0 - \alpha \bar{E}) Py = y^T F_0 y < 0 \quad (2.79a)$$

$$g_i(\mathbf{x}) = z^T \bar{A}_i z = y^T P^T \bar{A}_i Py = y^T F_i y \leq 0, \quad \forall i \in \mathcal{N}. \quad (2.79b)$$

The opposite direction (2.19a) $\Rightarrow$ (2.19b) is more elaborate. We first need to show the following equivalence:

$$\exists y \in \mathbb{R}^{p+1} \mid y^T F_0 y < 0 \wedge y^T \tilde{F}_i y \leq 0, \forall i \in \mathcal{N} \iff \quad (2.80a)$$

$$\exists \mathbf{s} \in \mathbb{R}^p \mid \mathbf{s}^T \tilde{F}_0 \mathbf{s} + 2\tilde{f}_0^T \mathbf{s} + d_0 < 0 \wedge \mathbf{s}^T \tilde{F}_i \mathbf{s} + e_i \leq 0 \quad \forall i \in \mathcal{N}, \quad (2.80b)$$

where $\tilde{F}_0, \tilde{f}_0, d_0, \tilde{F}_i$ and e_i are defined in (2.81).

Consider $y = (v^T, w)^T$ with $v \in \mathbb{R}^p$ and $w \in \mathbb{R}$. We have

$$\begin{pmatrix} v \\ w \end{pmatrix}^T \underbrace{\begin{pmatrix} \tilde{F}_0 & \tilde{f}_0 \\ \tilde{f}_0^T & d_0 \end{pmatrix}}_{F_0} \begin{pmatrix} v \\ w \end{pmatrix} < 0, \quad \begin{pmatrix} v \\ w \end{pmatrix}^T \underbrace{\begin{pmatrix} \tilde{F}_i & 0 \\ 0^T & e_i \end{pmatrix}}_{F_i} \begin{pmatrix} v \\ w \end{pmatrix} \leq 0, \quad \forall i \in \mathcal{N}. \quad (2.81)$$

If $w \neq 0$ we can choose $\mathbf{s} = v/w$ and (2.80b) is feasible. If $w = 0$ we have $v^T \tilde{F}_0 v < 0$ and $v^T \tilde{F}_i v \leq 0$. We can choose $\mathbf{s} = \sum_{i \in \mathcal{N}} t_{si} \hat{\mathbf{s}}_i + t_v v$, where $\hat{\mathbf{s}}$ is a point satisfying Slater's condition (see Assumption 1) in the congruent domain. We define $\hat{\mathbf{s}}$ such that $\hat{\mathbf{s}} = \sum_i \hat{\mathbf{s}}_i$ and the support of all $\hat{\mathbf{s}}_i$ do not overlap. This results into the following:

$$\begin{aligned} & \mathbf{s}^T \tilde{F}_0 \mathbf{s} + 2\tilde{f}_0^T \mathbf{s} + d_0 \\ &= \left(\sum_i t_{si} \hat{\mathbf{s}}_i \right)^T \tilde{F}_0 \left(\sum_i t_{si} \hat{\mathbf{s}}_i \right) + 2\tilde{f}_0^T \left(\sum_i t_{si} \hat{\mathbf{s}}_i \right) + d_0 + \underbrace{t_v^2 v^T \tilde{F}_0 v}_{<0} + 2t_v \left(\sum_i t_{si} \tilde{F}_0 \hat{\mathbf{s}}_i + \tilde{f}_0 \right)^T v \end{aligned} \quad (2.82a)$$

$$\mathbf{s}^T \tilde{F}_i \mathbf{s} + e_i = \underbrace{t_{si}^2 \hat{\mathbf{s}}_i^T \tilde{F}_i \hat{\mathbf{s}}_i}_{<0} + \underbrace{t_v^2 v^T \tilde{F}_i v}_{\leq 0} + 2t_v t_{si} \hat{\mathbf{s}}_i^T \tilde{F}_i v < 2t_v t_{si} \hat{\mathbf{s}}_i^T \tilde{F}_i v \quad (2.82b)$$

By taking $t_v \rightarrow \infty$ and choosing $t_{si} = \pm 1$ depending on the signs of $\hat{s}_i^T \tilde{F}_i v$, we can find an $\mathbf{s}$ such that (2.80b) is satisfied. In the original domain, we can recover $\mathbf{x}$ with $(\mathbf{x}, 1)^T = P \cdot (\mathbf{s}, 1)^T$, which satisfies (2.19b). This completes the proof. $\square$

Proof of Theorem 2. Because $Y \succeq 0$, it admits an orthogonal decomposition such that $Y = VV^T$ where $V = [v_1, v_2, \dots, v_R]$ and $R \leq p+1$ represents the rank of Y . Then,

$$\text{tr}[F_0 Y] = \text{tr}[V^T F_0 V] = \sum_{r=1}^R v_r^T F_0 v_r = \sum_{r=1}^R \sum_{i=1}^{p+1} (F_0)_{ii} v_{ri}^2 + 2 \sum_{j>i} (F_0)_{ij} v_{ri} v_{rj} < 0 \quad (2.83)$$

$$\text{tr}[F_i Y] = \text{tr}[V^T F_i V] = \sum_{r=1}^R v_r^T F_i v_r = \sum_{r=1}^R \sum_{j=1}^{p+1} (F_i)_{jj} v_{rj}^2 = \psi_i \quad \forall i \in \mathcal{N}, \quad (2.84)$$

where v_{rj} corresponds to the j 'th component of vector v_r . Recall that $(F_i)_{ij} = 0$ for $i \neq j$. We consider vector $y = (y_j)_{j=1}^{p+1}$ such that $y_j^2 = \sum_{r=1}^R v_{rj}^2$. This implies that

$$y^T F_i y = \sum_{j=1}^{p+1} (F_i)_{jj} y_j^2 = \sum_{j=1}^{p+1} (F_i)_{jj} \sum_{r=1}^R v_{rj}^2 = \sum_{r=1}^R \sum_{j=1}^{p+1} (F_i)_{jj} v_{rj}^2 = \text{tr}[F_i Y] = \psi_i. \quad (2.85)$$

We introduce a diagonal matrix D with $(D)_{jj} = \pm 1$ such that $y_j = (D)_{jj} \sqrt{\sum_{r=1}^R v_{rj}^2}$. The product with F_0 becomes:

$$\begin{aligned} y^T F_0 y &= \sum_{i=1}^{p+1} (F_0)_{ii} y_i^2 + 2 \sum_{j>i} (F_0)_{ij} y_i y_j \\ &= \sum_{i=1}^{p+1} (F_0)_{ii} \left(\sum_{r=1}^R v_{ri}^2 \right) + 2 \sum_{j>i} (F_0)_{ij} \left((D)_{ii} \sqrt{\sum_{r=1}^R v_{ri}^2} \right) \left((D)_{jj} \sqrt{\sum_{r=1}^R v_{rj}^2} \right) \end{aligned}$$

Consider the difference

$$y^T F_0 y - \text{tr}[F_0 Y] = 2 \sum_{j>i} (F_0)_{ij} \left((D)_{ii} (D)_{jj} \sqrt{\sum_{r,s=1}^R v_{ri}^2 v_{sj}^2} - \sum_{r=1}^R v_{ri} v_{rj} \right) \stackrel{?}{\leq} 0 \quad (2.86)$$

Expression (2.86) is nonpositive if $(F_0)_{ij} (D)_{ii} (D)_{jj}$ is nonpositive for every pair $i \neq j$. This is shown in Proposition 4. $\square$

Proposition 4. *The following relation holds true for any sequence v_{ri}, v_{rj} , $r \in \{1, \dots, R\}$ and pair $i, j \in \{1, \dots, p+1\}$:*

$$\sqrt{\sum_{r,s=1}^R v_{ri}^2 v_{sj}^2} \geq \sum_{r=1}^R v_{ri} v_{rj} \quad (2.87)$$

Proof. Take squares on the left and right hand sides of (2.87), which gives

$$\sum_{r,s=1}^R v_{ri}^2 v_{sj}^2 \geq \sum_{r,s=1}^R v_{ri} v_{sj} v_{rj} v_{si} \quad (2.88)$$

Note that $a^2 + b^2 \geq 2ab$ for any $a, b \in \mathbb{R}$. Introduce $a = v_{ri} v_{sj}$ and $b = v_{rj} v_{si}$:

$$v_{ri}^2 v_{sj}^2 + v_{rj}^2 v_{si}^2 \geq 2v_{ri} v_{sj} v_{rj} v_{si}, \quad (2.89)$$

which is satisfied for every pair of r, s . Summing all of these inequalities yields (2.88) which implies (2.87). $\square$

Proof of Lemma 5. Implication (2.20a) $\Leftarrow$ (2.20b) is immediate, since we can introduce $z = (\mathbf{x}^T, 1)^T = Py$ and satisfy (2.20a) same as in Lemma 4. The direction (2.20a) $\Rightarrow$ (2.20b) is more elaborate and we need to show the following equivalence:

$$\exists y \in \mathbb{R}^{p+1} \mid y^T F_0 y < 0 \wedge y^T \tilde{F}_i y = 0, \forall i \in \mathcal{N} \iff \quad (2.90a)$$

$$\exists \mathbf{s} \in \mathbb{R}^p \mid \mathbf{s}^T \tilde{F}_0 \mathbf{s} + 2\tilde{f}_0^T \mathbf{s} + d_0 < 0 \wedge \mathbf{s}^T \tilde{F}_i \mathbf{s} + e_i = 0, \forall i \in \mathcal{N}. \quad (2.90b)$$

Consider $y = (v^T, w)^T$ with $v \in \mathbb{R}^p$ and $w \in \mathbb{R}$. We have

$$\begin{pmatrix} v \\ w \end{pmatrix}^T \begin{pmatrix} \tilde{F}_0 & \tilde{f}_0 \\ \tilde{f}_0^T & d_0 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} < 0, \quad \begin{pmatrix} v \\ w \end{pmatrix}^T \begin{pmatrix} \tilde{F}_i & 0 \\ 0^T & e_i \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = 0, \quad \forall i \in \mathcal{N}. \quad (2.91)$$

If $w \neq 0$ we can choose $\mathbf{s} = v/w$ and (2.90b) is feasible. If $w = 0$ we have $v^T \tilde{F}_0 v < 0$ and $v^T \tilde{F}_i v = 0$ for all i . We require that $\tilde{F}_i \neq 0$ so that $\hat{s}_i^T \tilde{F}_i \hat{s}_i \neq 0$. This is equivalent to require that $A_i \neq 0$. We try $\mathbf{s} = \sum_i t_{si} \hat{s}_i + t_v v$ where the support of every $\hat{s}_i$ does not overlap. We get the following:

$$\mathbf{s}^T \tilde{F}_0 \mathbf{s} + 2\tilde{f}_0^T \mathbf{s} + d_0 \quad (2.92)$$

$$= (\sum_i t_{si} \hat{s}_i)^T \tilde{F}_0 (\sum_i t_{si} \hat{s}_i) + 2\tilde{f}_0^T (\sum_i t_{si} \hat{s}_i) + d_0 + \underbrace{t_v^2 v^T \tilde{F}_0 v}_{<0} + 2t_v (\sum_i t_{si} \tilde{F}_0 \hat{s}_i + \tilde{f}_0)^T v$$

$$\mathbf{s}^T \tilde{F}_i \mathbf{s} + e_i = \underbrace{t_{si}^2 \hat{s}_i^T \tilde{F}_i \hat{s}_i}_A + \underbrace{t_v^2 v^T \tilde{F}_i v}_{=0} + \underbrace{2t_{si} t_v \hat{s}_i^T \tilde{F}_i v}_B \quad (2.92a)$$

The idea is the same as before: make t_v large enough so that (2.93) becomes negative. We need to verify that (2.92a) equals zero for some t_{si} . We consider two cases:

1. If $v \in \text{null}[\tilde{F}_i]$, then $\hat{s}_i^T \tilde{F}_i v = 0$, so we need to choose $\hat{s}_i$ such that A equals zero. Such point is guaranteed to exist because of [Assumption 1](#).
2. If $v \notin \text{null}[\tilde{F}_i]$, then choose $\hat{s}_i$ such that $\hat{s}_i^T \tilde{F}_i \hat{s}_i \neq 0$. Start with $t_{si} = 0$ and increase its value until $A = -B$. Choose the appropriate sign accordingly.

We recover $\mathbf{x}$ with $(\mathbf{x}, 1)^T = P \cdot (\mathbf{s}, 1)^T$, which satisfies (2.20b). $\square$

Proof of Theorem 4. Consider $Y \succeq 0$ with spectral decomposition $Y = VV^T$ where $V = [v_1, \dots, v_R]$ and $R \leq p+1$ represents the rank of Y . We have in the constraints that

$$\text{tr}[F_i Y] = \text{tr}[V^T F_i V] = \sum_{r=1}^R \sum_{j \notin M} (F_i)_{jj} v_{rj}^2 + \sum_{m \in M} 2(F_i)_{m,p+1} v_{r,p+1} v_{r,m} = \psi_i \quad \forall i \in \mathcal{N}, \quad (2.93)$$

where $v_r = (v_{rj})_{j=1}^{p+1}$. We can choose a rank 1 candidate solution $y = (y_j)_{j=1}^{p+1}$ such that

$$y_j = (D)_{jj} \sqrt{\sum_r v_{rj}^2}, \quad j \notin M, \quad (2.94)$$

$$y_j = (D)_{p+1,p+1} \frac{\sum_r v_{r,p+1} v_{ri}}{y_{p+1}} = \frac{\sum_r v_{r,p+1} v_{ri}}{\sqrt{\sum_r v_{r,p+1}^2}}, \quad j \in M. \quad (2.95)$$

Using vector y in the constraints gives

$$y^T F_i y = \sum_{j \notin M} (F_i)_{jj} y_j^2 + \sum_{m \in M} 2(F_i)_{m,p+1} y_{p+1} y_m \quad (2.96)$$

$$= \sum_{r=1}^R \sum_{j \notin M} (F_i)_{jj} v_{rj}^2 + \sum_{m \in M} 2(F_i)_{m,p+1} v_{r,p+1} v_{r,m} = \psi_i \quad \forall i \in \mathcal{N}. \quad (2.97)$$

The choice of vector y fulfills $\text{tr}[F_i Y] = y^T F_i y = \psi_i$ for $i \in \mathcal{N}$. We need to verify that $y^T F_0 y \leq \text{tr}[F_0 Y]$. Consider the following:

$$\text{tr}[F_0 Y] = \text{tr}[V^T F_0 V] = \sum_{r=1}^R v_r^T F_0 v_r = \sum_{r=1}^R \sum_{i=1}^{p+1} (F_0)_{ii} v_{ri}^2 + 2 \sum_{j>i} (F_0)_{ij} v_{ri} v_{rj} < 0 \quad (2.98)$$

$$y^T F_0 y = \sum_i (F_0)_{ii} y_i^2 + 2 \sum_{j>i} (F_0)_{ij} y_i y_j \quad (2.99)$$

and compare both expressions term by term. From (2.99) we get

$$(F_0)_{ii}y_i^2 = \sum_r (F_0)_{ii}v_{ri}^2 \quad i \notin M \quad (2.100)$$

$$(F_0)_{ij}y_i y_j = \sum_r (D)_{ii}(D)_{jj}(F_0)_{ij} \sqrt{\sum_r v_{ri}^2} \sqrt{\sum_r v_{rj}^2} \quad i, j \notin M \quad (2.101)$$

$$(F_0)_{ii}y_i^2 = (F_0)_{ii} \frac{(\sum_r v_{r,p+1}v_{ri})^2}{\sum_r v_{r,p+1}^2} \quad i \in M \quad (2.102)$$

$$(F_0)_{i,p+1}y_i y_{p+1} = \sum_r (F_0)_{i,p+1}v_{ri}v_{r,p+1} \quad i \in M \quad (2.103)$$

$$(F_0)_{i,j} = 0 \quad i \in M, j \neq p+1, j \neq i \quad (2.104)$$

Now we can establish the following relations when comparing term by term with (2.98):

$$(F_0)_{ii}y_i^2 = \sum_r (F_0)_{ii}v_{ri}^2 \quad i \notin M \quad (2.105)$$

$$(F_0)_{ij}y_i y_j \leq \sum_r (F_0)_{ij}v_{ri}v_{rj} \quad i, j \notin M \text{ if } (D)_{ii}(D)_{jj}(F_0)_{ij} \leq 0 \quad (2.106)$$

$$(F_0)_{ii}y_i^2 \leq \sum_r (F_0)_{ii}v_{ri}^2 \quad i \in M \text{ if } (F_0)_{ii} \geq 0 \quad (2.107)$$

$$(F_0)_{i,p+1}y_i y_{p+1} = \sum_r (F_0)_{i,p+1}v_{ri}v_{r,p+1} \quad i \in M \quad (2.108)$$

$$(F_0)_{i,j}y_i y_j = \sum_r (F_0)_{i,j}v_{ri}v_{rj} = 0 \quad i \in M, j \neq p+1, j \neq i. \quad (2.109)$$

and conclude that

$$y^T F_0 y \leq \text{tr}[F_0 Y] \quad (2.110)$$

$$y^T F_i y = \text{tr}[F_i Y] = \psi_i \quad \forall i \in \mathcal{N}. \quad (2.111)$$

□

Proof of Lemma 6. Implication (2.24a) $\Leftarrow$ (2.24b) is verified, since we can introduce $z = (\mathbf{x}^T, 1)^T = Py$:

$$f(\mathbf{x}) - \alpha = z^T (\bar{A}_0 - \alpha \bar{E}) z = y^T P^T (\bar{A}_0 - \alpha \bar{E}) P y = y^T F_0 y < 0 \quad (2.112a)$$

$$g_i(\mathbf{x}) = z^T \bar{A}_i z = y^T P^T \bar{A}_i P y = y^T F_i y \leq 0, \quad \forall i \in \mathcal{N}. \quad (2.112b)$$

The opposite direction (2.24a) $\implies$ (2.24b) is also simple. Consider $y = (v^T, w)^T$ with $v \in \mathbb{R}^p$ and $w \in \mathbb{R}$.

$$\begin{pmatrix} v \\ w \end{pmatrix}^T \underbrace{\begin{pmatrix} \tilde{F}_0 & \tilde{f}_0 \\ \tilde{f}_0^T & d_0 \end{pmatrix}}_{F_0} \begin{pmatrix} v \\ w \end{pmatrix} < 0, \quad \begin{pmatrix} v \\ w \end{pmatrix}^T \underbrace{\begin{pmatrix} \tilde{F}_i & 0 \\ 0^T & e_i \end{pmatrix}}_{F_i} \begin{pmatrix} v \\ w \end{pmatrix} \leq 0, \quad \forall i \in \mathcal{N}. \quad (2.113)$$

If $w \neq 0$ we can choose $s = v/w$ and (2.24b) follows. We cannot have $w = 0$ because $A_0 \succeq 0$ by assumption, and $v^T \tilde{F}_0 v \not\leq 0$ for any v . $\square$

Proof of Proposition 3. We need to show that if $\lambda_i \rightarrow \pm\infty$, it necessarily implies that $\Upsilon_i(\lambda_i, \mathbf{x}^k, Z^k) \rightarrow -\infty$. We start with the case in which $\lambda_i \rightarrow +\infty$. We distinguish two cases, one in which $\tilde{A}_i$ has at least one negative eigenvalue, and another where all eigenvalues are nonnegative. In the first case, $\lambda_i^{\max} < \infty$, and $\Upsilon_i(\lambda_i; \mathbf{x}^k) = -\infty$ for $\lambda_i \notin \Lambda_i$. In the second case, $\lambda_i^{\max}$ is unbounded, so we have $\tilde{A}_0^{ii} + \rho_i \mathbf{I}_{n_i} + \lambda_i \tilde{A}_i \succeq 0$ for all $\lambda_i \geq \lambda_i^{\min}$ and the limit results into

$$\begin{aligned} \lim_{\lambda_i \rightarrow \infty} \Upsilon_i(\lambda_i; \mathbf{x}, Z^k) &= \lim_{\lambda_i \rightarrow \infty} \lambda_i \left(c_i - \left(\frac{r}{\lambda_i} + s_i \right)^T \left(\frac{1}{\lambda_i} (\tilde{A}_0^{ii} + \rho \mathbf{I}_{n_i}) + \tilde{A}_i \right)^\dagger \left(\frac{r}{\lambda_i} + s_i \right) \right) \\ &= \lim_{\lambda_i \rightarrow \infty} \lambda_i \left(c_i - s_i^T \tilde{A}_i^\dagger s_i \right) = -\infty \end{aligned} \quad (2.114)$$

where we have used the fact that the singular values of the pseudoinverse are continuous in λ_i and $c_i - s_i^T \tilde{A}_i^\dagger s_i < 0$ as analyzed in the proof of Proposition 2. The case in which $\lim_{\lambda_i \rightarrow -\infty}$ is similar to the case we have just proven.

Existence of a solution is proven by the Weirstrass theorem [9, Prop.3.2.1], since $\Upsilon_i(\lambda_i; \mathbf{x}^q, Z^k)$ is coercive and has non-empty interior. Existence of unique solution in (2.42) is satisfied because the Lagrangian becomes strongly convex in x_i for some $\rho_i > 0$. $\square$

Chapter 3

Squared ranged localization problems

3.1 Introduction

Localization problems arise in a variety of scenarios and applications, and have been amply studied in the literature. From GPS techniques developed for exterior localization, to interior mappings using WiFi base stations, the localization problem can be posed in many different ways for a wide range of applications.

In this chapter, we consider two localization problems: one in which we want to calculate the position of a target node using noisy measurements of distances, and a second one in which some nodes know their position but others do not. We refer to the first problem as target localization (TL) and to the second one as sensor network localization (SNL).

In particular, we consider different estimators used in the literature and propose distributed algorithms that converge to either a local minimum or to an optimal solution, depending on the scheme used. We simulate our proposals and illustrate their behavior comparing them with other state of the art alternatives. We also discuss the pros and cons of each method regarding the performance, convergence speed and complexity.

3.2 Target localization

The problem in this section consists on determining the position of a target node using noisy measurements of distances between the target and the network nodes. Given a set of nodes $\mathcal{N} = \{1, \dots, N\}$ with known position $s_i \in \mathbb{R}^p$, the distance measurements d_i between the network node $i \in \mathcal{N}$ and the target node, we consider solving the problem of estimating

the target node's position with the following cost function:

$$(\text{SRLS-TL}): \min_x \sum_{i \in \mathcal{N}} (\|x - s_i\|^2 - d_i^2)^2, \quad (3.1)$$

where $x \in \mathbb{R}^p$ and p is typically 2 or 3, if the localization is in the plane or in a three-dimensional space, respectively. The name of the estimator stands for “squared ranged least squares target localization” and the nomenclature is borrowed from [5].

Problem (3.1) has been widely studied in the literature, and even though it is a nonconvex problem, the global solution can be found efficiently in a centralized fashion, as shown in [5]. Many other centralized approaches solving the same cost function have been proposed as well, as [78, 53], which relax the problem in some manner in order to obtain an approximate convex version of it. However, these other proposals may not achieve a global solution.

Decentralized approaches have also appeared more recently, such as in [20] and [27], where every node contributes to minimize a global function by minimizing a local version, and afterwards, combine their estimate with the one from their neighbors. In the case of reference [20], the minimizing step is performed using a gradient descent of the nonconvex local function, whereas in reference [27], the minimization is performed by solving a local convexified version of (3.1). Convergence to a stationary point is guaranteed in [27], and in the case of reference [20], convergence has been later analyzed in [11]. Neither of the two references have convergence guarantees to the global minimizer.

The derivation of the optimal solution can be followed in Section II-B in [5], but we summarize it here for completeness. First, we reformulate (3.1) as an equivalent problem:

$$\begin{aligned} \min_{x \in \mathbb{R}^p} \quad & \sum_{i=1 \in \mathcal{N}} (\xi - 2s_i^T x + \|s_i\|^2 - d_i^2)^2 \\ \text{s.t.} \quad & \|x\|^2 = \xi. \end{aligned} \quad (3.2)$$

By introducing $y = (x, \xi)$ in problem (3.2), we get:

$$\begin{aligned} \min_{y \in \mathbb{R}^{p+1}} \quad & \|By - c\|^2 \\ \text{s.t.} \quad & y^T Dy + 2f^T y = 0 \end{aligned} \quad (3.3)$$

where B , c , D and f are defined as follows:

$$B = \begin{pmatrix} -2x_1^T & 1 \\ \vdots & \vdots \\ -2x_n^T & 1 \end{pmatrix}, \quad c = \begin{pmatrix} d_1^2 - \|x_1\|^2 \\ \vdots \\ d_n^2 - \|x_n\|^2 \end{pmatrix}, \quad D = \begin{pmatrix} \mathbf{I}_n & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & 0 \end{pmatrix}, \quad f = \begin{pmatrix} \mathbf{0}_{n \times 1} \\ -0.5 \end{pmatrix}. \quad (3.4)$$

The optimization problem (3.3) is an instance of a quadratic problem with a single quadratic equality constraint. As we saw in chapter 1, this kind of problems have strong duality between the primal and dual formulations provided certain constraint qualifications are satisfied, see Theorem 3.2 in [56]. Problem (3.3) satisfies these requirements and can be solved with the dual formulation. The necessary conditions that a candidate solution must satisfy are given by

$$(B^T B + \lambda D)y - B^T c - \lambda f = 0 \quad (3.5a)$$

$$y^T D y + 2f^T y = 0 \quad (3.5b)$$

$$B^T B + \lambda D \succeq 0, \quad (3.5c)$$

where (3.5a) corresponds to the first order condition derived from the Lagrangian, (3.5b) amounts to feasibility, and (3.5c) is a consequence of Theorem 1 from [52].

The previous conditions can be reformulated as a search problem:

$$\begin{aligned} & \underset{\lambda \in \mathbb{R}}{\text{solve}} \quad \hat{y}(\lambda)^T D \hat{y}(\lambda) + 2f^T \hat{y}(\lambda) = 0 \\ & \text{s.t.} \quad \hat{y}(\lambda) = (B^T B + \lambda D)^{-1} (B^T c - \lambda f) \\ & \quad \quad B^T B + \lambda D \succeq 0. \end{aligned} \quad (3.6)$$

The objective function of (3.6) is monotone in λ , so a bisection search algorithm can be used, [56, Th.5.2]. This method will converge very quickly to the unique root of the system rather than using interior point methods that solve the semidefinite equivalent problem.

In algorithm 9 we describe our own implementation based on a bisection algorithm. We defined $\bar{\lambda}$ and $\underline{\lambda}$ as the upper and lower limits of λ in the bisection search, respectively, and ε is a parameter that specifies the accuracy of the bisection search. Step 3 finds an upper limit where function $\hat{y}(\lambda)$ changes sign. From steps 4–9 the bisection search is performed.

The interest behind finding optimal solutions in the TL problem is because the problem has suboptimal solutions, i.e., not every stationary point is a global minimum of problem (3.1). This claim can be appreciated by the following example where 3 network nodes are placed in the plane at coordinates $s_1 = (-5, 8)^T$, $s_2 = (-3, -9)^T$ and $s_3 = (-4, 2)^T$ almost aligned but not completely. The global solution when the given distances to problem (3.1) have no noise is at point $x^* = (1, 1)^T$ and a second suboptimal stationary point is approximately at $x' = (-9.01, -0.15)^T$. The gradients of (3.1) cancel at these points, whereas the Hessian matrices, which are given by the expression

$$\nabla_x^2 f(x) = \sum_{i \in \mathcal{N}} 8(x - s_i)^T (x - s_i) + 4(\|x - s_i\|^2 - d_i^2) \mathbf{I}_p, \quad (3.7)$$

Algorithm 9 Centralized bisection algorithm for the SRLS-TL problem

```

1: Initialize  $\underline{\lambda} = -1/\max \text{eig}(\mathbf{D}, \mathbf{B}^T \mathbf{B})$ ,  $\bar{\lambda} = 2|\underline{\lambda}| + 1$ .
2: while  $\hat{y}(\lambda)^T D y(\lambda) + 2f^T \hat{y}(\lambda) \geq 0$  do  $\bar{\lambda} \leftarrow 2\bar{\lambda}$ 
3: end while
4: Set  $\lambda \leftarrow \frac{1}{2}(\bar{\lambda} + \underline{\lambda})$ .
5: while  $\bar{\lambda} - \underline{\lambda} \geq \varepsilon$  do
6:   if  $\hat{y}(\lambda)^T D y(\lambda) + 2f^T \hat{y}(\lambda) \geq 0$  then  $\underline{\lambda} \leftarrow \lambda$ 
7:   else  $\bar{\lambda} \leftarrow \lambda$ .
8:   end if
9: end while
10: Set  $[x^T, \alpha] \leftarrow \hat{y}^T(\lambda)$ . Return  $x$ .

```

are positive definite when particularized to these points. We denoted the objective function of the problem with $f(x)$, i.e., $f(x) = \sum_{i \in \mathcal{N}} (\|x - s_i\|^2 - d_i^2)^2$. [Figure 3.1a](#) shows the network disposition and [figure 3.1b](#) the nonconvexity of problem (3.1).

This example illustrates that any descent method that tries to solve the primal problem directly is prone to converge to a local minimum depending on the initialization point. We note however that if the scenario for TL is well conditioned, meaning that there are many network nodes, the nodes are well spread in the search space, and the target is placed in an area well covered, then, the cost function may present a unique minimum point. This kind of well conditioned scenario would justify the good behavior that most localization algorithms present when they use descent methods of the primal problem.

For these reasons, in the following subsections we propose and analyze three different distributed algorithms that solve problem (3.1). The first two algorithms are fast and require few communication exchanges, but only have convergence guarantees to suboptimal stationary points. On the other hand, the third algorithm we propose is based on dual decomposition techniques and it converges to the optimal solution of TL problem provided that the $\mathcal{S}$ -property holds in our quadratic decomposition. It has the drawback that it requires more communication messages, and may be slower to converge. We also compare the behavior of our algorithms with those present in the literature. [Table 3.1](#) provides a summary of our contributions and existing algorithms in the literature, and short names for ease of reference in our simulations. Conditions that analyze the $\mathcal{S}$ -property with our proposed decomposition are analyzed at the end of [section 3.2.3](#).

(a) Example of TL problem.

(b) Cost function of TL problem.

Fig. 3.1 Example of TL network and cost.

	Convergence	Description
Centralized	optimal	proposed in [5]
ADMM-Exact	suboptimal	algorithm 10
NEXT-Exact	suboptimal	algorithm 11
Dual ascent	optimal if the $\mathcal{S}$ -property is satisfied	algorithm 13
Distributed dual ascent	optimal if the $\mathcal{S}$ -property is satisfied	algorithm 14
Diffusion-ATC	suboptimal	proposed in [20]
NEXT-Linearized	suboptimal	proposed in [27]

Table 3.1 Summary of TL algorithms.

3.2.1 ADMM decomposition

The derivation of our first proposal is based on ADMM decomposition (Alternating Direction Method of Multipliers). ADMM methods form a general framework of algorithms that allow convex problems with a certain structure to be decomposed and solved in a distributed manner. A survey paper analyzing some of the last contributions in this area is [14]. However, ADMM has no convergence guarantees when the problem is nonconvex, as is the case in (3.1). Furthermore, and to the best of our knowledge, there are only few publications that consider nonconvex problems using ADMM, such as [81, 47] and some examples from [14], and no general convergence analysis in these cases. Therefore, we present some convergence results of our algorithm towards a stationary solution.

The original problem (3.1) can be reformulated as follows:

$$\begin{aligned} \min_{\mathbf{t}, \mathbf{x}, z} \quad & \sum_{i \in \mathcal{N}} (t_i - d_i^2)^2 \\ \text{s.t.} \quad & \|x_i - s_i\|^2 - t_i = 0 \quad \forall i \in \mathcal{N} \\ & x_i = z \quad \forall i \in \mathcal{N} \end{aligned} \quad (3.8)$$

where $\mathbf{x} = (x_i)_{i \in \mathcal{N}}$, $\mathbf{t} = (t_i)_{i \in \mathcal{N}}$, $t_i \in \mathbb{R}$ and $x_i, z \in \mathbb{R}^p$.

Problem (3.8) is nonconvex, but includes slack variable z to facilitate a separable formulation. In particular, the augmented Lagrangian can be decomposed as follows:

$$L_\rho(\mathbf{t}, \mathbf{x}, z, \boldsymbol{\mu}) = \sum_{i \in \mathcal{N}} L_\rho^i(t_i, x_i, z, \mu_i) \quad (3.9)$$

where the local Lagrangians have the following form:

$$L_\rho^i(t_i, x_i, z, \mu_i) = (t_i - d_i^2)^2 + \mu_i^T (x_i - z) + \rho \|x_i - z\|^2, \quad (3.10)$$

and $\boldsymbol{\mu} = (\mu_i)_{i \in \mathcal{N}}$, $\mu_i \in \mathbb{R}^p$ are dual variables associated with the cost of disagreement between the x_i and z ; $\rho > 0$ weights the proximal term that induces strong convexity in x_i ; and, variables x_i, t_i have to satisfy $\|x_i - s_i\|^2 = t_i$.

From the Lagrangian (3.9), we can derive the ADMM steps:

1. Solve $\forall i \in \mathcal{N}$ and assign the solution to t_i^{k+1} and x_i^{k+1} :

$$\begin{aligned} \min_{t_i, x_i} \quad & L_\rho^i(t_i, x_i, z^k, \mu_i^k) \\ \text{s.t.} \quad & \|x_i - s_i\|^2 - t_i = 0. \end{aligned} \quad (3.11)$$

2. Update $z^{k+1} = \frac{1}{N} \sum_{i \in \mathcal{N}} (x_i^{k+1} + \frac{1}{2\rho} \mu_i^k)$.

3. Update $\mu_i^{k+1} = \mu_i^k + 2\rho(x_i^{k+1} - z^{k+1})$.

In step 1 every network node $i \in \mathcal{N}$ solves a nonconvex problem in parallel. Then, estimates are interchanged between nodes, or average consensus is executed, see e.g. [84], in order to update z . Finally, dual variables μ_i are updated locally.

Nonconvex analysis of the quadratic problem using ADMM decomposition

The optimization problem (3.11) is an instance of a quadratic problem with a single quadratic equality constraint. These kind of problems have strong duality between its primal and dual formulations provided certain constraint qualifications are satisfied as we have already discussed (Theorem 3.2 [56]). We reformulate problem (3.11) into

$$\begin{aligned} \min_{y_i} \quad & y_i^T A_{i0} y_i + 2b_{i0}^T y_i + c_{i0} \\ \text{s.t.} \quad & y_i^T A_{i1} y_i + 2b_{i1}^T y_i + c_{i1} = 0, \end{aligned} \quad (3.12)$$

where $y_i = (t_i, x_i^T)^T \in \mathbb{R}^{p+1}$ and

$$\begin{aligned} A_{i0} &= \begin{pmatrix} 1 & 0_{1 \times p} \\ 0_{p \times 1} & \rho I_p \end{pmatrix}, & A_{i1} &= \begin{pmatrix} 0 & 0_{1 \times p} \\ 0_{p \times 1} & I_p \end{pmatrix} \\ b_{i0} &= [-d_i^2, \frac{1}{2}\mu_i^T - \rho z^T]^T, & b_{i1} &= [-\frac{1}{2}, -s_i^T]^T, \\ c_{i0} &= d_i^4 + \rho \|z\|^2 - \mu_i^T z, & c_{i1} &= \|s_i\|^2. \end{aligned}$$

We also define $q_i(y_i) = y_i^T A_{i1} y_i + 2b_{i1}^T y_i + c_{i1}$ referring to the constraint value. Note that this notation emphasizes that each node solves its own optimization problem.

The constraint qualifications have two requirements:

1. Feasible region is not empty,
2. $\min_{y_i^1} q_i(y_i^1) < 0 < \max_{y_i^2} q_i(y_i^2)$, $y_i^1, y_i^2 \in \mathbb{R}^{p+1}$.

Both are satisfied since we can always find $(t_i, x_i^T)^T$ points that fulfill the two conditions. From Theorem 3.2 [56], we can establish necessary and sufficient conditions for (3.12):

$$(A_{i0} + \lambda_i A_{i1}) y_i + b_{i0} + \lambda_i b_{i1} = 0, \quad (3.13a)$$

$$y_i^T A_{i1} y_i + 2b_{i1}^T y_i + c_{i1} = 0 \quad (3.13b)$$

$$A_{i0} + \lambda_i A_{i1} \succeq 0. \quad (3.13c)$$

where (3.13b) represents feasibility. As we have seen in chapter 1 regarding the semidefinite programming relaxation, we can write the dual problem of (3.12) and solve it using convex

optimization tools:

$$\begin{aligned} & \max_{\gamma_i, \lambda_i \in \mathbb{R}} \quad \gamma_i \\ & \text{s.t.} \quad \begin{pmatrix} A_{i0} + \lambda_i A_{i1} & b_{i0} + \lambda_i b_{i1} \\ b_{i0}^T + \lambda_i b_{i1}^T & c_{i0} + \lambda_i c_{i1} - \gamma_i \end{pmatrix} \succeq 0. \end{aligned} \quad (3.14)$$

We introduce the following operators which are solutions of (3.13a):

$$\hat{t}_i(\lambda_i) = \frac{\lambda_i}{2} + d_i^2, \quad \hat{x}_i(\lambda_i) = \frac{2(\rho z + \lambda_i s_i) - \mu_i}{2(\rho + \lambda_i)}. \quad (3.15)$$

The primal values are then recovered as $t_i^* = \hat{t}_i(\lambda_i^*)$ and $x_i^* = \hat{x}_i(\lambda_i^*)$, where λ_i^* corresponds to the solution of (3.14).

Efficient implementation

Dual problem (3.14) is general to solve quadratic problems in the form of (3.12). However, in this case, a more efficient method is possible. Rather than solving the SDP (3.14), we can reformulate (3.13) into

$$\text{solve}_{\lambda \in \mathbb{R}} \quad \phi_i(\lambda_i) = \|\hat{x}_i(\lambda_i) - s_i\|_2^2 - \hat{t}_i(\lambda_i) = 0 \quad (3.16a)$$

$$\text{s.t.} \quad \hat{t}_i(\lambda_i) = \frac{\lambda_i}{2} + d_i^2, \quad (3.16b)$$

$$\hat{x}_i(\lambda_i) = \frac{2(\rho z + \lambda_i s_i) - \mu_i}{2(\rho + \lambda_i)} \quad (3.16c)$$

$$A_{i0} + \lambda_i A_{i1} \succeq 0, \quad (3.16d)$$

From (3.16d), we necessarily have $\lambda_i \geq -\rho$. Because of [56, Th.5.2], $\phi_i(\lambda_i)$ is strictly decreasing in the previous interval and a bisection method to find λ_i^* can be devised. In particular, $\phi_i(-\rho) = +\infty$, so we need to find an upper bound $\bar{\lambda}_i$ so that $\phi_i(\bar{\lambda}_i) < 0$. This leads to the bisection method depicted in Algorithm 10.

The steps of algorithm 10 from 4–14 and 17 are performed in parallel, and only step 16 is coordinated with the other network nodes. The bisection limits are determined in steps 4–6, where $\phi_i(\lambda_i)$ changes sign. Then, from 7 to 13 the exact dual variable λ_i is calculated. Finally, steps 14–17 correspond to the ADMM updates.

We state the convergence results of algorithm 10:

Theorem 7. Assume that $\rho > 0$ is sufficiently large such that (3.13) can be satisfied at every iteration step. Then, algorithm 10 converges to a stationary solution of problem (3.8).

Algorithm 10 ADMM decomposition algorithm for the SRLS-TL problem

```

1: Initialize variables:  $t_i^0, x_i^0, \mu_i^0, z^0$ . Set  $k \leftarrow 0$ .
2: If a termination criterion is satisfied, STOP.
3: for  $i \in \mathcal{N}$  do in parallel
4:   Set  $\underline{\lambda}_i \leftarrow -\rho, \bar{\lambda}_i \leftarrow \rho$ 
5:   while  $\phi_i(\bar{\lambda}_i) \geq 0$  do  $\underline{\lambda}_i \leftarrow \bar{\lambda}_i, \bar{\lambda}_i \leftarrow 2\underline{\lambda}_i$ 
6:   end while
7:   Set  $\lambda_i \leftarrow (\bar{\lambda}_i + \underline{\lambda}_i)/2$ 
8:   while  $|\phi_i(\lambda_i)| \geq \varepsilon$  do
9:     Set  $\lambda_i \leftarrow (\bar{\lambda}_i + \underline{\lambda}_i)/2$ 
10:    if  $\phi_i(\lambda_i) \geq 0$  then set  $\underline{\lambda}_i \leftarrow \lambda_i$ 
11:    else set  $\bar{\lambda}_i \leftarrow \lambda_i$ 
12:    end if
13:  end while
14:  Set  $t_i^{k+1} \leftarrow \hat{t}_i(\lambda_i)$  and  $x_i^{k+1} \leftarrow \hat{x}_i(\lambda_i)$  for all  $i \in \mathcal{N}$ .
15: end for
16: Set  $z^{k+1} \leftarrow \frac{1}{N} \sum_{i \in \mathcal{N}} (x_i^{k+1} + \frac{1}{2\rho} \mu_i^k)$ .
17: Set  $\mu_i^{k+1} \leftarrow \mu_i^k + 2\rho(x_i^{k+1} - z)$  for all  $i \in \mathcal{N}$ .
18: Set  $k \leftarrow k + 1$ . Go to step 2.

```

Proof. See Appendix 3.A. □

3.2.2 NEXT decomposition

Distributed methods have rapidly evolved in the past years, where popular methods such as *consensus*, *diffusion* or *ADMM* arose as general frameworks to optimize problems in the form of sum of functions [57, 58, 77, 20, 68, 14]. These techniques have been used to solve convex problems. Analysis for nonconvex problems is limited, see for example [11, 47].

A new framework developed for sum of nonconvex functions was recently proposed in [27] which improves some aspects of previous approaches. It is named NEXT, after in-network nonconvex optimization, and it considers a distributed optimization problem for a sum of smooth nonconvex functions, plus a convex possibly nonsmooth regularizer. The framework admits time-varying connectivity as well as inexact computations, making it a very flexible and powerful tool for practical problems.

The proposal based on ADMM to solve the SRLS-TL problem we presented in section 3.2.1 is fast and improves previous results from other distributed implementations. ADMM presents however a non-negligible drawback regarding the requirements over the connectivity of a network: it is most efficient when the network is fully connected (every node is connected to every other node in the network) because, otherwise, it requires an average

consensus step over the primal and dual variables of the users, see step 16 in algorithm 10. This requirement over the network connectivity extends to every ADMM implementation, as it is inherent to the decomposition approach.

On the other hand, the solution presented in [27] over the SRLS-TL problem performs certain approximations of the original problem which may slow down the achievable convergence speed of the algorithm. In order to overcome, first, the connectivity limitations of ADMM and, secondly, a possibly slower convergence when using the NEXT framework with approximations, we propose another alternative that combines the decomposition techniques from section 3.2.1 with the NEXT framework. We name this proposal *NEXT-Exact*, in contrast to *NEXT-L* and *NEXT-PL* from [27], which fully linearizes or approximates with a quadratic function the objective of the SRLS-TL problem, respectively.

Nonconvex analysis of the quadratic problem using NEXT decomposition

The (simplified) NEXT framework is used for problems formed by a sum of functions, i.e.,

$$\min_{x \in \mathcal{X}} F(x) = \sum_{i \in \mathcal{N}} f_i(x), \quad (3.17)$$

with common variable x belonging to some set $\mathcal{X}$. The NEXT decomposition procedure can be understood as two distinct steps whose outcomes are reused as input for the next step and are repeated until convergence towards a stationary solution. The first step consists on a convexification of the local problem and computation of its solution. In particular, each agent (or node) tries to solve the following local problem

$$\min_{x_i \in \mathcal{X}} \tilde{f}_i(x_i) + \pi_i(x_i^k)^T (x_i - x_i^k) \quad (3.18)$$

where $\tilde{f}_i(x_i)$ corresponds to a strongly convex approximation of $f_i(x)$ over x_i and $\pi_i(x_i^k)$ corresponds to the gradient of $\sum_{j \neq i} f_j(x)$ evaluated at x_i^k , i.e.,

$$\pi_i(x_i^k) = \sum_{j \neq i} \nabla_{x_j} f_j(x_i^k). \quad (3.19)$$

However, since the real gradient $\pi_i(x_i^k)$ is not available locally at every node, the algorithm approximates this value with an averaged estimation denoted with $\tilde{\pi}_i(x_i^k)$. The second step consists precisely on the message exchange between the network nodes of their local optimized estimate and their local gradient. Then, this information is locally processed to obtain a global estimate of x_i^{k+1} and $\tilde{\pi}_i(x_i^{k+1})$.

The main difference with ADMM is that NEXT performs a single consensus message exchange, whereas ADMM requires a full average consensus stage. Unless the network is fully connected, the average consensus stage will require several message exchanges. Regarding *diffusion* techniques such as ATC-diffusion (adapt-then-combine), nodes receive the gradient information from their neighbors and compute an intermediate estimate. Then, neighbors share these intermediate values and they are combined for the final estimate [20]. At every iteration step, there are two distinct message exchanges between nodes, whereas NEXT performs a single message exchange with that same information. The difference is subtle, but in practice it results to halve the number of message exchanges. Finally, NEXT is also able to exploit the convex structure of $f_i(x)$, whereas diffusion or consensus linearizes the whole function $f_i(x)$ achieving in principle faster convergence speeds.

Back to the SRLS-TL problem, the individual local problem that each node has to solve is given by

$$\begin{aligned} \tilde{y}_i(x_i^k) = \arg \min_{t_i, x_i} \quad & (t_i - d_i^2)^2 + \tilde{\pi}_i(x_i^k)^T (x_i - x_i^k) + \rho \|x_i - x_i^k\|^2 \\ \text{s.t.} \quad & \|x_i - s_i\|^2 - t_i = 0, \end{aligned} \quad (3.20)$$

where we have introduced $\tilde{y}_i(x_i^k)$ as the operator that solves the nonconvex optimization problem for given x_i^k and returns its argument (t_i, x_i) . We have added the quadratic term $\rho \|x_i - x_i^k\|^2$ to enforce strong convexity in variable x_i . Finally, note that

$$\pi_i(x_i^k) = \sum_{j \neq i} 4(\|x_i^k - s_j\|^2 - d_j^2)(x_i^k - s_j), \quad (3.21)$$

where it is expected that $\tilde{\pi}_i(x_i^k) \rightarrow \pi_i(x_i^k)$ as k grows.

Same as in [section 3.2.1](#), we can transform problem (3.20) into a quadratic problem with a single equality constraint:

$$\begin{aligned} \min_{y_i} \quad & y_i^T A_{i0} y_i + 2b_{i0}^T y_i + c_{i0} \\ \text{s.t.} \quad & y_i^T A_{i1} y_i + 2b_{i1}^T y_i + c_{i1} = 0, \end{aligned} \quad (3.22)$$

where $y_i = (t_i, x_i^T)^T \in \mathbb{R}^{p+1}$ and

$$\begin{aligned} A_{i0} &= \begin{pmatrix} 1 & 0_{1 \times p} \\ 0_{p \times 1} & \rho I_p \end{pmatrix}, & A_{i1} &= \begin{pmatrix} 0 & 0_{1 \times p} \\ 0_{p \times 1} & I_p \end{pmatrix} \\ b_{i0} &= [-d_i^2, \frac{1}{2} \pi_i(x_i^k)^T - (\rho x_i^k)^T]^T, & b_{i1} &= [-\frac{1}{2}, -s_i^T]^T, \\ c_{i0} &= d_i^4 + \rho \|x_i^k\|^2 - \pi_i(x_i^k)^T x_i^k, & c_{i1} &= \|s_i\|^2. \end{aligned}$$

Function $\phi_i(\lambda_i)$ has the same expression as in equation 3.16a and y_i can be recovered with

$$y_i(\lambda_i, x_i^k) = -(A_{i0} + \lambda_i A_{i1})^{-1} (b_{i0} + \lambda_i b_{i1}) \quad (3.23)$$

The NEXT algorithm remains as in [27], which we reproduce in algorithm 11. An efficient implementation that solves problem (3.22) in step 4 is presented in algorithm 12. Note the following details:

1. $\tilde{y}_i(x_i^k)$ corresponds to the value given by (3.23) and has dimension $p + 1$.
2. Weights w_{ij} correspond to the network connectivity of the nodes. For simplicity we assumed that they are time invariant, but they could vary with time. In order to optimize the original problem (3.1), matrix W formed of w_{ij} must be double-stochastic, i.e., its rows and columns have to sum 1.
3. z_i^k is an estimate that lies in interval $(x_i^k, \tilde{y}_i(x_i^k)]$, according to $\alpha \in (0, 1]$.
4. ϕ_i^k corresponds to the averaged sum of local gradients particularized at their local estimates x_i^k , and it has dimension p .
5. $\tilde{\pi}_i(x_i^{k+1})$ is recovered locally from ϕ_i^k at every node i , and also has dimension p .

Algorithm 11 NEXT decomposition algorithm for the SRLS problem

- 1: Given y_i^0 and $\phi_i^0 = \nabla_x f_i(0)$, $\tilde{\pi}_i^0 = N\phi_i^0 - \nabla_x f_i(0)$ for all $i \in \mathcal{N}$. Set $k \leftarrow 0$.
 - 2: If y_i^k for all i satisfies a termination criterion, STOP.
 - 3: **for** $i \in \mathcal{N}$ **do in parallel**
 - 4: Solve $\tilde{y}_i(x_i^k)$ locally, as in algorithm 12.
 - 5: Update local variable $z_i^k = y_i^k + \alpha(\tilde{y}_i(x_i^k) - y_i^k)$
 - 6: **end for**
 - 7: Consensus update: exchange z_i^k and ϕ_i^k with neighbors.
 - 8: $y_i^{k+1} = (t_i^{k+1}, (x_i^{k+1})^T)^T = \sum_{j \in \mathcal{N}} w_{ij} z_j^k$.
 - 9: $\phi_i^{k+1} = \sum_{j \in \mathcal{N}} w_{ij} \phi_j^k + (\nabla f_i(x_i^{k+1}) - \nabla f_i(x_i^k))$.
 - 10: $\tilde{\pi}_i(x_i^{k+1}) = N\phi_i^{k+1} - \nabla f_i(x_i^{k+1})$
 - 11: Set $k \leftarrow k + 1$ and go to step 2.
-

Convergence

Algorithm 11 converges to a local minimum point of problem (3.1), provided the requirements established in Theorem 3 from [27] are met. The fact that the algorithm linearizes the local functions of other nodes prevents the algorithm to escape suboptimal stationary points,

Algorithm 12 Efficient implementation of problem (3.20) based on a bisection search.

```

1: Given variable  $x_i^k$  and  $\pi_i(x_i^k)$  for a specified  $i \in \mathcal{N}$ .
2: Set  $\underline{\lambda}_i \leftarrow -\rho$ ,  $\bar{\lambda}_i \leftarrow \rho$ 
3: while  $\phi_i(\bar{\lambda}_i) \geq 0$  do  $\underline{\lambda}_i \leftarrow \bar{\lambda}_i$ ,  $\bar{\lambda}_i \leftarrow 2\underline{\lambda}_i$ 
4: end while
5: Set  $\lambda_i \leftarrow (\bar{\lambda}_i + \underline{\lambda}_i)/2$ 
6: while  $|\phi(\lambda_i)| \geq \varepsilon$  do
7:   Set  $\lambda_i \leftarrow (\bar{\lambda}_i + \underline{\lambda}_i)/2$ 
8:   if  $\phi(\lambda_i) \geq 0$  then set  $\underline{\lambda}_i \leftarrow \lambda_i$ 
9:   else set  $\bar{\lambda}_i \leftarrow \lambda_i$ 
10:  end if
11: end while
12: return  $\tilde{y}_i(x_i^k) = y_i(\lambda_i, x_i^k)$ .

```

similar to the decomposition based on ADMM. For this reason, in [section 3.2.3](#) we propose an algorithm with convergence to the TL based on dual methods.

3.2.3 TL decomposition based on dual methods

At the beginning of [section 3.2](#) we discussed a centralized algorithm that solved the SRLS-TL problem optimally, originally proposed in [5]. To the best of our knowledge, there is no known algorithm in the literature that converges to an optimal solution in a distributed manner. For instance, every distributed algorithm we know uses descent techniques over the primal problem, as in [20, 27, 33], which do not guarantee that the necessary conditions for optimality are satisfied. And finally, other algorithms in the literature that study the TL problem are in general centralized, such as [40, 78, 53] and references therein, so there are not many results available on this matter. We find that we can propose an algorithm based on dual decomposition that converges to the optimal solution of the SRLS-TL problem under certain requirements. We discuss these requirements at the end of the section.

We start with a formulation inspired by the ADMM decomposition:

$$\begin{aligned}
\min_{\mathbf{t}, \mathbf{x}, \mathbf{z}} \quad & \sum_{i \in \mathcal{N}} (t_i - d_i^2)^2 + \mu_i^T (x_i - z) + \rho \|x_i - z\|^2 \\
\text{s.t.} \quad & \|x_i - s_i\|^2 - t_i = 0 \quad \forall i \in \mathcal{N}
\end{aligned} \tag{3.24}$$

with an extra complementarity problem

$$\mathbb{R} \ni \mu_i \perp x_i - z = 0 \quad \forall i \in \mathcal{N}, \tag{3.25}$$

where $\mathbf{x} = (x_i)_{i \in \mathcal{N}}$, $\mathbf{t} = (t_i)_{i \in \mathcal{N}}$, $t_i \in \mathbb{R}$ and $x_i, z \in \mathbb{R}^p$, and $a \perp b \Leftrightarrow a^T b = 0$. It is straightforward to see that (3.24)–(3.25) is an equivalent reformulation of (3.1) which introduces equality constraints and variables that must satisfy $x_1 = \dots = x_N = z$ at the optimal solution.

The idea behind this decomposition is to solve each of the problems at different stages, which will unfold to an algorithm of two loops. The inner loop will solve problem (3.24) for fixed $\boldsymbol{\mu} = (\mu_i)_{i \in \mathcal{N}}$, and after such solution the outer loop will take a gradient ascent step in order to solve the complementarity problem.

This two stage process is completely different from the ADMM approach, since ADMM performs three operations in a sequence, and does not have an inner loop. The implications are important because when ADMM solves problem (3.11) it does that for fixed z^k , and later updates z for fixed x_i^{k+1} . Because of the nonconvexity of the decomposition, ADMM is not solving (3.24) optimally when optimizing in separate steps, and possible convergence towards a stationary point may occur. The difficulty lies in solving (3.24) optimally in a distributed fashion.

Dual gradient ascent (centralized)

Before we introduce the fully distributed scheme, we discuss an intermediate approach that solves the QP using gradient descent on the dual problem associated to the constraints $x_i = z$.

We reformulate (3.24) to a standard quadratic form

$$\begin{aligned} \min_y \quad & y^T A_0 y + 2b_0^T y + c_0 \\ \text{s.t.} \quad & y^T A_i y + 2b_i^T y + c_i = 0 \quad \forall i \in \mathcal{N}, \end{aligned} \quad (3.26)$$

where $y = ((t_i)_{i \in \mathcal{N}}, (x_i^T)_{i \in \mathcal{N}}, z^T)^T \in \mathbb{R}^{N+Np+p}$ and

$$\begin{aligned} A_0 &= \begin{pmatrix} \mathbf{I}_N & \mathbf{0}_{N \times Np} & \mathbf{0}_{N \times p} \\ \mathbf{0}_{Np \times N} & \rho \mathbf{I}_{Np} & -\rho U \\ \mathbf{0}_{p \times N} & -\rho U^T & \rho N \mathbf{I}_p \end{pmatrix}, \\ b_0 &= [(-d_i^2)_{i \in \mathcal{N}}, \underbrace{-\mu_1^T, -\mu_2^T, \dots, \mu_N^T}_{1 \times Np}, \underbrace{-\sum_{i \in \mathcal{N}} \mu_i^T}_{1 \times p}]^T \\ c_0 &= \sum_{i \in \mathcal{N}} d_i^4, \\ A_i &= \text{diag}[\mathbf{0}_{N \times N}, \underbrace{\mathbf{0}_{p \times p}, \dots, \tilde{A}_i, \dots, \mathbf{0}_{p \times p}}_{p \times Np}, \mathbf{0}_{p \times p}], \\ b_i &= [\underbrace{0, \dots, -\frac{1}{2}, \dots, 0}_{1 \times N}, \underbrace{0, \dots, -s_i^T, \dots, 0}_{1 \times Np}, 0_{1 \times p}]^T, \\ c_i &= \|s_i\|^2. \end{aligned}$$

We introduced $U = \text{col}[(\mathbf{I}_{p \times p})_{i \in \mathcal{N}}]$, which is matrix of dimension $Np \times p$ of stacked identity matrices in column form, $\tilde{A}_i = \mathbf{I}_p$ for all $i \in \mathcal{N}$. Note that A_i is a square matrix with dimension $N + Np + p$ and has nonzero element $\tilde{A}_i$ corresponding to the position of x_i .

The idea is to solve (3.24) optimally as a first step, and then, take a descent step in the dual problem. In particular, solve (3.26) for fixed $\boldsymbol{\mu}^k$, and then update coefficients $\boldsymbol{\mu}^{k+1}$ according to the argument values:

$$\mu_i^{k+1} = \mu_i^k + \alpha(x_i^{k+1,q} - z^{k+1,q}) \quad \forall i \in \mathcal{N}. \quad (3.27)$$

The steps are described in [algorithm 13](#).

Algorithm 13 Dual gradient ascent of the TL problem.

- 1: Initialize $\boldsymbol{\mu}^0, \mathbf{x}^0, z^0$ and step sizes $\{\alpha^k\}$. Set $k \leftarrow 0$.
 - 2: **while** $\boldsymbol{\mu}^k$ is not a stationary point **do**
 - 3: Set $y^{k+1} \leftarrow$ solution of problem (3.26).
 - 4: Identify: $(\mathbf{x}^{k+1}, z^{k+1}) \leftarrow y^{k+1}$
 - 5: Set $\mu_i^{k+1} \leftarrow \mu_i^k + \alpha^k(x_i^k - z^k) \quad \forall i \in \mathcal{N}$.
 - 6: Set $k \leftarrow k + 1$.
 - 7: **end while**
-

Convergence properties of [algorithm 13](#) are essentially those from dual methods, as explained in Section 6.2 and 6.3 of [9], or in [16]. In practice, the algorithm is very stable and converges very fast, i.e., nodes reach a consensus after very few iterations. We will see this performance in [section 3.4.1](#). The next section will analyze an extension to achieve a fully distributed scheme.

Distributed dual ascent based on NOVA decomposition

We describe a distributed scheme to solve (3.24) for fixed $\boldsymbol{\mu}^k$. Optimality can be guaranteed using Theorem 3.1 from [52], which we reproduced in [Theorem 1 \(chapter 2\)](#). Assuming the $\mathcal{S}$ -property holds, a candidate solution must satisfy the KKT conditions plus

$$A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i \succeq 0, \quad (3.28)$$

which establishes a constraint in the dual domain and is the reason why descent algorithms of the primal problem may converge to local optima. We can introduce a dual variable Z and solve

$$0 \preceq Z \perp A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i \succeq 0 \quad (3.29)$$

as a dual problem. In particular, the update rule that arises is as follows:

$$Z^{q+1} = \left[Z^q - \alpha^q (A_0 + \sum_{i \in \mathcal{N}} \lambda_i^r A_i) \right]_+, \quad (3.30)$$

where q refers to an outer loop counting index and r to an inner loop index.

Following [section 2.5.3](#), we can formulate a global Lagrangian and group terms together as follows:

$$\begin{aligned} L(\mathbf{x}, z, \lambda, Z) & \quad (3.31) \\ &= \sum_{i \in \mathcal{N}} (t_i - d_i^2)^2 + \mu_i^T (x_i - z) + \rho \|x_i - z\|^2 + \lambda_i (\|x_i - s_i\|^2 - t_i) + \text{tr}[Z(A_0 + \sum_{i \in \mathcal{N}} \lambda_i A_i)] \\ &= \sum_{i \in \mathcal{N}} (t_i - d_i^2)^2 + \mu_i^T (x_i - z) + \rho \|x_i - z\|^2 + \lambda_i (\|x_i - s_i\|^2 - t_i + \text{tr}[ZA_i]) + \text{tr}[ZA_0]. \end{aligned}$$

From the Lagrangian formulation, we can introduce the following complementarity problems

$$-\rho \leq \lambda_i \perp \|x_i - s_i\|^2 - t_i + \text{tr}[ZA_i], \quad (3.32)$$

and develop a distributed scheme. Using the NOVA decomposition procedure from [\[69\]](#), we decouple (3.31) into subproblems that can be solved locally. We get the following:

$$\begin{aligned} \min_{t_i, x_i} \quad & (t_i - d_i^2)^2 + (\mu_i^k)^T (x_i - z^r) + \rho \|x_i - z^r\|^2 + \lambda_i (\|x_i - s_i\|^2 - t_i + \text{tr}[ZA_i]) \\ \text{s.t.} \quad & -\rho_i \leq \lambda_i \perp \|x_i - s_i\|^2 - t_i + \text{tr}[ZA_i], \end{aligned} \quad (3.33)$$

which is a minimization problem with one complementarity constraint. For central variable z we have

$$\min_z \sum_{i \in \mathcal{N}} (\mu_i^k)^T (x_i^r - z) + \rho \|x_i^r - z\|^2, \quad (3.34)$$

where (3.34) has closed form solution

$$z^{r+1} = \frac{1}{N} \sum_{i \in \mathcal{N}} \left(x_i^r + \frac{1}{2\rho} \mu_i^k \right). \quad (3.35)$$

We introduced x_i^r and z^r to indicate the corresponding values of x_i and z on iteration r .

Problem (3.33) is nonconvex and presents strong duality (it has only one constraint). It can be solved via its dual problem, where $\lambda_i \geq -\rho$. We denote with $\hat{\lambda}_i$ the solution of the dual problem, i.e.,

$$\hat{\lambda}_i(z^r, \mu_i^k) = \arg \max_{\lambda_i \geq -\rho} \Upsilon_i(\lambda_i, z^r; \mu_i^k). \quad (3.36)$$

The dual problem is solved via the bisection method as we have been seeing in previous sections. We can recover t_i^{r+1} and x_i^{r+1} as follows:

$$t_i^{r+1} = d_i^2 + \frac{\lambda_i^{r+1}}{2} \quad (3.37)$$

$$x_i^{r+1} = \frac{2\rho z^r + 2\lambda_i^{r+1}s_i}{2(\rho + \lambda_i^{r+1})}. \quad (3.38)$$

We present [algorithm 14](#) that solves the TL problem in a distributed fashion. It is composed of three nested loops. The loop in the outer procedure seeks consensus over the local variables. The other two loops attempt to solve problem (3.24) using parallel steps.

Variable Z helps to distinguish when the stationary solution is optimal or not. If $Z = 0$ then, the framework has converged to a global solution and the procedure can stop. If $Z \neq 0$, the inner loop has converged to a suboptimal stationary solution and the subproblems are modified accordingly until the stationary solution falls into the optimal region. This is repeated until a consensus in the source position is achieved. Overall, the algorithm is competitive as we will see in [section 3.4](#).

Algorithm 14 Distributed dual ascent algorithm

- 1: Initialize $\boldsymbol{\mu}^0$, Z^0 , $\mathbf{x}^0$ and z^0 . Set $k \leftarrow 0$.
 - 2: **while** $\boldsymbol{\mu}$ is not a stationary point **do**
 - 3: Initialize $Z^0 \leftarrow Z^q$. Set $q \leftarrow 0$.
 - 4: **while** Z^q is not a stationary point **do**
 - 5: Initialize $\mathbf{x}^0 \leftarrow \mathbf{x}^r$, $z^0 \leftarrow z^r$. Set $r \leftarrow 0$.
 - 6: **while** $\mathbf{x}^r$ and z^r are not stationary points **do**
 - 7: Set $\lambda_i^{r+1} \leftarrow \hat{\lambda}_i(x_i^r, \mu_i^k)$ for all $i \in \mathcal{N}$ **in parallel**.
 - 8: Set $t_i^{r+1} \leftarrow d_i^2 + \frac{\lambda_i^{r+1}}{2}$.
 - 9: Set $x_i^{r+1} \leftarrow \frac{2\rho z^r + 2\lambda_i^{r+1}s_i}{2(\rho + \lambda_i^{r+1})}$.
 - 10: Set $z^{r+1} \leftarrow \frac{1}{N} \sum_{i \in \mathcal{N}} \left(x_i^r + \frac{1}{2\rho} \mu_i^k \right)$.
 - 11: Set $r \leftarrow r + 1$.
 - 12: **end while**
 - 13: Set $Z^{q+1} = [Z^q - \alpha^k (A_0 + \sum_i \lambda_i^r A_i)]_+$.
 - 14: Set $q \leftarrow q + 1$.
 - 15: **end while**
 - 16: Set $\mu_i^{k+1} \leftarrow \mu_i^k + \alpha^k (x_i^r - z^r) \quad \forall i \in \mathcal{N}$.
 - 17: Set $k \leftarrow k + 1$.
 - 18: **end while**
-

Convergence discussion

Convergence of [algorithm 13](#) is guaranteed if three conditions are satisfied.

1. Step size α^k is sufficiently small. This requirement is related to the gradient ascent method and is common to any optimization problem.
2. Parameter ρ is sufficiently large such that $A_0 + \sum_{i \in \mathcal{N}} \lambda_i^* A_i \succeq 0$ where λ_i^* correspond to the dual variables that solve problem (3.24) and (3.25) simultaneously. A large value of ρ allows to use negative λ_i .
3. The $\mathcal{S}$ -property is satisfied in problem (3.24). In the next proposition we provide a sufficient and necessary condition to verify if it is satisfied. In the noiseless case it always holds.

Proposition 5. *Problem (3.24) has the $\mathcal{S}$ -property in the noiseless case, i.e., $d_i = \|x^* - s_i\|$ for every $i \in \mathcal{N}$, where x^* corresponds to the true location of the target. A necessary and sufficient condition for the noisy problem to hold the $\mathcal{S}$ -property is given by the following condition:*

$$N - \left(\sum_{i \in \mathcal{N}} \frac{\rho}{\lambda_i^* + \rho} \right) \geq 0, \quad (3.39)$$

where $\lambda_i^* = (\|x^* - s_i\|^2 - d_i^2)$ corresponds to the dual variables of the candidate solution x^* .

Proof. See [appendix 3.A](#). □

From our analysis we conclude that we cannot guarantee convergence of [algorithm 14](#) because we do not know if condition (3.39) is satisfied beforehand. This seems to be a common problem in many localization algorithms that rely on semidefinite relaxations, such as in [12, 73, 33]. In practice, however, it would make sense to use a suboptimal algorithm for fast convergence and then check the optimality condition. Afterwards, we can use algorithms based on dual decomposition accordingly.

Assuming that the $\mathcal{S}$ -property holds in problem (3.24), the gradient ascent method given by (3.27) escapes local minima because of weak duality. From duality theory we have that if $\Upsilon(\boldsymbol{\lambda}, \boldsymbol{\mu})$ represents the dual problem of (3.24) and (3.25) jointly, we have that

$$\Upsilon(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq \sum_{i \in \mathcal{N}} (\|x - s_i\|^2 - d_i^2)^2. \quad (3.40)$$

If we have that $x_i^* = z^*$ for every i is a candidate solution of problems (3.24) and (3.25), it must satisfy (3.40) if the solution of (3.24) is optimal. Otherwise we reach a contradiction.

This implies that even if we initialize [algorithm 14](#) at a suboptimal stationary solution of problem (3.1), the algorithm still converges to the global minimum. To the best of our knowledge, this trait is unique from our formulation, which is a consequence of working in the dual domain.

3.3 Sensor Network Localization

The sensor network localization problem (SNL) consists in finding the location of a set of nodes given a collection of measurements between the nodes that form the network. In this setting we assume there are anchor nodes who know their position exactly, and nodes without such knowledge. We denote the set of anchor nodes with $\mathcal{N}^a$ and the set of nodes with unknown location with $\mathcal{N}^u$. The total number of nodes with unknown location is $N = |\mathcal{N}^u|$. Furthermore, we also use $\mathcal{N}_i^a$ and $\mathcal{N}_i^u$ to denote the neighbors of node i who belong to the set of anchors and nodes with unknown location, respectively. In particular, the neighbors of node i correspond to the nodes that have a distance measurement and can communicate between each other. As in the TL problem, the nodes lie in an euclidean space of dimension p , typically 2 or 3. In this section we will consider the following estimator

$$(\text{SRLS-SNL}) : \min_{\mathbf{x} \in \mathbb{R}^{Np}} \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^a} (\|x_i - s_j\|^2 - d_{ij}^2)^2 + \sum_{j \in \mathcal{N}_i^u} (\|x_i - x_j\|^2 - d_{ij}^2)^2, \quad (3.41)$$

where s_j represents the location of anchor node j , $\mathbf{x} = (x_i)_{i \in \mathcal{N}^u}$ and $x_i \in \mathbb{R}^p$.

The SNL problem has been studied extensively, but it remains a challenging problem because of its nonconvexity and hardness. In particular, reference [1] shows that the problem is NP-hard in its most general formulation. This result was surprising after the successful evidence of several algorithms using semidefinite relaxation [12] or convex formulations [30] with promising results. Later, articles [74] and [32] showed independently that problems whose graphs were *uniquely localizable* could be solved in polynomial time, giving some kind of explanation to the good performance that many algorithms were getting in random scenarios and average connectivity. For reference, a *uniquely localizable* graph is one that first, has a unique realization in $\mathbb{R}^p$, and secondly, it does not have another realization whose affine span is $\mathbb{R}^h$, where $h > p$. However, this result only affects the noiseless SNL problem, and to the best of our knowledge, polynomial complexity for the noisy problem has not yet been established.

Semidefinite programming relaxations have been proposed in a variety of scenarios. For instance, [12] have such formulation for the noiseless model, extends it to ML estimator in case of Gaussian noise, and proposes a distributed version for large networks by means of

creating ad-hoc clusters. Another work that uses this approach is [54], which proposes a setting with uncertainties in the anchor locations and propagation speed. Other works such as [83] and [64] focus on further relaxing the centralized problem for easier computation and possible decoupled approach.

Distributed formulations may focus on different approximations, such as convex relaxations directly derived from the original problem, optimization techniques, and semidefinite relaxations. For instance, reference [76] replaces the equality constraints with inequalities, reducing the initial complexity to that of a convex problem. A different approach is followed in reference [23], who obtains a parallel and scalable scheme using a majorization-minimization technique over the nonconvex problem and little computational complexity. The work from [73] achieves a distributed scheme starting from a general ML estimator, then relaxing the problem to a semidefinite program and further decoupling the previous formulation with a second relaxation technique originally proposed in [83]. They end up with a convex problem that can be solved using an ADMM scheme.

And finally, more recent approaches such as [75] propose a fast, simple and asynchronous algorithm with favorable performance compared to previous approaches. Reference [59] analyzes different estimators in the same line as [5] and proposes a weighted SRLS-SNL estimator that approximates very efficiently the ML estimator in practice. However, they propose a Gauss-Seidel iterative scheme without convergence guarantees and in principle cannot run in parallel, which is an important property in distributed networks. Last, recent publication [72] considers a semidefinite approach with a robust formulation.

In this section we present two contributions. The first contribution is a distributed scheme different from those already in the literature. Our proposal is scalable, fast, uses minimum communication between the nodes and has convergence guarantees towards a stationary point. In order to do so, we use the framework developed in [70]. Furthermore, we test the algorithm in networks with minimal connectivity and still giving excellent performance with random initializations.

Our second contribution consists in solving the semidefinite program in a distributed fashion. It has the important property that if strong duality holds in the SNL problem, the algorithm converges to the optimal solution and avoids suboptimal solutions. In order to do so, we first propose a centralized algorithm and then an equivalent decoupled version. In this case, we follow the algorithmic proposals discussed in [chapter 2](#). [Table 3.2](#) presents a summary of our algorithmic contributions and a distributed scheme from the literature that we compare with.

	Convergence	Description
FLEXA poly2	suboptimal	algorithm 15 polynomial approx. order 2.
FLEXA poly4	suboptimal	algorithm 15 polynomial approx. order 4.
Centralized	optimal (if the $\mathcal{S}$ -property is satisfied)	described in section 3.3.2 , based on SDP.
Dual ascent	optimal (if the $\mathcal{S}$ -property is satisfied)	algorithm 16 .
Distributed dual ascent	optimal (if the $\mathcal{S}$ -property is satisfied)	algorithm 17 .
Costa et al.	suboptimal	proposed in [23] .

Table 3.2 Summary of SNL algorithms

Finally, we remark that the estimator presented in (3.41) problem can be easily extended with appropriate weights that would approximate the ML estimator as explained in [59], but we omit the explicit treatment to ease the presentation and simplify the notation.

3.3.1 FLEXA decomposition

Problem (3.41) is nonconvex and can be solved using algorithms of the family of successive convex approximations, such as the one proposed in [69] named FLEXA. The methodology consists of an iterative procedure where a convex function is used to approximate the non-convex original objective in a local region around a temporary fixed point. The surrogate function retains the convex parts and linearizes the non-convex ones. The convexification is performed around a temporary pivotal point, which is updated in subsequent iteration steps. Different transformations may be used for the problem at hand, adding some flexibility to the algorithmic complexity and convergence speed.

The objective function can be expanded into

$$\begin{aligned}
 f(\mathbf{x}) = & \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^a} \|x_i - s_j\|^4 + d_{ij}^4 \underbrace{- 2d_{ij}^2 \|x_i - s_j\|^2}_{\text{nonconvex}} \\
 & + \sum_{j \in \mathcal{N}_i^u} \|x_i - x_j\|^4 + d_{ij}^4 \underbrace{- 2d_{ij}^2 \|x_i - x_j\|^2}_{\text{nonconvex}},
 \end{aligned} \tag{3.42}$$

which allows two natural function approximations.

One possibility to obtain a convex surrogate is to linearize the concave contribution. Since it is a difference of convex functions, we can use a first order Taylor expansion of the

negative quadratic terms. With this idea in mind, each node $i \in \mathcal{N}_u$ can use as surrogate the following approximating function:

$$\begin{aligned} \tilde{f}_i^1(x_i, \mathbf{x}^k) = & \sum_{j \in \mathcal{N}_i^a} \|x_i - s_j\|^4 + d_{ij}^4 - 4d_{ij}^2(x_i^k - s_j)^T(x_i - x_i^k) \\ & + \sum_{j \in \mathcal{N}_i^u} \|x_i - x_j^k\|^4 + d_{ij}^4 - 4d_{ij}^2(x_i^k - x_j^k)^T(x_i - x_i^k) + \rho \|x_i - x_i^k\|^2, \end{aligned} \quad (3.43)$$

where $\mathbf{x}^k = (x_i^k)_{i \in \mathcal{N}_u}$ is the pivotal point that helps to decouple the problem. Value $\mathbf{x}^k$ is updated at different iteration steps, and $\rho > 0$ enforces strong convexity.

A second possibility that further simplifies the computational complexity of solving (3.43) is to use a second order Taylor expansion of problem (3.42) of the convex terms. This would give the following surrogate function:

$$\begin{aligned} \tilde{f}_i^2(x_i, \mathbf{x}^k) = & \sum_{j \in \mathcal{N}_i^a} (\nabla_{x_i} \|x_i - s_j^k\|^4|_{x_i=x_i^k})^T(x_i - x_i^k) + (x_i - x_i^k)^T(\nabla_{x_i}^2 \|x_i - s_j^k\|^4|_{x_i=x_i^k})(x_i - x_i^k) \\ & + \sum_{j \in \mathcal{N}_i^u} (\nabla_{x_i} \|x_i - x_j^k\|^4|_{x_i=x_i^k})^T(x_i - x_i^k) + (x_i - x_i^k)^T(\nabla_{x_i}^2 \|x_i - x_j^k\|^4|_{x_i=x_i^k})(x_i - x_i^k) \\ & + \sum_{j \in \mathcal{N}_i^a} d_{ij}^4 - 4d_{ij}^2(x_i^k - s_j)^T(x_i - x_i^k) + \sum_{j \in \mathcal{N}_i^u} d_{ij}^4 - 4d_{ij}^2(x_i^k - x_j^k)^T(x_i - x_i^k) + \rho \|x_i - x_i^k\|^2, \end{aligned} \quad (3.44)$$

where for every $i \in \mathcal{N}_u$ we have

$$\nabla_{x_i} \|x_i - s_j^k\|^4|_{x_i=x_i^k} = 4\|x_i - s_j^k\|^2(x_i^k - s_j^k) \quad (3.45)$$

$$\nabla_{x_i}^2 \|x_i - s_j^k\|^4|_{x_i=x_i^k} = 8(x_i^k - s_j^k)(x_i^k - s_j^k)^T + 4\|x_i^k - s_j^k\|^2 \mathbf{I}_p \quad (3.46)$$

$$\nabla_{x_i} \|x_i - x_j^k\|^4|_{x_i=x_i^k} = 4\|x_i - x_j^k\|^2(x_i^k - x_j^k) \quad (3.47)$$

$$\nabla_{x_i}^2 \|x_i - x_j^k\|^4|_{x_i=x_i^k} = 8(x_i^k - x_j^k)(x_i^k - x_j^k)^T + 4\|x_i^k - x_j^k\|^2 \mathbf{I}_p. \quad (3.48)$$

Although $\tilde{f}_i^2(x_i, \mathbf{x}^k)$ has a long expression, it is a simple convex quadratic function that aggregates the information of every neighboring node.

The FLEXA framework developed in [69] requires that every node solves a local problem of the following form:

$$\hat{x}_i^m(\mathbf{x}^k) = \arg \min_{x_i \in \mathbb{R}^p} \tilde{f}_i^m(x_i, \mathbf{x}^k) \quad (3.49)$$

where m specifies whether we are using surrogate function (3.43) or (3.44). We made it explicit that (3.49) is minimized only in x_i because the algorithm is solved in a distributed

manner. Note also that the nodes only need to exchange their own estimates with their neighbors in order to obtain $\tilde{f}_i^m(x_i, \mathbf{x}^k)$. Finally, note that $\hat{x}_i^m(\mathbf{x}^k)$ is an unconstrained strongly convex problem (for $\rho > 0$) and has a unique optimal solution.

In the case of $\hat{x}_i^1(\mathbf{x}^k)$, the problem can be solved using numerical methods that solve the system of equations that results from

$$\nabla_{x_i} \tilde{f}_i^1(x_i, \mathbf{x}^k) = 0. \quad (3.50)$$

Regarding surrogate problem $\hat{x}_i^2(\mathbf{x}^k)$, it has a closed form solution. If we rewrite

$$\tilde{f}_i^2(x_i, \mathbf{x}^k) = x_i^T A_{0i} x_i + 2b_{0i}^T x_i + c_{0i} \quad (3.51)$$

the solution becomes

$$\hat{x}_i^2(\mathbf{x}^k) = -A_{0i}^{-1} b_{0i} \quad (3.52)$$

where

$$\begin{aligned} A_{0i} &= \sum_{j \in \mathcal{N}_i^a} (\nabla_{x_i}^2 \|x_i - s_j^k\|^4|_{x_i=x_i^k}) + \sum_{j \in \mathcal{N}_i^u} (\nabla_{x_i}^2 \|x_i - x_j^k\|^4|_{x_i=x_i^k}) + \rho \mathbf{I}_p \\ b_{0i} &= \sum_{j \in \mathcal{N}_i^a} \frac{1}{2} \nabla_{x_i} \|x_i - s_j^k\|^4|_{x_i=x_i^k} - 2d_{ij}^2(x_i^k - s_j) - (\nabla_{x_i}^2 \|x_i - s_j^k\|^4|_{x_i=x_i^k}) x_i^k \\ &\quad + \sum_{j \in \mathcal{N}_i^u} \frac{1}{2} \nabla_{x_i} \|x_i - x_j^k\|^4|_{x_i=x_i^k} - 2d_{ij}^2(x_i^k - x_j^k) - (\nabla_{x_i}^2 \|x_i - x_j^k\|^4|_{x_i=x_i^k}) x_i^k - 2\rho x_i^k. \end{aligned}$$

Note that the inverse of A_{0i} is straightforward for $p \in \{2, 3\}$.

We proposed two surrogate functions to give some flexibility to the solution framework. Solving $\hat{x}_i^1(\mathbf{x}^k)$ probably reduces the amount of required packet exchanges because it exploits better the information available at the node, but at the cost of some more computational power. Problem $\hat{x}_i^2(\mathbf{x}^k)$ is very simple computationally speaking, it only uses simple operations, but may require some more iterations to converge. Depending on the network and node characteristics, each method may stand out above the other for these reasons.

The complete set of instructions to solve problem (3.41) is described in [algorithm 15](#). Note that as analyzed in Theorem 3 in [69], the algorithm converges to a stationary point of the non-convex problem (3.41) for $\alpha_k \in (0, 1]$ and $\rho \geq L_{\nabla f}$, where $L_{\nabla f}$ corresponds to the Lipschitz constant of the SNL problem. The feasibility region can be assumed to be compact in some region in $\mathbb{R}^p$ order to find this constant. In practice the algorithm is robust for many values of ρ . Finally, as analyzed in [18], it admits asynchronous updates when some nodes have an out of date pivotal point $\mathbf{x}^k$.

Algorithm 15 Distributed SNL algorithm using FLEXA decomposition.

- 1: Initialize $\mathbf{x}^0 = (x_i)_{i \in \mathcal{N}^u} \in \mathbb{R}^{pN}$ and α_k . Set $\rho \geq 0$. Set $k \leftarrow 0$.
 - 2: **while** $\mathbf{x}^k$ is not a stationary point **do**
 - 3: For all $i \in \mathcal{N}^u$, solve $\hat{x}_i^m(x^k)$ **in parallel**.
 - 4: $x_i^{k+1} = x_i^k + \alpha_k(\hat{x}_i^m(x^k) - x_i^k)$
 - 5: Exchange x_i^{k+1} with neighbors $j \in \mathcal{N}_i^u$, $\forall i \in \mathcal{N}^u$.
 - 6: Set $k \leftarrow k + 1$.
 - 7: **end while**
-

3.3.2 Centralized proposal based on semidefinite relaxation

We consider a semidefinite formulation different from the ones discussed in [12, 74, 54, 73, 72] with the aim of obtaining a distributed scheme in a subsequent development, which we will describe in section 3.3.3. First, we transform the polynomial of degree four (3.41) into a standard QPQC problem. Problem (3.41) becomes

$$\begin{aligned}
 \min_{\mathbf{t} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^{Np}} \quad & \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} (t_{ij} - d_{ij}^2)^2 \\
 \text{s.t.} \quad & \|x_i - s_j\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\
 & \|x_i - x_j\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u,
 \end{aligned} \tag{3.53}$$

where $\mathbf{t} = (t_{ij})_{i \in \mathcal{N}^u, j \in (\mathcal{N}_i^u \cup \mathcal{N}_i^a)}$ is a vector column of dimension $n = \sum_{i \in \mathcal{N}^u} |\mathcal{N}_i^u| + |\mathcal{N}_i^a|$ and $\mathbf{x} = (x_i)_{i \in \mathcal{N}^u}$ has dimension Np . Parameter n corresponds to the number of connections between nodes and, also, to the number of total constraints.

We transform (3.53) into a standard QPQC form:

$$\begin{aligned}
 \min_{y \in \mathbb{R}^{n+Np}} \quad & y^T A_0 y + 2b_0^T y + c_0 \\
 \text{s.t.} \quad & y^T A_{ij} y + 2b_{ij}^T y + c_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a
 \end{aligned} \tag{3.54}$$

where $y = (\mathbf{t}^T, \mathbf{x}^T)^T$ is a column vector of dimension $n + Np$. Other terms take the following values:

$$\begin{aligned}
A_0 &= \begin{pmatrix} \mathbf{I}_n & 0_{n \times Np} \\ 0_{Np \times n} & 0_{Np \times Np} \end{pmatrix}, & b_0 &= \left(\underbrace{(-d_{1j}^2)_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}, \dots, (-d_{Nj}^2)_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}}_{1 \times n}, \underbrace{0, \dots, 0}_{1 \times Np} \right)^T \\
A_{ij} &= \begin{pmatrix} 0_{n \times n} & 0_{n \times Np} \\ 0_{Np \times n} & U \end{pmatrix}, & b_{ij} &= \left(\underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times n}, \underbrace{0, \dots, 0}_{1 \times Np} \right)^T \quad \forall i \in \mathcal{N}_i^u, j \in \mathcal{N}_i^u \\
A_{ij} &= \begin{pmatrix} 0_{n \times n} & 0_{n \times Np} \\ 0_{Np \times n} & \text{diag}[\underbrace{0_{p \times p}, \dots, 0_{p \times p}, \mathbf{I}_p, 0_{p \times p}, \dots, 0_{p \times p}}_{p \times N}] \end{pmatrix} & & \forall i \in \mathcal{N}_i^u, j \in \mathcal{N}_i^a \\
b_{ij} &= \left(\underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times n}, \underbrace{0, \dots, 0, -s_j^T, 0, \dots, 0}_{1 \times Np} \right)^T & & \forall i \in \mathcal{N}_i^u, j \in \mathcal{N}_i^a \\
c_0 &= \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} d_{ij}^4, & c_{ij} &= \|s_j\|^2 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a, \quad c_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u
\end{aligned}$$

and U is a square matrix of dimension Np that couples variables x_i and x_j when $j \in \mathcal{N}_i^u$:

$$U = \begin{pmatrix} 0_{p \times p} & \cdots & \cdots & \cdots & \cdots & 0_{p \times p} \\ 0_{p \times p} & \mathbf{I}_p & 0_{p \times p} & \cdots & -\mathbf{I}_p & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_{p \times p} & -\mathbf{I}_p & 0_{p \times p} & \cdots & \mathbf{I}_p & \cdots \\ 0_{p \times p} & \underbrace{\cdots}_{i\text{'th column}} & \cdots & \cdots & \underbrace{\cdots}_{j\text{'th column}} & 0_{p \times p} \end{pmatrix}. \quad (3.55)$$

Once we have the QP in standard form, we can solve the Lagrangian dual problem, i.e.

$$\begin{aligned}
& \max_{\gamma \in \mathbb{R}, \boldsymbol{\lambda} \in \mathbb{R}^n} \gamma \\
& \text{s.t.} \quad \begin{pmatrix} A_0 & b_0 \\ b_0^T & c_0 - \gamma \end{pmatrix} + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij} \begin{pmatrix} A_{ij} & b_{ij} \\ b_{ij}^T & c_{ij} \end{pmatrix} \succeq 0
\end{aligned} \quad (3.56)$$

using semidefinite optimization tools, such as [42]. We introduced $\boldsymbol{\lambda} = (\lambda_{ij})_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}$.

If strong duality holds in the SNL problem, we can recover the solution with

$$y^* \in - \left(A_0 + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij}^* A_{ij} \right)^\dagger \left(b_0 + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij}^* b_{ij} \right) \quad (3.57)$$

$$+ \text{null} \left[A_0 + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij}^* A_{ij} \right], \quad (3.58)$$

such that y^* satisfies

$$\mathbb{R} \ni \lambda_{ij} \perp y^T A_{ij} y + 2b_{ij}^T y + c_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a. \quad (3.59)$$

Values $\mathbf{t}^*$ and $\mathbf{x}^*$ are directly obtained from y^* . Recall that λ_{ij}^* correspond to the optimal solution of convex problem (3.56).

A simple test to verify if $\mathbf{x}^*$ is an optimal solution is to check if strong duality holds, i.e.,

$$\gamma^* \stackrel{?}{=} (y^*)^T A_0 y^* + 2b_0^T y^* + c_0 \quad (3.60)$$

is fulfilled. As we already mentioned in the introduction, [1] showed that the SNL problem is NP-hard in its most general form. However, we also discussed that, in practice, semidefinite formulations have excellent behavior in most network problems. In all of our simulations we found that problem (3.53) had strong duality too, motivating the usefulness of our formulation.

3.3.3 Distributed proposal based on semidefinite relaxation

The previous section discusses a formulation that is completely centralized and cannot be easily distributed. Instigated by that formulation, we extend the scheme introducing slack variables that decouple the constraints. In essence, each node not only estimates its unknown location, but also optimizes the unknown location of its neighbors. In subsequent steps, convergence over all location estimates is pursued. If strong duality holds on the intermediate problems, optimality of the SRLS-SNL problem is achieved.

Problem (3.41) has the following equivalent formulation

$$\begin{aligned} \min_{\mathbf{t}, \mathbf{x}, \mathbf{z}} \quad & \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} (t_{ij} - d_{ij}^2)^2 \\ \text{s.t.} \quad & \|x_i - s_j\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\ & \|x_i - z_{ji}\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \\ & x_j = z_{ji} \quad \forall i \in \mathcal{N}_i^u, j \in \mathcal{N}_i^u \\ & \mathbf{t} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^{Np}, \mathbf{z} \in \mathbb{R}^{mp}, \end{aligned} \quad (3.61)$$

where $\mathbf{t} = (t_{ij})_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}$, $\mathbf{x} = (x_i)_{i \in \mathcal{N}_u}$ and $\mathbf{z} = (z_{ji})_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u}$. Also, $t_{ij} \in \mathbb{R}$, $x_i \in \mathbb{R}^p$ and $z_{ji} \in \mathbb{R}^p$. Variable z_{ji} refers to the estimate that node i has over the position of node j , whenever $j \in \mathcal{N}_i^u$. Regarding the corresponding dimensions, $n = \sum_{i \in \mathcal{N}^u} |\mathcal{N}_i^u| + |\mathcal{N}_i^a|$ indicates the total number of connections in the network graph. Similarly, $m = \sum_{i \in \mathcal{N}_i^u} |\mathcal{N}_i^u|$ corresponds to the total number of connections between neighbors with unknown position.

Dual ascent decomposition

We split (3.61) into two parts, a minimization problem, and a complementarity problem. The minimization problem is as follows:

$$\begin{aligned}
 \min_{\mathbf{t}, \mathbf{x}, \mathbf{z}} \quad & \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}^u \cup \mathcal{N}^a} (t_{ij} - d_{ij}^2)^2 + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u} \mu_{ji}^T (x_j - z_{ji}) + \rho \|x_j - z_{ji}\|^2 \\
 \text{s.t.} \quad & \|x_i - s_j\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\
 & \|x_i - z_{ji}\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \\
 & \mathbf{t} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^{Np}, \mathbf{z} \in \mathbb{R}^{mp},
 \end{aligned} \tag{3.62}$$

and the complementarity problem looks to satisfy

$$\mathbb{R} \in \mu_{ji} \perp x_j - z_{ji} = 0 \quad \forall i \in \mathcal{N}_i^u, j \in \mathcal{N}_i^u. \tag{3.63}$$

Problems (3.62) and (3.63) are equivalent to (3.61) in the sense that they have the same optimal cost, and present the same optimal solutions. This implies that solving both problems jointly amounts to solving (3.61).

The idea behind this decomposition is to solve each of the problems at different stages, which will unfold to an algorithm of two loops. The inner loop will solve problem (3.62) for fixed $\boldsymbol{\mu} = (\mu_{ji})_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u}$, and after such solution the outer loop will take a gradient ascent step in order to solve the complementarity problem (3.63). We already discussed a similar approach in section 3.2.3 for the TL problem.

As we have been doing for localization problems, we reformulate (3.62) to a standard quadratic form:

$$\begin{aligned}
 \min_y \quad & y^T A_0 y + 2b_0^T y + c_0 \\
 \text{s.t.} \quad & y^T A_{ij} y + 2b_{ij}^T y + c_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a
 \end{aligned} \tag{3.64}$$

where $y = (\mathbf{t}^T, \mathbf{x}^T, \mathbf{z}^T)^T$ is a column vector of dimension $n + Np + mp$. Other terms take the following values:

$$A_0 = \begin{pmatrix} \mathbf{I}_n & \mathbf{0}_{n \times Np} & \mathbf{0}_{n \times mp} \\ \mathbf{0}_{Np \times n} & \rho \text{diag}[|\mathcal{N}_1^u| \mathbf{I}_p, \dots, |\mathcal{N}_N^u| \mathbf{I}_p] & \mathbf{U}_0^{xz} \\ \mathbf{0}_{mp \times n} & (\mathbf{U}_0^{xz})^T & \mathbf{I}_{mp} \end{pmatrix},$$

$$\begin{aligned}
b_0 &= \underbrace{((-d_{1j}^2)^T, \dots, (-d_{Nj}^2)^T)_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}}_{1 \times n}, \underbrace{\sum_{j \in \mathcal{N}_1^u} \mu_{j1}^T, \dots, \sum_{j \in \mathcal{N}_N^u} \mu_{jN}^T}_{1 \times Np}, \underbrace{(\mu_{j1}^T)_{j \in \mathcal{N}_i^u}, \dots, (\mu_{jN}^T)_{j \in \mathcal{N}_i^u}}_{1 \times mp}^T \\
A_{ij} &= \begin{pmatrix} 0_{n \times n} & 0_{n \times Np+mp} \\ 0_{Np+mp \times n} & U_{ij} \end{pmatrix}, \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \\
b_{ij} &= \underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times n}, \underbrace{(0, \dots, 0)}_{1 \times Np}, \underbrace{(0, \dots, 0)}_{1 \times mp}^T \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \\
A_{ij} &= \begin{pmatrix} 0_{n \times n} & 0_{n \times Np} & 0_{n \times m} \\ 0_{Np \times n} & \text{diag}[0_{p \times p}, \dots, 0_{p \times p}, \mathbf{I}_p, 0_{p \times p}, \dots, 0_{p \times p}] & 0_{Np \times m} \\ 0_{m \times n} & 0_{m \times Np} & 0_{m \times m} \end{pmatrix} \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\
b_{ij} &= \underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times n}, \underbrace{(0, \dots, 0, -s_j^T, 0, \dots, 0)}_{1 \times Np}, \underbrace{(0, \dots, 0)}_{1 \times mp}^T \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\
c_0 &= \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} d_{ij}^4, \quad c_{ij} = \|s_j\|^2 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a, \quad c_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u.
\end{aligned}$$

Elements of matrix U_0^{xz} can be determined as

$$(U_0^{xz})_{ij} = \begin{cases} -\rho \mathbf{I}_p & \text{if } i \in \mathcal{N}_j^u \\ 0_{p \times p} & \text{otherwise,} \end{cases} \quad (3.65)$$

and U_{ij} has the following form:

$$U_{ij} = \begin{pmatrix} 0_{p \times p} & \cdots & \cdots & \cdots & \cdots & 0_{p \times p} \\ 0_{p \times p} & \mathbf{I}_p & 0_{p \times p} & \cdots & -\mathbf{I}_p & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_{p \times p} & -\mathbf{I}_p & 0_{p \times p} & \cdots & \mathbf{I}_p & \cdots \\ 0_{p \times p} & \underbrace{\cdots}_{j\text{'th column}} & \cdots & \cdots & \underbrace{\cdots}_{Np+j\text{'th column}} & 0_{p \times p} \\ & \underbrace{\hspace{10em}}_{Np} & & \underbrace{\hspace{10em}}_{mp} & & \end{pmatrix}, \quad (3.66)$$

where U_0^{xz} is a matrix of dimension $Np \times mp$, and U_{ij} is a square matrix of dimension $Np + mp$ that couple variables x_j and z_{ji} when $j \in \mathcal{N}_i^u$.

Rewriting problem (3.62) to (3.64) is of interest, because once we have the problem in standard form we can use convex optimization tools to solve the corresponding relaxed semidefinite problem. If strong duality holds between (3.62) and its dual, we can expect to

recover the optimal solution from the semidefinite problem. Then, we can use the dual ascent method to update the dual variables $\boldsymbol{\mu}$ and solve the complementarity problem (3.63).

The dual ascent method updates the variables as follows:

$$\boldsymbol{\mu}_{ji}^{k+1} = \boldsymbol{\mu}_{ji}^k + \alpha_k (x_j^{k+1} - z_{ji}^{k+1}) \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u, \quad (3.67)$$

where k index indicates the iteration step and α_k the step size of the dual ascent method. The previous dual variables can be used in the following problem:

$$\begin{aligned} (\hat{\mathbf{t}}, \hat{\mathbf{x}}, \hat{\mathbf{z}})(\boldsymbol{\mu}^k) = \arg \min_{\mathbf{t}, \mathbf{x}, \mathbf{z}} \quad & \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} (t_{ij} - d_{ij}^2)^2 + \sum_{i \in \mathcal{N}^u} \sum_{j \in \mathcal{N}_i^u} (\boldsymbol{\mu}_{ji}^k)^T (x_j - z_{ji}) + \rho \|x_j - z_{ji}\|^2 \\ \text{s.t.} \quad & \|x_i - s_j\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^a \\ & \|x_i - z_{ji}\|^2 - t_{ij} = 0 \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u \\ & \mathbf{t} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^{Np}, \mathbf{z} \in \mathbb{R}^{mp}. \end{aligned} \quad (3.68)$$

Iteratively solving these problems, we propose [algorithm 16](#). The algorithm is still centralized because problem (3.68) is not separable.

Algorithm 16 Dual ascent method of problem (3.61).

- 1: Initialize $\mathbf{x}^0, \mathbf{z}^0$ and $\boldsymbol{\mu}$. Set $k \leftarrow 0$ and $\{\alpha_k\}$.
 - 2: **while** $\mathbf{x}^k$ and $\mathbf{z}^k$ are not a stationary points **do**
 - 3: $\mathbf{x}^{k+1} \leftarrow \hat{\mathbf{x}}(\boldsymbol{\mu}^k)$, i.e., solution of (3.68).
 - 4: $\mathbf{z}^{k+1} \leftarrow \hat{\mathbf{z}}(\boldsymbol{\mu}^k)$, i.e., solution of (3.68).
 - 5: $\boldsymbol{\mu}_{ij}^{k+1} = \boldsymbol{\mu}_{ij}^k + \alpha_k (x_i^{k+1} - z_{ij}^{k+1}) \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u$.
 - 6: **end while**
-

NOVA decomposition of joint problem

As we already pointed out, (3.64) is not separable. Furthermore, if the $\mathcal{S}$ -property holds, it is required that the matrices associated to the quadratic functions satisfy a positive semidefinite relation. Introducing dual variable Z , we can reformulate the requirement as a complementarity problem:

$$0 \preceq Z \perp X = A_0 + \sum_{i \in \mathcal{N}_i^u} \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij} A_{ij} \succeq 0, \quad (3.69)$$

where we introduced X for notational convenience. An iterative procedure that aims to solve (3.69) is obtained with

$$Z^{k+1} = \left[Z^k - \alpha^k X \right]_+, \quad (3.70)$$

as we have discussed in [section 2.5.3](#). Interestingly, it is possible to process step (3.70) only using neighbor communications. In particular, X has a block diagonal structure, i.e., $X = \text{diag}[X_1 \otimes \mathbf{I}_p, \dots, X_N \otimes \mathbf{I}_p]$, where

$$X_i = \begin{pmatrix} \rho|\mathcal{N}_i^u| + \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} \lambda_{ij} & -\rho - (\lambda_{ji})_{j \in \mathcal{N}_i^u} \\ (-\rho - (\lambda_{ji})_{j \in \mathcal{N}_i^u})^T & \text{diag}[\rho + (\lambda_{ji})_{j \in \mathcal{N}_i^u}] \end{pmatrix} \quad (3.71)$$

and $X_i \in \mathcal{S}^{|\mathcal{N}_i^u|+1}$. We conclude that $Z_i^{k+1} = [Z_i^k - \alpha^k X_i]_+$ can be computed locally by node i aggregating the dual variables λ_{ji} from its neighboring nodes. This allows to process a completely distributed algorithm.

The next step is to solve (3.68) in a distributed way. In order to accomplish that, we may use the NOVA decomposition framework from [69]. We already introduced this framework in [chapter 2](#) and used it in the TL problem in [section 3.3.3](#).

In essence, we introduce N decoupled subproblems that only depend on local variables of the nodes. The coupling variables are fixed to a value from a previous iteration. And convergence of local variables to a joint consensus is pursued by solving the complementarity problems with a gradient ascent procedure. To the best of our knowledge, this is the first attempt to solve the semidefinite formulation of the SNL problem fully, without further simplifications than the original relaxation of the quadratic problem.

The first step is to specify the local problem that each node must solve. Each program consists on optimizing t_{ij} , x_i and z_{ji} jointly (for all $j \in \mathcal{N}_i^u$) in a single instance:

$$\begin{aligned} (\hat{\mathbf{t}}_i, \hat{x}_i, \hat{\mathbf{z}}_i)(\boldsymbol{\mu}^k, Z^q, \mathbf{x}^r, \mathbf{z}^r) = \arg \min_{\mathbf{t}_i, x_i, \mathbf{z}_i} & \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} (t_{ij} - d_{ij}^2)^2 + \rho \|x_j^r - z_{ji}\|^2 + \rho \|x_i - z_{ij}^r\|^2 \\ & + \sum_{j \in \mathcal{N}_i^u} \lambda_{ij} (\|x_i - z_{ji}\|^2 - t_{ij} + \text{tr}[Z^q A_{ij}]) + (\boldsymbol{\mu}_{ji}^k)^T (x_j^{k,q} - z_{ji}) + (\boldsymbol{\mu}_{ij}^k)^T (x_i - z_{ij}^{k,q}) \\ \text{s.t.} \quad & -\rho \leq \lambda_{ij} \perp \|x_i - z_{ji}\|^2 - t_{ij} + \text{tr}[Z^q A_{ij}] \quad \forall j \in \mathcal{N}_i^u \\ & \|x_i - z_{ji}\|^2 - t_{ij} = 0 \quad \forall j \in \mathcal{N}_i^a \\ & \mathbf{t}_i \in \mathbb{R}^{|\mathcal{N}_i^u|+|\mathcal{N}_i^a|}, x_i \in \mathbb{R}^p, \mathbf{z}_i \in \mathbb{R}^{|\mathcal{N}_i^u|p}, \end{aligned} \quad (3.72)$$

where $\mathbf{t}_i = (t_{ij})_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}$, $\mathbf{z}_i = (z_{ji})_{j \in \mathcal{N}_i^u}$, $\boldsymbol{\mu}^k = (\boldsymbol{\mu}_{ij}^k)_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u}$, $\mathbf{x}^r = (x_j^r)_{j \in \mathcal{N}^u}$ and $\mathbf{z}^r = (z_{ij}^r)_{i \in \mathcal{N}^u, j \in \mathcal{N}_i^u}$. Note that the decomposition gives rise to a three loop procedure. Parameter k refers to the outer loop iteration counter, q indicates the iteration number of the middle loop and r the most inner loop. The problem only refers to local variables of node i , and requires as input the past estimated location of its neighbors as well as to the dual variables associated to the consensus procedure and positive semidefiniteness property of the QP, i.e., μ_{ij} for each

$j \in \mathcal{N}_i^u$ and Z^q . The formulation has several constraints, and computing its solution requires semidefinite optimization tools, such as interior point methods.

Once more, problem (3.72) needs to be expressed as a standard QP for easy computation:

$$\begin{aligned} \min_{y_i} \quad & y_i^T A_{0i} y_i + 2b_{0i}^T y_i + c_{0i} \\ \text{s.t.} \quad & y_i^T A_{ij} y_i + 2b_{ij}^T y_i + c_{ij} = 0 \quad \forall j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a \end{aligned} \quad (3.73)$$

where $y_i = (\mathbf{t}_i, x_i, \mathbf{z}_i)$ has dimension $|\mathcal{N}_i^u| + |\mathcal{N}_i^a| + p + |\mathcal{N}_i^u|p$ and

$$\begin{aligned} A_{0i} &= \begin{pmatrix} \mathbf{I}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times p} & \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times |\mathcal{N}_i^u|p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \rho |\mathcal{N}_i^u| \mathbf{I}_p & \mathbf{0}_{p \times |\mathcal{N}_i^u|p} \\ \mathbf{0}_{|\mathcal{N}_i^u|p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{0}_{|\mathcal{N}_i^u|p \times p} & \rho \mathbf{I}_{|\mathcal{N}_i^u|p} \end{pmatrix}, \\ b_{0i} &= \underbrace{\left((-d_{ij}^2)^T \right)_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}}_{1 \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|}, \underbrace{\sum_{j \in \mathcal{N}_i^u} \frac{1}{2} \mu_{ij}^T - \rho z_{ij}^{k,q}}_{1 \times p}, \underbrace{-\left(\frac{1}{2} \mu_{ji}^T + \rho x_j^{k,q} \right)_{j \in \mathcal{N}_i^u}}_{1 \times |\mathcal{N}_i^u|p}^T, \\ c_{0i} &= \sum_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a} d_{ij}^4 + \sum_{j \in \mathcal{N}_i^u} \left((\mu_{ji}^k)^T x_j^{k,q} - (\mu_{ij}^k)^T z_{ij}^{k,q} + \rho (\|x_j^{k,q}\|^2 + \|z_{ij}^{k,q}\|^2) \right), \\ A_{ij} &= \begin{pmatrix} \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \cdots & \cdots & \cdots & \cdots & \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{I}_p & \cdots & -\mathbf{I}_p & \cdots & \mathbf{0}_{p \times p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \ddots & \ddots & \ddots & \cdots & \mathbf{0}_{p \times p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & -\mathbf{I}_p & \cdots & \mathbf{I}_p & \cdots & \mathbf{0}_{p \times p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \cdots & \cdots & \cdots & \cdots & \mathbf{0}_{p \times p} \end{pmatrix} \quad \forall j \in \mathcal{N}_i^u, \\ & \quad \underbrace{\hspace{10em}}_{|\mathcal{N}_i^u|p \text{ columns}} \\ b_{ij} &= \underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|}, \underbrace{(0, \dots, 0)}_{1 \times p}, \underbrace{(0, \dots, 0)}_{1 \times |\mathcal{N}_i^u|p}^T, \quad c_{ij} = \text{tr}[Z^q A_{ij}] \quad \forall j \in \mathcal{N}_i^u, \\ A_{ij} &= \begin{pmatrix} \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times p} & \mathbf{0}_{|\mathcal{N}_i^u|+|\mathcal{N}_i^a| \times |\mathcal{N}_i^u|p} \\ \mathbf{0}_{p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{I}_p & \mathbf{0}_{p \times |\mathcal{N}_i^u|p} \\ \mathbf{0}_{|\mathcal{N}_i^u|p \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|} & \mathbf{0}_{|\mathcal{N}_i^u|p \times p} & \mathbf{0}_{|\mathcal{N}_i^u|p \times |\mathcal{N}_i^u|p} \end{pmatrix} \quad \forall j \in \mathcal{N}_i^a, \\ b_{ij} &= \underbrace{(0, \dots, 0, -\frac{1}{2}, 0, \dots, 0)}_{1 \times |\mathcal{N}_i^u|+|\mathcal{N}_i^a|}, \underbrace{-s_j^T}_{1 \times p}, \underbrace{(0, \dots, 0)}_{1 \times |\mathcal{N}_i^u|p}^T, \quad c_{ij} = \|s_j\|^2 \quad \forall j \in \mathcal{N}_i^a. \end{aligned}$$

where $\text{tr}[Z^q A_{ij}] = p(Z_i)_{1,1} - \sum_{j \in \mathcal{N}_i^u} 2p(Z_j)_{j+1,i+1} + p(Z_j)_{j+1,j+1}$.

Denoting with $\Upsilon_i(\boldsymbol{\lambda}_i; \mathbf{x}^{k,q}, \mathbf{z}^{k,q}, \boldsymbol{\mu}^k, Z^q)$ the dual function of (3.73) of node i , the purpose is to solve all dual problems. In order to do so, each node solves:

$$\boldsymbol{\lambda}_i^{r+1} = \arg \max_{\boldsymbol{\lambda}_i \in \mathbb{R}^{|\mathcal{N}_i^u| + |\mathcal{N}_i^a|}} \Upsilon_i(\boldsymbol{\lambda}_i; \mathbf{x}^r, \mathbf{z}^r, \boldsymbol{\mu}^k, Z^q), \quad (3.74)$$

where $\boldsymbol{\lambda}_i = (\lambda_{ij})_{j \in \mathcal{N}_i^u \cup \mathcal{N}_i^a}$.

We can recover $\mathbf{x}^{r+1}$ and $\mathbf{z}^{r+1}$ from the KKT conditions. We construct the Lagrangian:

$$\begin{aligned} L(\mathbf{t}_i, x_i, \mathbf{z}_i, \boldsymbol{\lambda}_i^{r+1}; \mathbf{x}^r, \mathbf{z}^r) = & \sum_{j \in (\mathcal{N}_i^u \cup \mathcal{N}_i^a)} (t_{ij} - d_{ij}^2)^2 + \sum_{j \in \mathcal{N}_i^u} \left(\rho \|x_i - z_{ij}^r\|^2 + \rho \|x_j^r - z_{ji}\|^2 \right) \\ & + \sum_{j \in \mathcal{N}_i^u} \lambda_{ij}^{r+1} \left(\|x_i - z_{ji}\|^2 - d_{ij}^2 + \text{tr}[Z^q A_{ij}] \right) + \sum_{j \in \mathcal{N}_i^a} \lambda_{ij}^{r+1} \left(\|x_i - s_j\|^2 - d_{ij}^2 \right) \\ & + \sum_{j \in \mathcal{N}_i^u} \left((\boldsymbol{\mu}_{ij}^k)^T (x_i - z_{ij}^r) + (\boldsymbol{\mu}_{ji}^k)^T (x_j^r - z_{ji}) \right). \end{aligned}$$

We determine the partial derivatives and set them equal to zero:

$$\frac{\partial L}{\partial t_{ij}} = 2(t_{ij} - d_{ij}^2) - \lambda_{ij} = 0 \quad (3.75)$$

$$\frac{\partial L}{\partial x_i} = \sum_{j \in \mathcal{N}_i^u} \left(2\rho(x_i - z_{ij}^r) + \boldsymbol{\mu}_{ij}^k + 2\lambda_{ij}(x_i - z_{ji}) \right) + \sum_{j \in \mathcal{N}_i^a} 2\lambda_{ij}(x_i - s_j) = 0 \quad (3.76)$$

$$\frac{\partial L}{\partial z_{ji}} = -2\rho(x_j^r - z_{ji}) - \boldsymbol{\mu}_{ji}^k - 2\lambda_{ij}(x_i - z_{ji}) = 0. \quad (3.77)$$

Solving the system of equalities gives:

$$t_{ij}^{r+1} = \frac{\lambda_{ij}}{2} + d_{ij}^2 \quad (3.78)$$

$$x_i^{r+1} = \frac{\sum_{j \in \mathcal{N}_i^a} \lambda_{ij}^{r+1} s_j + \sum_{j \in \mathcal{N}_i^u} \left(\rho z_{ij}^{k,1} - \frac{1}{2} \boldsymbol{\mu}_{ij}^k + \lambda_{ij}^q \frac{\boldsymbol{\mu}_{ji}^k + 2\rho_j^k}{2\rho + 2\lambda_{ij}^{r+1}} \right)}{\sum_{j \in \mathcal{N}_i^a} \lambda_{ij}^{r+1} + \sum_{j \in \mathcal{N}_i^u} \rho - \frac{(\lambda_{ij}^{r+1})^2}{\rho + \lambda_{ij}^{r+1}}} \quad (3.79)$$

$$z_{ji}^{k,q+1} = \frac{\boldsymbol{\mu}_{ji}^k + 2\rho x_j^{k,q} + 2\lambda_{ij}^{r+1} x_i}{2\rho + 2\lambda_{ij}^{r+1}}, \quad (3.80)$$

which consist on analytical expressions that depend on $\boldsymbol{\lambda}_i^{r+1}$.

The dual ascent method is performed over $\boldsymbol{\mu}_{ij}$ after convergence of the inner loop:

$$\boldsymbol{\mu}_{ji}^{k+1} = \boldsymbol{\mu}_{ji}^k + \alpha_k (x_j^{k,q} - z_{ji}^{k,q}) \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u, \quad (3.81)$$

where k index indicates the iteration step in the outer loop and α_k the step size of the dual ascent method. The update is performed locally on both nodes i and j , who exchange their local estimations. Specifically, node i communicates its estimates x_i^r and z_{ji}^r to node j and vice versa. The full description of steps is presented in [algorithm 17](#).

Algorithm 17 Distributed algorithm of SNL problem based on NOVA decomposition

```

1: Initialize  $\mu^0, Z^0, \mathbf{x}^0, \mathbf{z}^0, \{\alpha_k\}$ . Set  $k \leftarrow 0$ .
2: while  $\mu$  is not a stationary point do
3:   Set  $Z^0 \leftarrow Z^q$ . Set  $q \leftarrow 0$ .
4:   while  $Z^q$  is not a stationary point do
5:     Set  $\mathbf{x}^0 \leftarrow \mathbf{x}^r$  and  $\mathbf{z}^0 \leftarrow \mathbf{z}^r$ . Set  $r \leftarrow 0$ .
6:     while  $\mathbf{x}^r$  is not a stationary point do
7:       Solve  $\lambda_i^{r+1} \leftarrow \arg \max_{\lambda_i \in \mathbb{R}^{|\mathcal{N}_i^u|+|\mathcal{N}_i^a|}} \Upsilon_i(\lambda_i; \mathbf{x}^r, \mathbf{z}^r, \mu^k, Z^q) \forall i \in \mathcal{N}^u$  in parallel.
8:       Set  $t_{ij}^{r+1} \leftarrow (3.78)$ .
9:       Set  $x_i^{r+1} \leftarrow (3.79)$ .
10:      Set  $z^{r+1} \leftarrow (3.80)$ .
11:      Set  $r \leftarrow r + 1$ .
12:     end while
13:     Set  $Z_i^{q+1} = Z_i^q + \alpha^k X_i$  for all  $i \in \mathcal{N}^u$ .
14:     Set  $q \leftarrow q + 1$ .
15:   end while
16:   Set  $\mu_{ji}^{k+1} \leftarrow \mu_{ji}^k + \alpha_k (x_j^{k,q} - z_{ji}^{k,q}) \quad \forall i \in \mathcal{N}^u, j \in \mathcal{N}_i^u$ .
17:   Set  $k \leftarrow k + 1$ .
18: end while

```

Convergence discussion

The following discussion is similar to the one we introduced in [section 3.2.3](#). Convergence of [algorithm 16](#) and [algorithm 17](#) is guaranteed if three conditions are satisfied in the problem.

1. Step size α in the outer loop introduced in [\(3.27\)](#) is sufficiently small. This requirement is typical from gradient descent methods.
2. Parameter ρ is sufficiently large such that $A_0 + \sum_{i \in \mathcal{N}^u, j \in (\mathcal{N}_i^u \cup \mathcal{N}_i^a)} \lambda_{ij}^* A_i \succeq 0$ where λ_{ij}^* correspond to the dual variables that solve problem [\(3.62\)](#) and [\(3.63\)](#) simultaneously.
3. The $\mathcal{S}$ -property is satisfied in problem [\(3.62\)](#).

From our analysis we conclude that we cannot guarantee convergence of [algorithm 16](#) nor [algorithm 17](#) because we do not know if condition 3 is always satisfied. This seems to be a common problem in many localization algorithms that rely on semidefinite relaxations,

such as in [12, 73, 33]. In practice, however, convergence occurred in every scenario we simulated.

Finally, this implies that even if we initialize [algorithm 14](#) at a suboptimal stationary solution of problem (3.1), the algorithm still converges to the global minimum. To the best of our knowledge, this trait is unique from our formulation.

3.4 Simulations

In this section we illustrate the performance of the proposed algorithms as well as those existing in the literature. First we analyze the TL problem and then the SNL problem. In both cases we simulate the algorithms with suboptimal and optimal convergence guarantees.

3.4.1 Simulations on the TL problem

We consider a network of nodes uniformly deployed within a squared area of dimensions $(-10, 10) \times (-10, 10)$ in which target node is deployed at $(1, 1)$. We assume that the network is fully connected. This network structure favors the convergence speed of all algorithms we use, since the information is distributed in the fastest way possible. However, it is true that NEXT based algorithms and diffusion-ATC are able to exchange packets with local neighbors only, whereas ADMM requires a consensus step to fuse every node's estimates whenever the network is not fully connected. This sets NEXT and diffusion based algorithms at an advantage in networks with less connectivity, requiring less number of packet exchanges. In the following we analyze a set of different scenarios to show evidence of the pros and cons of each algorithm.

N=3 nodes and noiseless case

In this section we consider a network with very few nodes ($N=3$) and no added noise to analyze the performance of algorithms in scenarios where optimality is attainable, but measurements are scarce.

We use different merit functions to evaluate the performance and compare with the number of packet exchanges. A single transmission step is required per node on each iteration for all algorithms, except for diffusion-ATC which uses two communication steps per iteration as described in [20]: one to transmit local gradients to neighbors, and a second one to transmit local estimates. All graphs have been averaged over 100 simulations.

We denote as 'diffusion' the algorithm presented in Section V.B in [20] based on diffusion techniques, using the *adapt-then-combine* (ATC) scheme. We also compare our methods

with ‘NEXT-Linearized’ proposed in Section V.A in [27]. We denote our methods with ‘NEXT-Exact’ to the one we propose in section 3.2.2, and as ‘ADMM-Exact’ to the algorithm we describe in section 3.2.1.

The simulation parameters for this methods are as follows. In our method ‘ADMM-Exact’ we used $\rho = 1000$ and $\varepsilon = 10^{-8}$ for step 8 of Algorithm 10. For ‘diffusion’ we used a step size of 0.2 for the *adaptation* step. For ‘NEXT-Linearized’ we used $\rho_{\text{Lin}} = 1000$ and fixed step size $\alpha_{\text{Lin}}^k = 0.1$; and for ‘NEXT-Exact’ we used $\rho_{\text{Exact}} = 1000$ and initial step size $\alpha_{\text{Exactn}}^k = 1$. All parameters were chosen to obtain the best performance of each algorithm as well as present a robust performance in the 100 simulations analyzed.

In order to compare all algorithms we use three different merit functions. The original cost function (3.1) measures the convergence speed towards the solution point evaluated at the mean point of all nodes:

$$f(x^k) = \sum_{i \in \mathcal{N}} (\|\bar{x}^k - s_i\|^2 - d_i^2)^2, \quad (3.82)$$

where $\bar{x}^k$ is the mean over all node estimates x_i^k . These results are depicted in figure 3.2a. In this case, ‘ADMM-Exact’ converges the fastest, followed by ‘NEXT-Exact’, ‘NEXT-Linearized’ and ‘diffusion-ATC’. This performance is reasonable because ADMM performs well in fully connected networks, and NEXT based algorithms may get a slight delay in their estimates while their local gradients get assimilated by every node. Furthermore, ‘NEXT-Exact’ performs better than ‘NEXT-Linearized’ because it accounts for more information in the optimization subproblems, and diffusion gets the worst performance in part because it mimics a gradient descent, and also, it requires two packet transmissions per iteration.

A second merit function we consider measures the speed of iterates over $\bar{x}^k$, which we denote as J^k :

$$J^k = \|\bar{x}^k - \bar{x}^{k-1}\|_{\infty}. \quad (3.83)$$

Its results are depicted in figure 3.2b. These results are coherent with the previous ones, as they show convergence speeds similar to those in the previous figure. Fastest speed for ADMM, a slight accommodating time at the beginning for NEXT based schemes, and a smooth behavior for diffusion.

Finally, the third merit function measures the divergence of the node’s estimates compared to the mean $\bar{x}^k$:

$$D^k = \frac{1}{N} \sum_{i \in \mathcal{N}} \|x_i^k - \bar{x}^k\|^2 \quad (3.84)$$

For this merit function we only evaluate ‘ADMM-Exact’ because at every iteration step there is no divergence between the mean and local estimates in the other algorithms. The

consensus in ‘NEXT-Exact’, ‘NEXT-Linearized’ and ‘diffusion’ is exact, since the network is fully connected. The result is depicted in [figure 3.2c](#), which shows a consistent agreement between nodes as the number of iterations increase.

N=3 nodes and high noise

In this scenario we want to give evidence that when there is high noise and a low number of nodes, it is possible that the algorithms converge to suboptimal stationary solutions rather than the global ones. We introduce in this case the algorithm presented in Section II.B in [5] and described in [section 3.2](#) to determine the global (centralized) solution of problem (3.1). We denote this algorithm as ‘optimal’ in our graphs.

We consider a network of $N=3$ nodes and additive Gaussian noise with zero mean and deviation $\sigma = 2$. The simulation parameters are the same as in the previous case. [Figure 3.3a](#) now shows a small gap with the optimal solution for all algorithms. Since they all have the same initialization point, they reach the same stationary solution, and sometimes, there is a small gap with the optimal value. This may be of importance, because although the gap is small, the stationary solution may not be close to the optimal value.

The other two [figures 3.3b](#) and [3.3c](#) show consistent behavior in the presence of noise. All figures are averaged over 100 simulations. Note that dual methods only converge if the conditions specified by [Proposition 5](#) are satisfied, so we verified them before every proposed scenario.

N=15 nodes and noiseless case

In the last scenario with the previous algorithms, we take a network of $N = 15$ nodes. We want to give evidence that with more nodes, the chances to fall into a suboptimal solution reduce. For these simulations we chose $\rho = 1000$, $\alpha_{\text{Lin}} = 0.01$ and $\alpha_{\text{Exact}} = 0.1$.

[Figure 3.4](#) presents all results. [Figure 3.4a](#) shows that all algorithms converge to the optimal solution, regardless of the network configuration or initialization point in the 100 trials. As in previous cases, [figures 3.4b](#) and [3.4c](#) show convergence speed of the iterates and consensus performance.

N=15 nodes and high noise

The purpose is the same as before: show that with more nodes, the nodes converge optimally, even in the presence of noise. Simulation parameters are the same as before, and the figures have been averaged on 100 trials.

Fig. 3.2 TL network simulation with $N = 3$ nodes and $\sigma = 0$.

Fig. 3.3 TL network simulation with $N = 3$ nodes and $\sigma = 1$.

(a)

(b)

(c)

Fig. 3.4 TL network simulation with $N = 15$ nodes and $\sigma = 0$.

The simulations are summarized in [figure 3.5](#). The cost function from [3.5a](#) gives evidence that all algorithms are able to converge optimally in all simulations. All algorithms perform very well, although dual ascent, NEXT-Exact and ADMM-Exact are the fastest.

Distributed dual ascent

We show under different network configurations, the required number of packets to solve the TL problem when using a fully distributed algorithm based on dual ascent methods. We illustrate the behavior of the cost function per packet exchange, that is, at every inner loop iteration in the following figures. There are several jumps in the figures corresponding to the change of dual variable μ or Z , and straight lines until the convergence criteria of the inner loop is satisfied. This decomposition is certainly more complicated than the suboptimal versions we have analyzed, but it is also the first attempt at guaranteeing optimality. It remains as future work to reduce the number of loops as well as to attain competitive convergence speeds.

[Figures 3.6](#) and [3.7](#) represent the convergence speed per packet exchange for a network of 3 nodes in the noiseless case and with additive Gaussian noise of 1 m deviation, respectively.

Likewise, [figures 3.8](#) and [3.9](#) represent the convergence speed for a network of 15 nodes in the noiseless case and with additive Gaussian noise of 1 m deviation, respectively.

N=3 and noiseless case: fixed deployment

We want to show the performance of primal problem descent algorithms vs. the performance of the algorithms based on the dual ascent decomposition. In this case we use the network scheme introduced at the beginning of the chapter where three nodes were positioned at $s_1 = (-5, 8)^T$, $s_2 = (-3, -9)^T$ and $s_3 = (-4, 2)^T$, and target node at $(1, 1)$. We compare the dual descent algorithm [algorithm 13](#) with ADMM and NEXT using exact decompositions, [algorithm 10](#) and [algorithm 11](#), respectively.

All algorithms have been initialized in the neighborhood of local stationary point $(-9.01, -0.14)^T$, where we added Gaussian noise of deviation 1. Simulation parameters have been set as follows: $\rho = 100$ and step size for NEXT have been set to 0.1. All plots have been averaged over 100 simulations with random initialization points.

We plot the cost functions versus the iteration number. Note that [algorithm 13](#) and [algorithm 14](#) have the same values at each outer iteration, since the later is derived from the former. [Figure 3.10a](#) shows the cost function achieved by each algorithm. Note how suboptimal algorithms reach a stationary solution which is not optimal. [Figure 3.10b](#) shows

Fig. 3.5 TL network simulation with $N = 15$ nodes and $\sigma = 1$.

Fig. 3.6 TL network simulation with $N = 3$ nodes and $\sigma = 0$.

Fig. 3.7 TL network simulation with $N = 3$ nodes and $\sigma = 1$.

Fig. 3.8 TL network simulation with $N = 15$ nodes and $\sigma = 0$.

Fig. 3.9 TL network simulation with $N = 15$ nodes and $\sigma = 1$.

how iterate values converge over time, and finally, [figure 3.10c](#) shows how consensus is achieved in both ADMM as well as dual gradient ascent algorithms.

3.4.2 Simulations of the SNL problem

We consider a network of 21 nodes forming a rectangular grid of -15×15 height and -45×45 width. We have 4 anchor nodes placed at $s_1 = (-45, 15)^T$, $s_2 = (-45, -15)^T$, $s_3 = (45, 15)^T$ and $s_4 = (45, -15)^T$ (in the corners of the grid) and 17 nodes with unknown location. These nodes are placed at fixed positions throughout the grid in a disposition of 7 columns and 3 rows. The purpose in this problem is to locate the nodes with unknown location with different values of additive noise and communication distances, which effectively vary the number of neighbors of each node. We compare our proposed algorithms with the parallel scheme described in [\[23\]](#). The initialization point for the location of the nodes is uniformly random and the same for every algorithm.

The merit function we will be using in this section is the *root mean square error* (RMSE) defined as:

$$\text{RMSE} = \left(\frac{1}{N} \sum_{i \in \mathcal{N}^u} \|x_i - x_i^0\|^2 \right)^{-1/2}, \quad (3.85)$$

where x_i^0 is the true position of node i . The reason for using this cost function is that it gives a qualitative understanding of the algorithmic results, whereas a comparison with the cost function [\(3.41\)](#) would give relative results among algorithms, but difficult to interpret individually.

N=17 nodes, noiseless case, communication distance 22 m

We consider the noiseless scenario where every node measures distances exactly and where each node can communicate up to a distance of 22 m. This ensures that every node has on average 5 neighbors in this scenario. [Figure 3.11](#) represents the performance of our algorithms based on the FLEXA decomposition framework and dual ascent, as well as the one proposed in [\[23\]](#). The figure has been averaged over 100 simulations with random initialization points. The optimal value of every scenario is zero, but the algorithms may fail to converge to the optimal value on some occasions as a result of their suboptimality convergence guarantees and low connectivity. We observe that FLEXA converges more often to the optimal solution than [\[23\]](#), and the dual ascent method converges to the optimal solution systematically. [Figure 3.12](#) shows a qualitative result of the estimated location of nodes using [algorithm 15](#) whenever the method converges to the optimal solution.

(a)

(b)

(c)

Fig. 3.10 TL network simulation with $N = 3$ nodes, $\sigma = 0$ and a fixed deployment. The network has multiple local minima. Algorithms are initialized in the neighborhood of a local minima.

Fig. 3.11 SNL network with $N = 17$ nodes, $\sigma = 0$ and communication distance 22 m.

Fig. 3.12 SNL network with $N = 17$ nodes, $\sigma = 0$ and communication distance 22 m.

Fig. 3.13 SNL network with $N = 17$ nodes, $\sigma = 1$ and communication distance 22 m.

Fig. 3.14 SNL network with $N = 17$ nodes, $\sigma = 1$ and communication distance 22 m.

N=17 nodes, high noise, communication distance 22 m

In this case we consider additive Gaussian noise with deviation of 1 m to every measured distance between nodes within communication distance. The performance in this case is slightly deteriorated because of the presence of noise. In addition to the previous analyzed algorithms, we also draw the optimal result that can be obtained using the centralized result described in [section 3.3.2](#). The resulting graph is represented in [figure 3.13](#), where all curves have been averaged over 100 simulations. Note that there is a gap between the attainable optimal result and the achieved curves because due to the random initialization point, the algorithms may converge on certain simulations to a stationary non-optimal solution. As in the previous case, we observe that [\[23\]](#) fails more often than FLEXA with bad initialization points. Dual gradient ascent converges optimally in every iteration. The qualitative graph of a successful result towards an optimal solution is represented in [figure 3.14](#).

Fig. 3.15 SNL network with $N = 11$ nodes, $\sigma = 0$ and communication distance 34 m.

N=11 nodes, noiseless, communication distance 34 m

We also want to represent the impact of connectivity in the performance. By increasing the connectivity distance of the network to 34 m, our algorithms become more robust, converging to the optimal solution systematically and reaching high accuracy. With such high connectivity, the average number of neighbors amounts to 10 and no node is connected to more than two anchor nodes at once.

We reduced the size of the network to 15 nodes because the dual ascent algorithm is computationally demanding and grows with the size of the network and number of connections. Similar to the previous described network, we placed the four anchor nodes at $s_1 = (-30, 15)^T$, $s_2 = (-30, -15)^T$, $s_3 = (30, 15)^T$ and $s_4 = (30, -15)^T$ (in the corners of the grid) and 11 nodes with unknown location. All nodes are placed in a grid of 3 rows and five columns.

We represent in figure 3.15 the comparison of our algorithms with [23], noting that the optimal cost is zero in the noiseless case. The figure has been averaged over 100 simulations, noting that our algorithms have converged optimally in all of them, whereas [23] have failed on some occasions.

N=11 nodes, high noise, communication distance 34 m

We plot in figure 3.16 the same scenario as in the previous case with additive Gaussian noise of 1 m of deviation. In this case, all algorithms converge very precisely to the optimal solution. Note that RMSE error is below 2 m in every algorithm, and that [23] reaches the highest precision. Figure 3.17 shows the qualitative results after augmenting the connectivity, which improves the precision compared with less connected networks.

Fig. 3.16 SNL network with $N = 11$ nodes, $\sigma = 1$ and communication distance 34 m.

Fig. 3.17 SNL network with $N = 11$ nodes, $\sigma = 1$ and communication distance 34 m.

Fig. 3.18 SNL network with $N = 11$ nodes, $\sigma = 0$ and communication distance 22 m.

N=11 nodes, distributed dual ascent

We want to illustrate the convergence speed of the distributed dual ascent method based on the number of exchanged packets. We consider the smaller network formed by 15 nodes, 5 anchors and 11 nodes with unknown location, and varying communication distances and added noise.

In figure 3.18 shows the convergence speed of algorithm 17 in a fully distributed setting without any added noise. Figure 3.19 is the same problem in the presence of Gaussian noise with deviation $\sigma = 1$ m. Note that we represent the number of packet exchanges in the x-axis, which correspond to the total number of iterations in the most inner loop. Due to the three loop scheme it is difficult to improve the precision of the results. Nonetheless, convergence is attained in all graphs.

Figure 3.20 considers another network with communication distance of 34 m (average number of neighbors around 10) without added noise, and, figure 3.21 with added Gaussian noise. In both cases, convergence is attained. The improved precision in figure 3.21 with regards to the centralized method is due to the achievable precision typical of interior-point methods and CVX [42] use.

Fig. 3.19 SNL network with $N = 11$ nodes, $\sigma = 1$ and communication distance 22 m.

Fig. 3.20 SNL network with $N = 11$ nodes, $\sigma = 0$ and communication distance 34 m.

Fig. 3.21 SNL network with $N = 11$ nodes, $\sigma = 1$ and communication distance 34 m.

Appendix 3.A Proofs of TL problem

Proof of Theorem 7. ADMM convergence is analyzed in [14] and it requires two assumptions to establish convergence of the objective value: *i)* that the functions are closed, proper and convex, and *ii)*, that a saddle point of the unaugmented Lagrangian exists. Our problem is closed and proper, but is non-convex. And we need to show that a saddle point exists.

The road map for our proof is as follows: *i)* show the existence of solution for (3.8); *ii)* prove the existence of a saddle point for the unaugmented Lagrangian; *iii)* review and verify that although the problem is non-convex, necessary conditions for ADMM to converge are satisfied.

We start analyzing the existence of solution: the objective function of (3.8) is continuous and coercive and the feasible region is the whole space $\mathbb{R}^p$. By the Weierstrass theorem [9, Prop.3.2.1] at least a solution exists, which we denote as x^* .

We define the unaugmented Lagrangian of (3.8) as follows:

$$L_0(\mathbf{x}, z, \boldsymbol{\mu}) = \sum_{i \in \mathcal{N}} ((\|x_i - s_i\|^2 - d_i^2)^2 + \tilde{I}_{\mathcal{Z}}(z) + \boldsymbol{\mu}_i^T (x_i - z))$$

where $\tilde{I}_{\mathcal{Z}}(z)$ is the extended indicator function for set $\mathcal{Z} = \{z \in \mathbb{R}^p | z = \frac{1}{Q} \sum_i (x_i + \frac{1}{2\rho} \boldsymbol{\mu}_i)\}$. It is defined as $\tilde{I}_{\mathcal{Z}}(z) = 0$ if $z \in \mathcal{Z}$, and $\tilde{I}_{\mathcal{Z}}(z) = +\infty$ otherwise. The indicator function is an instrumental term considered in the convergence analysis so that the unaugmented Lagrangian is not unbounded below on z (it would be linear without the indicator function). It does not affect the ADMM steps presented in section 3.2.1, since it is a projection on the algorithm iterates.

We make the following statement:

Lemma 7. L_0 has a saddle point $(\mathbf{x}^*, z^*, \boldsymbol{\mu}^*)$, i.e.,

$$L_0(\mathbf{x}^*, z^*, \boldsymbol{\mu}) \leq L_0(\mathbf{x}^*, z^*, \boldsymbol{\mu}^*) \leq L_0(\mathbf{x}, z, \boldsymbol{\mu}^*) \quad (3.86)$$

is satisfied for all $\mathbf{x}, z, \boldsymbol{\mu}$. Furthermore, strong duality holds.

Proof. We already showed (3.1) has a solution x^* . Because it is a differentiable unconstrained problem, it has to satisfy the null gradient property at the optimum:

$$\sum_{i \in \mathcal{N}} 4(\|x^* - s_i\|^2 - d_i^2)(x^* - s_i) = 0. \quad (3.87)$$

In order to prove the existence of a saddle point, it suffices to see that a set of primal and dual variables exist that satisfy the KKT conditions, and that the primal variables minimize the

objective. Let's consider primal variables $x_i^* = z^* = x^*$, and dual variables those that satisfy the null gradient of L_0 :

$$\frac{\partial L_0}{\partial x_i} = 0 \Leftrightarrow \mu_i^* = -4(\|x_i^* - s_i\|^2 - d_i^2)(x_i^* - s_i), \quad (3.88)$$

and for z we have to satisfy

$$z^* = \frac{1}{Q} \sum_i (x_i^* + \frac{1}{2\rho} \mu_i^*) = x^* + \frac{1}{2N\rho} \sum_i \mu_i^* \quad (3.89)$$

which is derived from the minimization of $L_0(\mathbf{x}^*, z, \boldsymbol{\mu}^*)$. Our candidate solution is feasible if $\sum_i \mu_i^* = 0$, which is satisfied because of (3.87). This proves the existence of a tuple $(\mathbf{x}^*, z^*, \boldsymbol{\mu}^*)$ that satisfies the KKT constraints.

Finally, proving eq. (3.86) is straightforward. The left inequality is tight since $\mu_i^T (x_i^* - z^*) = 0$ for all μ_i and $i \in \mathcal{N}$. The right inequality is also satisfied since $L_0(\mathbf{x}, z, \boldsymbol{\mu}^*)$ has a minimum which is $L_0(\mathbf{x}^*, z^*, \boldsymbol{\mu}^*)$. $\square$

Finally, we review the conditions that allow the steps derived in section 3.2.1 to converge. The proof presented in [14, Appx.A] allows to state the following result:

1. Residual convergence: $(x_i^k - z^k) \rightarrow 0$ as $k \rightarrow \infty$, i.e., the iterates approach feasibility.
2. Objective convergence: the objective function of the iterates approaches the optimal value.
3. Dual variable convergence: $\mu_i^k \rightarrow \mu_i^*$ as $k \rightarrow \infty$.

The proof is based on two key inequalities, A.2 and A3 from [14], which we verify are also satisfied in our problem. Before that, we introduce some new notation for improved clarity in the exposition:

$$f(\mathbf{x}) = \sum_{i \in \mathcal{N}} f_i(x_i), \quad f_i(x_i) = (\|x_i - s_i\|^2 - d_i^2)^2$$

$$\mathbf{r}^{k+1} = (r_i^{k+1})_{i \in \mathcal{N}}, \quad r_i^{k+1} = x_i^{k+1} - z^{k+1}.$$

The first key inequality A3 has the following form:

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^{k+1}) + (\boldsymbol{\mu}^*)^T \mathbf{r}^{k+1}, \quad (\text{A.3})$$

which we can see is satisfied in our problem, since it is a direct consequence of the existence of a saddle point in the unaugmented Lagrangian, which we proved in Lemma 7.

The second key inequality A2 has the following form:

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) \leq -(\boldsymbol{\mu}^{k+1})^T \mathbf{r}^{k+1} - 2\rho(B(\mathbf{z}^{k+1} - \mathbf{z}^*))^T (-\mathbf{r}^{k+1} + B(\mathbf{z}^{k+1} - \mathbf{z}^*)) \quad (\text{A.2})$$

where $B = -\text{col}((\mathbf{I}_p)_{i \in \mathcal{N}})$ is column matrix of size $pN \times p$. A.2 can be derived for our problem as follows. First, we specify the first order necessary condition of optimality at iteration step $k + 1$:

$$\nabla_{x_i} L^i(x_i^{k+1}, \mathbf{z}^k, \boldsymbol{\mu}^k) = \nabla_{x_i} f_i(x_i^{k+1}) + \boldsymbol{\mu}_i^k + 2\rho(x_i^{k+1} - \mathbf{z}^k) = 0 \quad (3.90)$$

where we impose that x_i^{k+1} is the actual minimizer of L^i as we derived in [section 3.2.1](#), for all $i \in \mathcal{N}$.

Since $\boldsymbol{\mu}_i^{k+1} = \boldsymbol{\mu}_i^k + 2\rho r_i^{k+1}$ (see step 17 in Algorithm 10), we have $\boldsymbol{\mu}_i^k = \boldsymbol{\mu}_i^{k+1} - 2\rho r_i^{k+1}$, which we can introduce in equation (3.90):

$$\nabla_{x_i} f_i(x_i^{k+1}) + \boldsymbol{\mu}_i^{k+1} + 2\rho(\mathbf{z}^{k+1} - \mathbf{z}^k) = 0. \quad (3.91)$$

From the last result, we obtain that x_i^{k+1} also minimizes

$$f_i(x_i) + (\boldsymbol{\mu}_i^{k+1} + 2\rho(\mathbf{z}^{k+1} - \mathbf{z}^k))^T x_i \quad (3.92)$$

from which we conclude

$$f_i(x_i^{k+1}) + (\boldsymbol{\mu}_i^{k+1} + 2\rho(\mathbf{z}^{k+1} - \mathbf{z}^k))^T x_i^{k+1} \leq f_i(x_i^*) + (\boldsymbol{\mu}_i^{k+1} + 2\rho(\mathbf{z}^{k+1} - \mathbf{z}^k))^T x_i^*. \quad (3.93)$$

Summing inequalities (3.93) for all $i \in \mathcal{N}$ yields

$$f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) \leq -\sum_{i \in \mathcal{N}} \boldsymbol{\mu}_i^{k+1}(x_i^{k+1} - x_i^*) - \sum_{i \in \mathcal{N}} 2\rho(\mathbf{z}^{k+1} - \mathbf{z}^k)^T (x_i^{k+1} - x_i^*). \quad (3.94)$$

We show $\sum_{i \in \mathcal{N}} \boldsymbol{\mu}_i^{k+1} = 0$. Consider the update rule $\boldsymbol{\mu}_i^{k+1} = \boldsymbol{\mu}_i^k + 2\rho(x_i^{k+1} - \mathbf{z}^{k+1})$. Summing $\boldsymbol{\mu}_i^{k+1}$ for all $i \in \mathcal{N}$ and combining with $\mathbf{z}^{k+1} = \frac{1}{Q} \sum_i (x_i^{k+1} + \frac{1}{2\rho} \boldsymbol{\mu}_i^k)$ gives the desired result.

Adding the term $-\sum_{i \in \mathcal{N}} (\boldsymbol{\mu}_i^{k+1})^T (\mathbf{z}^* - \mathbf{z}^{k+1})$ to the right hand side of (3.94)—which is zero as seen in the previous paragraph—, grouping terms and substituting $x_i^* = \mathbf{z}^*$, $x_i^{k+1} = r_i^{k+1} + \mathbf{z}^{k+1}$, yields the final result A.2.

To conclude, because A.2 and A.3 are fulfilled, we can extrapolate the convergence results from [14] and establish the same convergence results. $\square$

Proof of Proposition 5. We have to prove that problem (3.24) presents the $\mathcal{S}$ -property in the noiseless case for any $\boldsymbol{\mu}$ and $\rho > 0$. We will use Theorem 1 presented in [52]. It states that the $\mathcal{S}$ -property holds if and only if the KKT conditions are satisfied plus $A_0 + \sum_i A_i \succeq 0$, which is the approach we follow here.

Given the Lagrangian of problem (3.24):

$$L(\mathbf{t}, \mathbf{x}, z, \boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{i \in \mathcal{N}} (t_i - d_i^2)^2 + \mu_i^T (x_i - z) + \rho \|x_i - z\|^2 + \lambda_i (\|x_i - s_i\|^2 - t_i), \quad (3.95)$$

we derive its KKT necessary conditions:

$$\frac{\partial L}{\partial t_i} = 2(t_i - d_i^2) - \lambda_i = 0 \quad (3.96)$$

$$\frac{\partial L}{\partial x_i} = \mu_i + 2\rho(x_i - z) + 2\lambda_i(x_i - s_i) = 0 \quad (3.97)$$

$$\frac{\partial L}{\partial z} = -\mu_i + \rho(x_i - z) = 0 \quad (3.98)$$

$$\mathbb{R} \ni \lambda_i \perp (\|x_i - s_i\|^2 - t_i) = 0 \quad (3.99)$$

$$\mathbb{R} \ni \mu_i \perp (x_i - z) = 0. \quad (3.100)$$

We propose as candidate solution $x^* = x_i^* = z^*$ and $t_i^* = \|x^* - s_i\|^2$, where x^* corresponds to an optimal solution of the SRLS-TL problem. We can also determine $\lambda_i^* = 2(\|x^* - s_i\|^2 - d_i^2)$ uniquely. Furthermore, $\mu_i^* = -4(\|x^* - s_i\|^2 - d_i^2)(x^* - s_i)$. All these values verify the KKT conditions (3.96)–(3.100). We only need to check if $A_0 + \sum_i \lambda_i^* A_i \succeq 0$ is also satisfied. We have

$$A_0 + \sum_i \lambda_i^* A_i = \begin{pmatrix} \mathbf{I}_N & 0_{p \times Np} & 0_{N \times p} \\ 0_{Np \times N} & \text{diag}[\lambda_1^* + \rho \mathbf{I}_p, \dots, \lambda_N^* + \rho \mathbf{I}_p] & -\rho \text{col}[\mathbf{I}_p, \dots, \mathbf{I}_p] \\ 0_{p \times N} & -\rho \text{row}[\mathbf{I}_p, \dots, \mathbf{I}_p] & N\rho \mathbf{I}_p \end{pmatrix} \quad (3.101)$$

At this point we can use the Schur complement of $A_0 + \sum_i \lambda_i^* A_i$ to see if it is positive semidefinite. Given matrix X and S ,

$$X = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}, \quad S = C - B^T A^{-1} B, \quad (3.102)$$

we obtain:

$$S = N\rho \mathbf{I}_p - \rho \left(\sum_{i \in \mathcal{N}} \frac{\rho}{\lambda_i^* + \rho} \right) \mathbf{I}_p. \quad (3.103)$$

In the noiseless case we have that $\lambda_i^* = 2(\|x^* - s_i\|^2 - d_i^2) = 0$, and $S = 0$ for any $\rho > 0$. In other scenarios, $S \succeq 0$ is a necessary and sufficient condition for the $\mathcal{S}$ -property to hold. $\square$

Chapter 4

Robust worst-case analysis of demand-side management in smart grids

Demand-side management presents significant benefits in reducing the energy load in smart grids by balancing consumption demands or including energy generation and/or storage devices in the user's side. These techniques coordinate the energy load so that users minimize their monetary expenditure. However, these methods require accurate predictions in the energy consumption profiles, which make them inflexible to real demand variations. In this chapter we propose a realistic model that accounts for uncertainty in these variations and calculates a robust price for all users in the smart grid. We analyze the existence of solutions for this novel scenario, propose convergent distributed algorithms to find them, and perform simulations considering energy expenditure. We show that this model can effectively reduce the monetary expenses for all users in a real-time market, while at the same time it provides a reliable production cost estimate to the energy supplier.

4.1 Introduction

Smart grids represent the concept behind the intended evolution of the electric grid, which through information acquisition from the end users will allow for more efficient energy distribution, flexibility in network topology, adaptive load management and better integration of renewable energy plants with the consumption requirements of the users.

One important concept to achieve the smart grid's objectives is the demand-side management, which includes the techniques for better energy efficiency programs, distributed energy generation, energy storage and load management. The model presented in this work establishes a robust energy price one day ahead, which takes into account the production

costs of energy as well as uncertainties in the expected energy loads. We also consider a real-time market, where users are charged their specific demanded energy at the robust price, permitting some deviation without any extra penalty. This model provides some flexibility to every user in their real-time consumption energy loads and calculates the energy prices in a conservative manner, taking into account production costs and possible errors. In particular, we focus on a robust analysis of the demand-side management algorithms, where the energy strategies account for worst-case error deviations in the predicted loads. In our formulation, these error terms vary from user to user and depend on the energy load profiles.

We show that this model can effectively reduce the monetary expenses of all users in the real-time market, while at the same time it can provide a confident production cost estimate to the energy supplier. Our analysis also shows that the added computational cost of these error calculations is minimal and the monetary gain is significant. Moreover, the calculations are amenable to a distributed implementation within the communications network of the smart grid. This is important because distributed algorithms are beneficial in terms of communication efficiency and computation scalability [10].

To further support our results we perform simulations in a real-time modeled scenario. We also compare our robust proposal with a naive version where users do not take into account price deviations, but where the energy supplier still establishes a robust price. Finally, we also present results illustrating the convergence behavior of our algorithm.

This study can be of special interest to energy supply authorities who can analyze how much energy they should account for if they plan for grid load estimation errors and peak demand costs. Even though users may be only interested in their monetary expenditure as energy consumers, supply authorities have to account for local deviations in the distribution network and coordinate with all agents to provide unobstructed service by controlling regular and peak supply stations. The fact that an agent in the network can alter the expected energy load by injecting or subtracting energy unexpectedly is a source of extra costs that have to be taken into account by suppliers that monitor the grid's energy demand. Therefore, these energy provision costs justify a modified pricing model and a worst-case analysis of the demand-side network.

4.1.1 Related Work

The work presented here is focused on the study of the demand-side management framework through noncooperative game theory. We build our results from [3], where the authors consider a day-ahead scheduling algorithm and the users optimize energy generation and storage profiles in a distributed manner. Our model extends [3] by adding error terms to the expected consumption profiles (energy loads in the smart grid) and admitting variations

in the estimated values with a bounded error. In addition, we analyze the effect that these error terms produce in the expected energy demand. In order to do so, we extrapolate some techniques from reference [24], where a robust algorithm for communications is developed.

Regarding a real-time energy market, several models have been proposed in the literature. For instance, the authors from reference [2] propose a real-time cost function with penalization terms in which the users minimize the expected monetary expenses. The energy production model assumes that the cost of energy is given by a quadratic function of the total demand, and therefore, the whole energy price of the smart grid is affected by the real-time variations. In such setting, the energy price is predetermined one day ahead, but it does not account for the unpredictable production costs of real-time demand. The penalty functions should account for these extra expenses, but they are also predetermined in advance. If they are set too costly, they would affect greatly the monetary expenses of users in the real-time market. On the other hand, if the penalties are set too low, they would cause an economic loss to the energy supplier. Our robust model, on the contrary, sets up an energy price that will allow the energy supplier to produce energy without incurring unplanned extra costs, and will allow users flexibility in their energy demand.

Another real-time model is introduced by reference [67], which proposes an optimization problem in which users demand energy according to a personal utility function. In this scenario prices are updated according to production costs, but demand is fully elastic, within some predetermined bounds, and the problem optimizes a social equilibrium point, rather than accounting for individual energy requirements. Furthermore, the energy supplier may experience monetary losses since the pricing mechanism does not cover production costs. In contrast, our algorithm accommodates individual energy requirements and ensures that production costs are covered by the pricing mechanism.

The authors of reference [22] propose a robust scenario for a real-time market based on linear programming, which allocates energy load profiles within a time frame. Because of the uncertainty of prices in real-time, users decide when to consume energy according to some modeling parameters, price range estimates and energy requirements. Although the algorithm is robust in the sense that it assumes a range of real-time energy prices, the model considers that prices are given by the energy supplier without taking into account production costs. Our robust model, on the other hand, establishes the prices one day ahead considering possible worst-case real-time variations, and takes into account production costs.

A relevant survey describing some of the last results and problems of the smart grid infrastructure can be found in [66], which analyzes different contributions and open problems in the topic, where demand-side-management is one example of these. Another publication that models the interaction among resident users and distributed energy resources is the one

introduced by [79], but it considers prefixed energy generation and storage load profiles, as opposed to our framework, which can adapt the profiles depending on the energy prices. Regarding user behavior, the authors of reference [82] propose a model that takes into account the subjective behavior of users, but still does not consider imprecise load predictions.

Therefore, our contribution lies in considering a day-ahead energy market and minimizing the energy costs of users in a robust situation where energy demand errors are taken into account. In Section 4.2 we describe the demand-side model, the day-ahead cost function, the real-time market and briefly describe the day-ahead optimization process. In Section 4.3 we introduce the robust game and analyze it's solution through an equivalent formulation. We also present a distributed algorithm that solves the game and analyze it's convergence properties. Finally, in Section 4.4 we present simulations to illustrate the validity of our results.

4.2 Smart Grid Model

In this section we introduce the smart grid model, the types of users in the demand-side of the network, the energy model, the pricing plan and the error terms. We consider a general framework with arbitrary convex regions and convex objectives.

4.2.1 Demand-Side Model

The smart grid topology can be divided in three parts, namely the supply-side, a central unit or regulation authority, and the demand-side. The supply-side involves the energy producers, as well as the distribution network. The central unit coordinates the optimization process. The demand-side entails the individual end-users as consumers, possibly with means to generate and/or store energy and/or vary their consumption profiles. The demand-side may also include users with great impact on the smart grid, such as industries with high energy demands or opportunistic agents who participate as local energy suppliers in the network.

In this paper, the demand-side management focuses on calculating optimal policies for monetary savings on the consumers' side, while accounting for worst-case deviations from the consumption profiles due to unpredictable considerations in the demand-side. The optimization process involves all users together with the central unit, who exchange information for a more efficient use and distribution of energy. This communication is accomplished with smart meters, which are the devices in charge of the bidirectional transmission of data with the central unit and the optimization process.

We assume that all end users indicate their intended amount of energy consumption one day ahead, so that energy prices can be calculated based on demand levels during the optimization process. Then, the users can readjust their generation, storing and consumption profiles according to the variability of prices. This process is repeated iteratively until convergence. Algorithms that analyze this process have been studied in [3] and [55]. However, due to errors in their estimations, the real consumption may vary from the announced values. If these variations are uncorrelated among users, sometimes the energy left unused by some consumers will be spent by others inducing that, on average, the supply-side will not require to adjust their generation rates. However, this expected result may not always be unconditionally true. In addition, there is interest from the supply-side to provide energy for unexpected events that may require higher demands. Therefore, we present analytical results on identifying worst-case energy errors, for a given confidence interval.

We represent the set of users who participate in this process with $\mathcal{D}$, and additionally subdivide them into the set of active users $\mathcal{N}$ and the set of passive users $\mathcal{P}$. Active users have some means of generating energy on demand (from dispatchable sources), possess devices to store energy for future use or adapt their consumption profiles. On the contrary, passive users do not adjust their energy consumption profiles and their consumption is estimated from historical data.

The consumption values are determined one day ahead among all users, and are subdivided in time-slots $h \in H = \{1, \dots, |H|\}$. We define the *estimated per-slot energy load profile* $l_n(h)$ that indicates the estimated energy usage of user $n \in \mathcal{D}$ at time-slot h . We also represent these variables as row vectors $\mathbf{l}_n = [l_n(1), \dots, l_n(|H|)] \in \mathbb{R}^{|H|}$, $\forall n \in \mathcal{D}$. Additionally, let $\mathbf{l} = [\mathbf{l}_1, \dots, \mathbf{l}_{|\mathcal{D}|}] \in \mathbb{R}^{|H| \times |\mathcal{D}|}$ denote the global vector with the energy loads of all users. Each individual term $\mathbf{l}_n$ includes the energy consumption of user n and the energy contributions or added expenditures that may decrease or increase the energy load due to the user owning some device to generate or store energy, respectively. Naturally, these profiles are variables to the optimization process and will vary according to the different strategies the user can select. It is satisfied that $l_n(h) > 0$ if the energy flows from the grid to the user, and $l_n(h) < 0$ otherwise, when the user is selling energy to the grid.

We assume that the total load in the network—*aggregate per-slot energy load*—at time slot h denoted by $L(h)$ is positive:

$$0 < L(h) \triangleq \sum_{n \in \mathcal{D}} l_n(h) \quad (4.1)$$

and it is calculated by the central unit by aggregating everybody's estimated load, including passive and active user's load.

Finally, we introduce the *per-slot energy load error profile* $\delta_n(h)$ as the difference from the estimated *per-slot energy load profile* $l_n(h)$ and the real value. This difference comes from the assumption that demand-side users do not know their consumption requirements with precision and they may deviate from their estimates. Additionally, active users may also deviate from their calculated generation and storing policies for unpredictable reasons such as malfunctioning, disconnections or lack of collaboration. For these reasons, introducing an error term makes the model more realistic. We represent these error terms both as row vectors $\boldsymbol{\delta}_n = [\delta_n(1), \dots, \delta_n(H)]$, $\forall n \in \mathcal{D}$ as the error profiles of user n and as column vectors $\boldsymbol{\delta}(h) = [\delta_1(h), \dots, \delta_{|\mathcal{D}|}(h)]^T$, $\forall h \in H$. Note that the index n or h allows to easily distinguish which one we are referring to. Additionally, let $\boldsymbol{\delta} = [\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_{|\mathcal{D}|}] \in \mathbb{R}^{|\mathcal{D}| \times |H|}$ denote the global vector with the error terms of all users.

4.2.2 Day-ahead Cost Model

The purpose of our day-ahead model is to establish an estimate of the energy price that is robust against real-time energy demand variations. From a practical perspective, it determines a worst-case energy production cost taking into account expected user energy loads plus uncertainties, so that the energy supplier does not suffer monetary losses due to unplanned real-time demand.

The energy price in the network is determined by the *grid cost function* C_h , which is fixed by the supply-authority and depends on the aggregate per-slot energy load $L(h)$ at each time-slot. We use a quadratic cost function, which is widely used in the literature (see e.g. [3, 55]):

$$C_h(L(h), \boldsymbol{\delta}(h)) \triangleq K_h \left(L(h) + \sum_{n \in \mathcal{D}} \delta_n(h) \right)^2 \quad (4.2)$$

where the coefficients $\{K_h\}_{h=1}^{|H|} > 0$ represent the energy cost at each time slot determined by the supply-side. In addition, the individual cost for user $n \in \mathcal{D}$ at time slot h is its proportional part of the total cost, i.e., $C_h(L(h), \boldsymbol{\delta}(h)) \cdot \left((l_n(h) + \delta_n(h)) / (L(h) + \sum_{n \in \mathcal{D}} \delta_n(h)) \right)$, so that its cumulative extension for all time-slots is given by:

$$f_n(\mathbf{l}, \boldsymbol{\delta}) \triangleq \sum_{h \in H} K_h \left(L(h) + \mathbf{1}^T \boldsymbol{\delta}(h) \right) (l_n(h) + \delta_n(h)) + M(\boldsymbol{\delta}_n) \quad (4.3)$$

where $\mathbf{l}_n$ is a convex function; $\mathbf{l}_n \in \Omega_{\mathbf{l}_n}$; all regions $\Omega_{\mathbf{l}_n}$ are convex, compact and independent among users; and $\mathbf{1}$ is a column vector of appropriate size. Additionally, we have included a penalization term $M(\boldsymbol{\delta}_n)$ that includes management or distribution costs, which accounts for

the extra expenses that the supply authority has to account for during the pricing process. We separate these penalization terms from the energy prices due to the inherent unpredictable nature of the energy demand, which is an additional consideration with regards to the models in references [3, 55].

Since we consider an arbitrary form of $\mathbf{l}_n$ and $\Omega_{\mathbf{l}_n}$, our robust proposal can be adapted to different demand-side models. For instance, in the model from [3], $\mathbf{l}_n$ has a linear expression which is a function of the generation and storing capabilities of its users. Likewise, the model from [55] optimizes an energy user demand scheduling algorithm where $\mathbf{l}_n$ is linear and the feasible region is convex. We can, therefore, adapt a robust energy price that accounts for real-time uncertainties. And apart from the models [3] and [55], we can trivially assume a third one: one in which all users are passive, have no elastic demand at all, and simply do not optimize their energy load profiles. In such case, our robust model can still be applied to determine worst-case estimation errors.

Finally, the error terms we have accounted for in (4.2) contribute to the user's monetary expense $f_n(\mathbf{l}, \boldsymbol{\delta})$, since they affect the real-time energy price. In order to deal with these unknown demand terms, in Section 4.3 we propose a worst-case analysis by solving a coupled min-max game. This analysis will allow the supply authority to provide for excess energy demands and establish a robust energy price that accounts for unplanned demand. In the next subsection we present how to integrate these worst-case error terms in a real-time scenario to obtain a robust model.

4.2.3 Real-time Pricing Model

Once the robust price has been established one day ahead, if their demand remains within the worst-case estimation error limits, users are charged the amount of energy they demand in real time at the robust price. Outside of these limits, a penalty function is introduced to cover any extra expenses. Specifically, our proposed real-time cost model is

$$f_n^{\text{rt}}(\mathbf{l}, \boldsymbol{\delta}^*, \mathbf{l}^{\text{rt}}) = \sum_{h \in H} K_h \hat{L}_r(h) (l_n^{\text{rt}}(h) + \Psi_{nh}(\mathbf{l}, \boldsymbol{\delta}^*, \mathbf{l}^{\text{rt}})) + M(\boldsymbol{\delta}_n) \quad (4.4)$$

where $\mathbf{l}$ is the energy load profile obtained from the day-ahead optimization process, $\boldsymbol{\delta}^*$ is the worst-case estimation error (which is analyzed in Section 4.3), and

$$\hat{L}_r(h) = \sum_{m \in \mathcal{D}} l_m(h) + \delta_m^*(h) \quad (4.5)$$

is the robust total demanded energy plus worst-case estimation errors, calculated in the day-ahead optimization process. Variable $\mathbf{l}^{\text{rt}}$ refers to the actual energy load that users demand in

real-time, and Ψ_h is a penalty function that increases the cost whenever a user deviates from its specified load profile range; for example the following function is a real-time cost model inspired by [2]:

$$\begin{aligned} \Psi_{nh}(\mathbf{l}, \boldsymbol{\delta}^*, \mathbf{l}^{\text{rt}}) = & v_h(l_n(h) - \delta_n^*(h) - l_n^{\text{rt}}(h))^+ \\ & + v_h(l_n^{\text{rt}}(h) - l_n(h) - \delta_n^*(h))^+ \end{aligned} \quad (4.6)$$

where v_h and v_h are scalar terms and $(\cdot)^+ = \max(\cdot, 0)$. The penalty values discourage the return of unused energy during night or low demand times, and higher use of energy during day or high demand times.

Note that the penalty function proposed in (4.6) differs from [2] in that it does not charge any extra cost within the uncertainty margins. The reason is that our robust algorithm already takes into account the extra costs of unplanned demand from the production model. Note also that a user can still deviate further than the worst-case error estimates and, in such case, some penalty function should dissuade the user from these extremes; nevertheless, these deviations are limited with some probabilistic confidence.

Thus, our robust model acknowledges the cost of producing a certain amount of energy plus uncertainties, whereas current real-time models (like [2]) only establish a surcharge for extra demand. Robust pricing models are of interest for energy suppliers as they take into account production costs under real-time price fluctuations. This is an important remark, because the real-time demand affects the whole energy price, and not only the extra demanded energy.

4.2.4 Day-ahead Optimization Process

We can now introduce the game $\mathcal{G}_{\boldsymbol{\delta}} = \langle \Omega_{\mathbf{l}}, \mathbf{f} \rangle$ that minimizes the user's monetary expense with fixed $\boldsymbol{\delta}$:

$$\mathcal{G}_{\boldsymbol{\delta}} : \begin{cases} \min_{\mathbf{l}_n} & f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta}) \\ \text{s.t.} & \mathbf{l}_n \in \Omega_{\mathbf{l}_n}, \end{cases} \quad \forall n \in \mathcal{D}$$

where $\mathbf{l}_{-n} = [\mathbf{l}_1, \dots, \mathbf{l}_{l-1}, \mathbf{l}_{l+1}, \dots, \mathbf{l}_{|\mathcal{D}|}]$ represents the rest of the other user's policies, which influence the user's decision. We consider the Nash Equilibrium (NE) as the solution concept of interest for this game, defined as a feasible point $\mathbf{l}^* = [\mathbf{l}_1^*, \dots, \mathbf{l}_{|\mathcal{D}|}^*]$, such that it satisfies

$$f_n(\mathbf{l}_n^*, \mathbf{l}_{-n}^*, \boldsymbol{\delta}) \leq f_n(\mathbf{l}_n, \mathbf{l}_{-n}^*, \boldsymbol{\delta}) \quad \forall \mathbf{l}_n \in \Omega_{\mathbf{l}_n} \quad (4.7)$$

for every player $n \in \mathcal{D}$. In our framework, $\mathcal{G}_{\delta}$ has at least an equilibrium point, which can be reached through a convergent algorithm as established by the following theorem:

Theorem 8. *The following holds for the game $\mathcal{G}_{\delta}$ with given and fixed error values δ :*

- a) *It has a nonempty, compact and convex solution set.*
- b) *The individual cost values for the payoff functions of each player are equal for any NE solution of the game.*
- c) *The game is monotone and NE solutions can be reached through algorithms of the kind proposed in [71].*

Proof. See Appendix 4.A. □

Convergent algorithms to reach such NE solutions are described in [71], which depend specifically on the convex structure of the game. In particular, they require a single-loop best response algorithm if the game is strongly monotone, or a double-loop best-response algorithm if it is just monotone (with an added proximity term). The specific steps, which are similar to those described in [3, Alg.1] for their particular model, but including fixed error terms, are as follows. First, users determine their respective $\mathbf{l}_n$ based on a day-ahead optimization process (considering δ fixed), where every user knows their energy consumption requirements and energy costs, and calculate their best strategies depending on the energy prices. Then, all users communicate their intended energy load profile $\mathbf{l}_n$ to the central unit and energy prices are recalculated. This procedure is repeated iteratively until all users converge to an NE.

4.3 Worst-Case Analysis of the Energy Load Profiles

In the formulation of $\mathcal{G}_{\delta}$ we implicitly assumed that δ is given or known a priori, or otherwise is set to zero. However, these error profiles are in fact unknown random terms that we added in order to correct any deviation from the real performance. In this section, these error terms will be treated as optimization variables, and analyzed from a worst-case performance perspective. This allows the regulation authority to jointly consider both the price and management/availability costs of energy. With this idea in mind, it is realistic to consider bounded error terms, represented by

$$\Delta_h \triangleq \{\delta(h) \in \mathbb{R}^{|\mathcal{D}|} \mid \|\delta(h)\|_2^2 \leq \alpha(h)\} \quad \forall h \in H \quad (4.8)$$

where $\alpha(h)$ are prefixed in advance and depend on the time-slot h due to more or less confidence in the regulator predictions. In practice, these values can be inferred by the regulation authority, based on historic data from consumers and expected variations. Quadratic constraints on the error terms are common in the literature, see e.g. [24, 80].

In order to propose this error model, we have assumed two conditions that are satisfied generally in practice: *i*) independence of the error terms among users, and *ii*), no knowledge of the specific probabilistic error profiles of users. Property *i*) is satisfied if we assume that users alter their predicted energy profiles independently (e.g. a user decides to charge an electric vehicle for an unexpected event). The energy profiles of different users can be correlated, but what we assume to be independent are the deviation errors from these users. Property *ii*) implies that if we do not have any knowledge of the error distribution of a specific user, then it is sensible to substitute such profile with the error distribution of an average user profile. By the central limit theorem such average user profile will approximate a Gaussian distribution. Then, the error contribution of all users can be modeled as a Gaussian distribution of zero mean, and variance the sum of all user variances. Specifically, equation (4.8) represents an upper bound to the overall network error variance.

In the following, we analyze a scenario that considers that every player has to account individually for their worst-case error terms and is charged accordingly. The model assumes that each player is responsible for some maximum error profiles in joint consideration with the other players.

Before the analysis, we bring up a useful result that guarantees global optimality for a maximization problem with quadratic convex objectives and constraints that we introduced in chapter 1.

Theorem 9. (from Theorem 3 in [45]). *Given the problem*

$$\begin{aligned} \max_{x \in \mathbb{R}^n} \quad & \frac{1}{2}x^T A x + b^T x + c \\ \text{s.t.} \quad & \frac{1}{2}x^T Q x + d^T x + e \leq 0 \end{aligned} \tag{4.9}$$

where $A \neq 0, Q \in \mathbb{R}^{n \times n}$ are positive semidefinite matrices, $b, d \in \mathbb{R}^n$ and $c, e \in \mathbb{R}$, and Slater's condition is satisfied: then $\bar{x}$ is a global maximizer iff there exists $\bar{\lambda} \geq 0$ satisfying

$$A\bar{x} + b = \bar{\lambda}(Q\bar{x} + d) \tag{4.10}$$

$$-A + \bar{\lambda}Q \text{ is positive semidefinite.} \tag{4.11}$$

Note that (4.10) is a KKT necessary condition, plus a second requirement (4.11) that will guarantee global maximum.

4.3.1 Worst-Case Min-max Game Formulation

In this section we consider a worst-case situation where the error terms are coupled among all of the users. We analyze which are the user's best response strategies and how to calculate the worst-case error terms. In particular, this transforms $\mathcal{G}_{\delta}$ into a game where all users have to solve a min-max cost function, with coupled interactions and constraints. The new game formulation $\mathcal{G}_m$, where each user n has to determine variables $\mathbf{l}_n$ and δ_n , is given by

$$\mathcal{G}_m : \begin{cases} \min_{\mathbf{l}_n} & \max_{\delta_n} f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \delta_n, \delta_{-n}) \\ \text{s.t.} & \delta(h) \in \Delta_h, \quad \forall h \in H \\ & \mathbf{l}_n \in \Omega_{\mathbf{l}_n} \end{cases} \quad \forall n \in \mathcal{D},$$

where region $\Omega_{\mathbf{l}_n}$ is convex, compact and nonempty. The min-max of game $\mathcal{G}_m$ for every user $n \in \mathcal{D}$ represents a game with a convex objective function in variables $\mathbf{l}_n$ and convex regions. Note that for the specific case that $M(\delta_n)$ is quadratic and convex in variable δ_n , then the maximization problem falls into the category to readily use Theorem 9, since all equations are quadratic. More generally, we can provide the following theorem for any continuous function $M(\delta_n)$:

Theorem 10. *The following holds for game $\mathcal{G}_m$ with min-max objectives:*

- a) *It has a nonempty, compact and convex solution set.*
- b) *It is monotone in variables $\mathbf{l}_n$, and NE can be reached through algorithms of the kind proposed in [71].*

Proof. The max operation inside the objective of each user in game $\mathcal{G}_m$ is performed over a convex and compact set and, therefore, it is well defined (it always has a solution). The maximum function in $\mathcal{G}_m$ preserves convexity of the objective in variable $\mathbf{l}_n$. Furthermore, if the point that maximizes the objective is unique, then the objective is differentiable in $\mathbf{l}_n$ with gradient $\nabla_{\mathbf{l}_n} f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \delta_n^*, \delta_{-n})$, and where $\delta_n^* = \arg \max_{\delta_n} f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \delta_n, \delta_{-n})$ denotes the maximal point. This property is guaranteed by Danskin's Theorem [8, Sec. B.5]. With this analysis in mind, the required assumptions from [71, Th.4.1.a] are satisfied and an NE exists. Furthermore, the Jacobian analysis performed for Theorem 8 (see Appendix 4.A) remains the same in $\mathcal{G}_m$, with the exception of substituting the fixed δ_n with the optimal solution of the max problem δ_n^* . Likewise, we can claim that the game is monotone and, therefore, that the solution set is also convex [71, Th.4.1.b]. By Danskin's theorem, if the maximum is not a unique point then the gradient would become a subdifferential, and slight variations would have to be considered that include the use of multifunctions [35]. $\square$

Game $\mathcal{G}_m$ falls into a rather novel category of min-max problems recently analyzed in [36]. Such reference proposes an alternative formulation for min-max games, where extra players are added to solve the maximization parts to decouple the min-max form. Moreover, reference [36] proves equivalence of NE for both formulations, and proposes solutions under monotonicity and other requirements. However, our problem $\mathcal{G}_m$ falls into a more general form with individual max objectives (rather than a common one), where the monotonicity property does not hold. Therefore, the approach from [36] becomes unsuitable for distributed algorithms. Furthermore, $\mathcal{G}_m$ incorporates global constraints which are not considered in the previous reference. For this reason, we have to develop a novel approach for this particular problem.

In order to analyze and solve the coupled maximization problems for all players, we first formulate an equivalent game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$, which is more tractable, together with a nonlinear complementarity problem (NCP) indicated by (4.12). The NCP is required to establish the equivalence together with other assumptions on the dual variables λ_h . We have also particularized the cost term to $M(\boldsymbol{\delta}_n) = \beta_m \|\boldsymbol{\delta}_n\|_2^2$ with $\beta_m \geq 0$, which is motivated to penalize the user's local deviations. Furthermore, this particular choice will allow us to obtain tractable results for the game and the NCP:

$$\begin{aligned} \tilde{\mathcal{G}}_m(\boldsymbol{\lambda}) : \begin{cases} \min_{\boldsymbol{l}_n} & \max_{\boldsymbol{\delta}_n} p_n(\boldsymbol{l}_n, \boldsymbol{l}_{-n}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda}) \\ \text{s.t.} & \boldsymbol{l}_n \in \Omega_{\boldsymbol{l}_n} \end{cases} \quad \forall n \in \mathcal{D} \\ 0 < K_h + \beta_m \leq \lambda_h \perp -\left(\|\boldsymbol{\delta}(h)\|_2^2 - \alpha(h)\right) \geq 0, \quad \forall h \end{aligned} \quad (4.12)$$

where $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{|H|}]^T$, $a \perp b$ indicates $a^T b = 0$, and

$$\begin{aligned} p_n(\boldsymbol{l}_n, \boldsymbol{l}_{-n}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda}) = \sum_{h \in H} (\beta_m - \lambda_h) \delta_n(h)^2 \\ + \sum_{h \in H} K_h \left(L(h) + \sum_{k \in \mathcal{D}} \delta_k(h) \right) (l_n(h) + \delta_n(h)). \end{aligned} \quad (4.13)$$

Next lemma shows the equivalence between $\mathcal{G}_m$ and $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$.

Lemma 8. *Games $\mathcal{G}_m$ and $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ together with the NCP specified by equation (4.12) present the same NE solutions.*

Proof. The KKT system of equations of $\mathcal{G}_m$ and those of game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ jointly with the NCP are the same, and therefore, any NE candidate solution can be derived from either formulation. A maximum point in both problems exists since all regions are compact. Inequality $\lambda_h \geq K_h + \beta_m$ guarantees that such point is maximum because Theorem 9 is

satisfied for every user in $\mathcal{G}_m$ and, likewise, is maximum in $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ because it makes the objective concave in $\boldsymbol{\delta}_n$. Therefore, the NE coincide. $\square$

Note that, in this equivalent formulation of $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ we have focused on the special case where all λ_h are equal for all players for every $h \in H$. The equilibrium points of this formulation are normally referred to as “variational” or “normalized” solutions [34], and the practical implication is that they establish a common price on a resource for all agents. These solutions also retain some stability properties compared to other equilibrium points where the dual variables are different among users, see [34] for further details.

One benefit from dealing with the new formulation $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ is that the *strong max-min property* is directly fulfilled if the objective is *convex-concave*, as opposed to $\mathcal{G}_m$ where it remains *convex-convex* (see [15, Ex.3.14b] for further details). This implies that the min-max can be changed into a max-min and the solution is not altered, i.e., equivalence of both problems holds. We see this is true as long as $\lambda_h \geq K_h + \beta_m$ because the objective is then concave in $\boldsymbol{\delta}_n$. This allows us to analyze the monotonicity of the new game setting $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ in these variables. Let us present a lemma that guarantees the existence of common λ_h ’s satisfying the previous requirements:

Lemma 9. *There always exists λ_h satisfying (4.12) (i.e., making $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ concave in $\boldsymbol{\delta}_n$) that is dual optimal in $\mathcal{G}_m$. A lower bound to this value is: $\lambda_h > \beta_m + K_h$ for all $h \in H$.*

Proof. See Appendix 4.B. $\square$

Now, we can establish the following result.

Theorem 11. *The following holds:*

- a) *Game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ has a nonempty and compact solution set.*
- b) *A variational solution of game $\mathcal{G}_m$ always exists.*
- c) *If additionally $\lambda_h > \frac{1}{2}K_h(|D| + 1) + \beta_m$, then the game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ is strongly monotone in variables $\boldsymbol{\delta}_n$ for all $n \in \mathcal{D}$, it has a unique NE solution, and it can be reached by a distributed asynchronous best response algorithm of the kind proposed by [71, Alg.4.1].*

Proof. See Appendix 4.B $\square$

In order to solve $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ and the NCP described by (4.12), we need to study the inner game formed from the coupled maximization problems in $\boldsymbol{\delta}$. In order to do that, we assume that $\boldsymbol{l}$ is fixed. Specifically, we analyze the following game:

$$\left\{ \max_{\boldsymbol{\delta}_n \in \mathbb{R}^{|H|}} p_n(\boldsymbol{l}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda}) \quad \forall n \in \mathcal{D} \right. \quad (4.14)$$

where (4.14) is solved jointly with the NCP (4.12). First, note that all users' objectives can be separated in $|H|$ independent problems. This is possible because the objective function p_n is expressed as a sum over $h \in H$ and (4.12) is expressed separately for every $h \in H$.

Each individual problem has a quadratic form and, hence, Theorem 9 can be applied to each of them. We can transform the model given by (4.12) and (4.14) into a form resembling (4.9), and identify terms $A = K_h + \beta_m$ and $Q = 1$. We conclude that (4.11) is satisfied if $\lambda_h \geq K_h + \beta_m$. Finally, we obtain the first order necessary conditions from (4.14) for all $n \in \mathcal{D}$:

$$K_h \left(l_n(h) + L(h) + \sum_{k \neq n} \delta_k(h) \right) + 2 \left(K_h + \beta_m - \lambda_h \right) \delta_n(h) = 0,$$

and form a system of equations together with (4.12). By separating variables and solving, we get

$$\delta_n(h) = \frac{\sqrt{\alpha(h)} (l_n(h) + L(h) + \sum_{k \neq n} \delta_k(h))}{\sqrt{\sum_n (l_n(h) + L(h) + \sum_{k \neq n} \delta_k(h))^2}}, \forall n \in \mathcal{D} \quad (4.15)$$

$$\lambda_h = K_h + \beta_m + K_h \frac{\sqrt{\sum_n (l_n(h) + L(h) + \sum_{k \neq n} \delta_k(h))^2}}{2\sqrt{\alpha(h)}} \quad (4.16)$$

where we have chosen positive sign to satisfy $\lambda_h \geq K_h + \beta_m$.

Since the solution to the fixed point equations is derived from conditions (4.10)–(4.11), they are, therefore, global maximum of the individual objective functions. As no player can further maximize their objectives given the other players strategies, they form an NE. If the game is strongly monotone (Th. 11c is satisfied), then asynchronous fixed-point iterations over these equations will converge to the global maximum. If the game is not strongly monotone, then the set of optimal equations given by (4.15)–(4.16) have to be solved in an alternative fashion. We propose a fixed-point iterative mapping that results from calculating equation (4.15) repeatedly, plus a projection operation. The mapping in vectorial form becomes:

$$T_h(\boldsymbol{\delta}(h), \mathbf{a}_h) = \Pi_{X_h} \left(\sqrt{\alpha(h)} \frac{\mathbf{a}_h + A \boldsymbol{\delta}(h)}{\|\mathbf{a}_h + A \boldsymbol{\delta}(h)\|} \right) \quad (4.17)$$

where $\mathbf{a}_h = L(h) + \mathbf{l}(h)$, $A = \mathbf{1}\mathbf{1}^T - \mathbf{I}$ with size $|\mathcal{D}| \times |\mathcal{D}|$ and $\Pi_{X_h}(\cdot)$ represents the euclidean projection onto region X_h . Such region X_h is defined as

$$X_h \triangleq \left\{ \boldsymbol{\delta}(h) \in \mathbb{R}_+^{|\mathcal{D}|} \mid \mathbf{1}^T \boldsymbol{\delta}(h) + \frac{1}{|\mathcal{D}|-1} \mathbf{1}^T \mathbf{a}_h - \sqrt{\alpha(h)|\mathcal{D}|} \geq 0 \right\} \quad (4.18)$$

where $\mathbb{R}_+^{|\mathcal{D}|}$ refers to the nonnegative quadrant, so that X_h corresponds to the upper halfspace region limited by the hyperplane in the definition.

Next we give the following properties of the mapping T_h :

Theorem 12. *Given the mapping $T_h(\boldsymbol{\delta}(h), \mathbf{a}_h)$ defined in (4.17) where $\mathbf{a}_h$ is fixed, then it follows:*

- a) $T_h : X_h \rightarrow X_h$ is a self-map for every $h \in H$.
- b) T_h is a contraction mapping.
- c) The update rule $\boldsymbol{\delta}(h)^{k+1} = T_h(\boldsymbol{\delta}(h)^k, \mathbf{a}_h)$ converges to a unique point $\boldsymbol{\delta}(h)^* \in X_h$ that solves the fixed point equation:

$$\boldsymbol{\delta}^*(h) = T_h(\boldsymbol{\delta}^*(h), \mathbf{a}_h).$$

- d) The point $\boldsymbol{\delta}^*(h)$ is an NE of game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$, with $\boldsymbol{\lambda}$ satisfying (4.12) and fixed strategies $\mathbf{l}(h)$.

Proof. See Appendix 4.C □

Because of Theorem 12 and the equivalence principle of Lemma 8, the fixed-point solution of the mapping also maximizes the objectives of game $\mathcal{G}_m$. Thus, the mapping provides an iterative method to find a solution of the game, regardless of its monotonicity properties.

Region X_h in (4.18) becomes a technical adjustment that allows the operator to be a contraction self-mapping. It has an involved interpretation, but in simple terms it limits the search space of the fixed-point to the upper region of the hyperplane. In particular, in such upper halfspace the mapping is always a contraction, while in the lower halfspace such property is not always satisfied. Therefore, a fixed point in X_h always exists, and a recursive algorithm will converge to such point.

Finally, we can explicitly give an expression of the projection operation into the halfspace X_h , which is given by

$$\Pi_{X_h}(x) \triangleq \begin{cases} x & \text{if } x \in X_h \\ x + \left(\sqrt{\alpha(h)} - (A^{-1}\mathbf{a}_h + x)^T \frac{\mathbf{1}}{\|\mathbf{1}\|} \right) \frac{\mathbf{1}}{\|\mathbf{1}\|} & \text{if } x \notin X_h, \end{cases}$$

which only involves linear operations.

4.3.2 Distributed Robust Algorithm

Distributed implementations are beneficial for the smart grid because of their computation scalability. Therefore, we propose the distributed Algorithm 18.

The algorithm solves $\mathcal{G}_m$, and because of the monotonicity shown in Theorem 10 in variable $\mathbf{l}$, convergence is guaranteed. The proximal term included in step (S.3) is necessary if $\mathcal{G}_m$ is only monotone, and τ needs to be sufficiently large and satisfy that the game becomes strongly monotone, as indicated in [71, Cor.4.1]. Any optimization toolbox would suffice to solve the strongly convex problem indicated in (S.3), which has a unique solution. In step (S.4.), we solve the inner max game $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ among all players with fixed components $\mathbf{l}_n^{i+1}$. In order to do so, we iteratively calculate $T(\boldsymbol{\delta}(h)^k, \mathbf{a}_h)$ until convergence, which is guaranteed by Theorem 12. The main outer loop is also repeated until convergence.

Regarding the information exchange costs, users solve their local optimization problems, and then communicate their intended energy load profiles to the central unit. The central unit aggregates all energy loads and computes the individual worst-case error terms and the robust energy price defined in (4.5). Then, the robust energy price and the individual worst-case error estimates are communicated back to the users (privacy of individual values is preserved). Finally, the users re-optimize their new energy load profiles.

It seems sensible that the inner loop described on step 4 of Algorithm 18 is performed by the central unit. It involves simple operations described by the mapping $T_h(\boldsymbol{\delta}(h), \mathbf{a}_h)$, which scales linearly with the number of users $|\mathcal{D}|$, and requires no communication in order to find a solution. An alternative computation of the error terms in a distributed fashion is also possible, but it would increase the required communication.

4.4 Simulation Results

In this section, we present simulation results from our robust analysis. First, in Subsection 4.4.1 we present a real-time cost comparison with a non-robust model to illustrate that the robust model can effectively reduce the monetary costs. Secondly, in Subsection 4.4.2, we compare our algorithm with a model formed by naive users who do not take into account the worst-case error terms. We show that the monetary losses in this case are not negligible, and emphasize the importance of optimizing the energy loads taking into account the error terms. In addition, we include convergence results.

In the following simulations, we used Algorithm 18 to determine $\boldsymbol{\delta}_n$ and $\mathbf{l}_n$ applied to a scenario similar to the one proposed in [2]. In this kind of scenario, users decide upon local energy generation and storage strategies. The total number of users $|\mathcal{D}|$ will vary between simulations, but the proportion of active users remains fixed in all of them, where

Algorithm 18 Distributed robust algorithm for game $\mathcal{G}_m$

(S.1) Given K_h for all $h \in H$. Set iteration index $i = 0$.

Set $\tau > 0$. Initialize $\boldsymbol{\delta}^0(h) = \frac{\sqrt{\alpha(h)}}{\sqrt{|\mathcal{D}|}} \mathbf{1}$.

(S.2) If a termination criterion is satisfied, STOP.

(S.3) Solve for every $n \in \mathcal{N}$

$$\mathbf{l}_n^{i+1} = \arg \min_{\mathbf{l}_n \in \Omega_{\mathbf{l}_n}} f(\mathbf{l}_n, \mathbf{l}_{-n}^i, \boldsymbol{\delta}^i) + \frac{\tau}{2} (\mathbf{l}_n - \bar{\mathbf{l}}_n^i)^2$$

(S.4) Iterate from $k = 0$ until convergence

$$\hat{\boldsymbol{\delta}}^{k+1}(h) = T(\hat{\boldsymbol{\delta}}^k(h), \mathbf{a}_h^{i+1}), \quad \forall h \in H \quad (4.19)$$

where $\mathbf{a}_h^{i+1} = \sum_{n \in \mathcal{D}} \mathbf{l}_n^{i+1}(h) + \mathbf{l}^{i+1}(h)$.

Set the fixed-point solution: $\boldsymbol{\delta}^{i+1} \leftarrow \hat{\boldsymbol{\delta}}^{k+1}$.

(S.5) If an NE has been reached, update centroids $\bar{\mathbf{l}}_n^i = \mathbf{l}_n^i$.

(S.6) Set $i \leftarrow i + 1$. Go to (S.2).

$|\mathcal{N}| = \lfloor |\mathcal{D}|/2 \rfloor$. From the active users set $\mathcal{N}$, the number of users that generate and store energy is $GS = |\mathcal{N}| - 2\lfloor |\mathcal{N}|/3 \rfloor$, the number of users who only generate is $G = \lfloor |\mathcal{N}|/3 \rfloor$ and the number of users who only store energy is $S = \lfloor |\mathcal{N}|/3 \rfloor$. The day is divided in $|H| = 24$ time slots. The average consumption per user is of 4.5 kWh with cost coefficients $K_{\text{day}} = 1.5K_{\text{night}}$, where we assumed there is lower price at times $h_{\text{night}} = [1, \dots, |H|/3]$ and higher price at time slots $h_{\text{day}} = [|H|/3 + 1, \dots, |H|]$. These values are determined so that the initial average price for the day without any demand-side management is of 0.1412 £/kWh (same as in [2]).

Other parameters involving active users with generation capabilities, GS and G , are: $g_n^{(\max)} = 0.4 \text{ kWh/h}$, which is the maximum energy production capability per hour by the user's owned infrastructure; and $\gamma_n^{(\max)} = 0.8g_n^{(\max)}$, which is the maximum energy production capability per day. For active users owning storing devices, GS and S , we define the following variables: the hourly batteries leakage rate $\alpha_n = \sqrt[24]{0.9}$, the charge efficiency $v_n^+ = 0.9$, the efficiency at discharge $v_n^- = 1.1$, the capacity of the storage device $c_n = 4 \text{ kWh}$, the maximum charge rate per hour $s_n^{(\max)} = 0.125c_n$, the charge level at the beginning of the day $q_n(0) = 0.25c_n$ and the difference of charge at the next day $\varepsilon_n = 0$. All these variables are further explained in reference [2]. In essence, the parameters presented form a convex model of the problem that can be solved using convex optimization solvers.

The energy demand profiles per user have been simulated using a typical day profile similar to the one used in [2] plus some random Gaussian noise that adds some variance to

the users. The $\alpha(h)$ terms have been fixed to be the 10% of the total energy demand per time slot $h \in H$.

4.4.1 Real-time Cost Comparison

In this subsection, we present a comparison of our robust model with the non-robust model from [2], when users deviate from their day-ahead optimized energy load profiles. Specifically, we present the average cost function over 100 independent simulations, where the real time error deviations are given by white Gaussian processes with zero mean and variance $\alpha(h)/|\mathcal{D}|$ for each user, and where the number of users $|\mathcal{D}|$ ranges from 100 to 2000. The cost function we use to evaluate the performance in real-time of Algorithm 18 is described in (4.4). Likewise, the cost function for user n we compare with, denoted as ‘non-robust’, has the following form:

$$f_n^{\text{nr}}(\mathbf{l}, \boldsymbol{\delta}^*, \mathbf{l}^{\text{rt}}) = \sum_{h \in H} K_h \hat{L}_{wc}(h) \left(l_n^{\text{rt}}(h) + v_h(l_n(h) - l_n^{\text{rt}}(h))^+ + v_h(l_n^{\text{rt}}(h) - l_n(h))^+ \right), \quad (4.20)$$

which has no error terms in contrast to (4.4). Note also that we are representing the average cost of all users in the network. Finally, values v_h and v_h are set to $v_h = 0.2\kappa$ for $h \in [0, \dots, 8]$, $v_h = 0.8\kappa$ for $h \in [9, \dots, 24]$ and $v_h = (\kappa - v_h)$, where $\kappa = \sqrt{|\mathcal{D}|/\alpha(h)}$. The specific simulation parameters v_h and v_h are obtained from [2], and parameter κ is introduced to incorporate the uncertainty of the real-time consumption in the penalty terms. Note that the value of κ is the inverse of the deviation of every user’s real-time error distribution.

The average results of the cost models are presented in Figure 4.1. The term $M(\boldsymbol{\delta}_n)$ is not present in (4.20), so we set $M(\boldsymbol{\delta}_n) = 0$ in (4.4) for a fair comparison. Commenting on these results, we observe that the cost for the users is significantly lower than with a penalty based model as (4.20). Therefore, this model effectively reduces the monetary expenses of all users in the real-time market, while at the same time provides a reliable production cost estimate to the energy supplier.

4.4.2 Comparison of Algorithm 18 with naive users

In order to further support our results, we include the performance of our robust algorithm and compare it to a scenario where users do not consider the worst-case error terms in their energy load profiles. In figure 4.2, we plot the monetary cost of all users in different sizes of the smart grid. The number of users in this plot varies from 100 to 2000 users, and the

Fig. 4.1 Real time averaged cost functions over 100 simulations comparing robust function (4.4) and (4.20). The left axis shows the monetary cost of all users and the right axis shows (in %) the mean relative gain when using the robust algorithm.

total number of time slots is $|H| = 24$. The curve we refer to as “naive users” is the solution to the game without accounting for any error terms (i.e., $\delta = 0$), but with added worst-case calculations in the price determined by the energy supplier. On the other hand, the robust curve represents the total energy cost when using Algorithm 18, which does take into account the worst-case error terms. The average relative gain over all users when using the robust algorithm is around the 7%, which remains almost constant with the number of users; these savings values are represented in the right hand axis of the figure for different number of users. This result illustrates that users should adjust their energy load profiles after taking the worst-case error terms into account. For these simulations we fixed the term $\beta_m = 0.001$.

In figure 4.3a, we plot the convergence rate of Algorithm 1, which depicts the averaged convergence rate particularized for 1000 users. The convergence rate is very similar to the results presented in earlier approaches (see, e.g., as [2, Alg.1]). Our algorithm has the same algorithmic steps except for the inner loop involved in step 4, which we didn’t observe to impact the computation time. In figure 4.3b, we plot the convergence speed of the contraction mapping from step (S.4) in algorithm 18. The x-axis shows the number of iterations and the y-axis shows the magnitude $\sum_{h \in H} \|\delta^{k+1}(h) - \delta^k(h)\|$. The stopping criteria was set to 10^{-8} . Recall that convergence is guaranteed since the computation in (S.4) is a contraction mapping.

Fig. 4.2 Energy cost of all players vs. number of users. The left axis shows the monetary cost and the right axis shows the mean relative gain (in %) when using the robust algorithm.

The proposed algorithm was implemented in Matlab R2015a using an Intel i5-2500 CPU @ 3.30GHz. The user's optimization problems were run in sequential order (although in practice they can run in parallel). The local convex optimization problems have unique optimal points, so any optimization solver will produce similar results as ours. In particular, we used the *fmincon* function from Matlab. The whole simulation took about 61.38 seconds for 1000 users.

As a conclusion, these simulations show consistent results in which there is an actual benefit of around the 7% when using a robust algorithm vs. a naive one (i.e., that does not take into account the error terms for the optimization process), while the extra computation required is minimal.

Appendix 4.A

Proof of Theorem 8. We first need to establish that function $f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta})$ is convex provided that $\mathbf{l}_n$ is a convex function. Indeed, this is satisfied because of the scalar composition rule [15, Sec.3.2.4], where $f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta})$ is monotone if $L > 0$, as required by (4.1). Then, because $\Omega_{\mathbf{l}_n}$ is convex by assumption, the individual user problem is also convex.

The rest of the results we establish are similar to the ones considered in [2, Th.1 & Th.2]. These are based on the equivalence of game $\mathcal{G}_{\boldsymbol{\delta}}$ to a variational inequality, as explained in [71]. To show a) is satisfied, it is sufficient to verify that the regions $\Omega_{\mathbf{l}_n}$ are compact and convex, which is satisfied by definition of $\mathcal{G}_{\boldsymbol{\delta}}$, and that the objective functions of all players

(a) Convergence rate of Algorithm 1 with convergence criteria $\|\mathbf{l}^i - \mathbf{l}^{i-1}\|_2 / \|\mathbf{l}^i\| \leq 10^{-2}$.

(b) Convergence rate of $\delta^k(h)$ from step 4 in Algorithm 18 with convergence criteria $\sum_{h \in H} \|\delta^{k+1}(h) - \delta^k(h)\| \leq 10^{-8}$.

Fig. 4.3 Convergence rates for 1000 users.

are differentiable and convex in variables $\mathbf{l}_n, \forall n \in \mathcal{D}$. Indeed, the convexity is guaranteed since the objective functions are quadratic with positive terms in the squared elements so that the Hessian results into a diagonal matrix of the form $\text{diag}(\{K_h\}_h)$ that is positive definite. Hence, the claim is satisfied applying [71, Th.4.1].

The justification of $b)$ is readily available by observing that the objective is convex in $\mathbf{l}_n(h)$ and that other players actions only affect the individual objective through the aggregate $L(h)$. Therefore, the maximum value is unique for each player given any strategy profile $\mathbf{l}_{-n}^*$ since the maximum does not depend on individual decisions. This shows that even if there are multiple NEs that reach the maximum of $f_n(\mathbf{l}_n, \mathbf{l}_{-n})$, the objective value of the user remains constant.

The game is monotone if the Jacobian $JF_{\mathcal{G}_\delta}(\mathbf{l}, \boldsymbol{\delta})$ is positive semidefinite, where $JF_{\mathcal{G}_\delta}(\mathbf{l}, \boldsymbol{\delta}) = (\nabla_{\mathbf{l}_m} F_n(\mathbf{l}, \boldsymbol{\delta}))_{n,m \in \mathcal{D}}$, $F_n = (\nabla_{\mathbf{l}_n} f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta}))_{n \in \mathcal{D}}$ and where $f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta})$ is described in (4.3). The Hessian $\nabla_{\mathbf{l}_n, \mathbf{l}_n}^2 f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta}) = \text{diag}(\{2K_h\}_{h \in H})$ is positive definite and the other terms $\nabla_{\mathbf{l}_m, \mathbf{l}_n}^2 f_n(\mathbf{l}_n, \mathbf{l}_{-n}, \boldsymbol{\delta}) = \text{diag}(\{K_h\}_{h \in H})$ for $m \neq n$ are also positive definite and, therefore, so is $JF_{\mathcal{G}_\delta}(\mathbf{l}, \boldsymbol{\delta})$. The claim is satisfied applying [71, Eq.(4.8)]. $\square$

Appendix 4.B

Proof of Lemma 9. We use the *extreme value theorem* from *calculus* (also known as the Bolzano-Weierstrass theorem), which states that a real-valued function on a nonempty compact space is bounded above, and attains its supremum. Then, Theorem 9 states that if a maximum exists, there also exists a $\bar{\lambda}$ that satisfies equation (4.11) and, therefore, we necessarily have that there exists some $\lambda_h \geq K_h + \beta_m$ that solves the problem. $\square$

Proof of Theorem 11. To proof part $a)$ of the theorem (the existence of NE in $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$) we use a result from [61, Th.1]. We need to show that assumptions A1-A3 and A5 are satisfied from the mentioned reference (A4 is not required). Indeed, $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ has a number of finite players, the regions for all players are compact and convex, and all objective functions are continuous on these regions (which satisfy A1-A3). Condition A5 is satisfied if we show that $p_n(\mathbf{l}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda})$ is concave in $\boldsymbol{\delta}_n$, which is true if its hessian is negative semidefinite. Indeed, $\nabla_{\boldsymbol{\delta}_n, \boldsymbol{\delta}_n}^2 p_n = -2\text{diag}(\{\lambda_h - K_h - \beta_m\})$ is diagonal, and is negative semidefinite if $\lambda_h \geq K_h + \beta_m$. This also justifies that the *strong max-min property* of $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ holds and the min-max can be transformed into a max-min without altering the result. This concludes the proof of part $a)$ of the theorem.

To prove part $b)$ we need Lemma 8, which guarantees the equivalence of $\mathcal{G}_m$ and $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ when (4.12) is satisfied. Since by Lemma 9 we have $\lambda_h \geq \beta_m + K_h$ satisfied for all players, we conclude that a variational solution (a solution in which all λ_h are shared among all users)

exists. Note that the game $\mathcal{G}_m$ cannot satisfy condition A5 from previous paragraph directly and, therefore, the reformulation as $\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})$ is convenient to conclude the existence of an NE.

In order to prove part c), it suffices to show that the Jacobian $JF_{\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})}(\boldsymbol{\delta}(h))$ of the variational inequality associated with the game is negative definite [71, (4.8)]. This is defined as $JF_{\tilde{\mathcal{G}}_m(\boldsymbol{\lambda})}(\mathbf{l}, \boldsymbol{\delta}) = (\nabla_m F_n(\mathbf{l}, \boldsymbol{\delta}))_{n,m \in \mathcal{D}}$, where $F_n = (\nabla_{\boldsymbol{\delta}_n} p_n(\mathbf{l}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda}))_{n \in \mathcal{D}}$ and $p_n(\mathbf{l}, \boldsymbol{\delta}_n, \boldsymbol{\delta}_{-n}, \boldsymbol{\lambda})$ is described in (4.13). The Hessian for each $h \in H$ is diagonal on all players, and the Jacobian on each $h \in H$ becomes

$$\nabla_{\boldsymbol{\delta}(h), \boldsymbol{\delta}(h)}^2 p_n(\mathbf{l}, \boldsymbol{\delta}, \boldsymbol{\lambda}) = (K_h + 2\beta_m - 2\lambda_h)\mathbf{I} + K_h \mathbf{1}\mathbf{1}^T \quad (4.21)$$

where $\mathbf{I}$ is the identity matrix of size $|\mathcal{D}| \times |\mathcal{D}|$, and $\mathbf{1}$ is column vector of ones of length $|\mathcal{D}|$. The Hessian matrix becomes diagonally dominant with negative diagonal entries for $\lambda_h > \frac{1}{2}K_h(|\mathcal{D}| + 1) + \beta_m$ and, thereby, negative definite [48, Cor.7.2.3]. If such condition is satisfied, then the game is strongly monotone and it has a unique NE that can be reached by a suitable algorithm [71]. $\square$

Appendix 4.C

For notation simplicity, we refer in this section $T_h(\boldsymbol{\delta}(h), \mathbf{a}_h)$ defined in (4.17) simply as $T(\mathbf{x}) = \Pi_X(\sqrt{\alpha} \frac{A\mathbf{x} + \mathbf{a}}{\|A\mathbf{x} + \mathbf{a}\|})$, where $\mathbf{a} = \mathbf{a}_h$ is of size $|\mathcal{D}| \times 1$, $A = \mathbf{1}\mathbf{1}^T - \mathbf{I}$ of size $|\mathcal{D}| \times |\mathcal{D}|$, $\mathbf{x} = \boldsymbol{\delta}(h)$ and where we drop all indexes on h .

Before proving Theorem 12 we first introduce the following general result:

Lemma 10. *Given any two vectors $x_1, x_2 \in \mathbb{R}^{|\mathcal{D}|}$, the following holds:*

$$\left\| \frac{x_1}{\|x_1\|} - \frac{x_2}{\|x_2\|} \right\| \leq \left\| \frac{x_1}{\gamma_s} - \frac{x_2}{\gamma_s} \right\| \leq \left\| \frac{x_1}{\gamma} - \frac{x_2}{\gamma} \right\|$$

where $\gamma_s \triangleq \min\{\|x_1\|, \|x_2\|\}$ and $\gamma \triangleq \min_{x \in X} \|x\|$, assuming in the last inequality that $\gamma_s \neq 0 \neq \gamma$.

Proof. We start with the first inequality and assume without loss of generality that $\|x_1\| \geq \|x_2\|$. Introduce the shorthands $a = \frac{x_1}{\|x_1\|} - \frac{x_2}{\|x_2\|}$, $b = \frac{x_1}{\|x_2\|} - \frac{x_1}{\|x_1\|}$ and, therefore, $a + b = \frac{x_1}{\|x_2\|} - \frac{x_2}{\|x_2\|}$. By the definition of scalar product, $\|a + b\|^2 = \|a\|^2 + \|b\|^2 + 2a^T b$. Hence, if $a^T b \geq 0$, we necessarily have $\|a + b\| \geq \|a\|$, since all the sums involve positive scalars.

Then, the inner product results:

$$a^T b = \left(\frac{x_1}{\|x_1\|} - \frac{x_2}{\|x_2\|} \right)^T \left(\frac{x_1}{\|x_2\|} - \frac{x_1}{\|x_1\|} \right) \quad (4.22)$$

$$= \frac{\|x_1\|}{\|x_2\|} - 1 - \frac{x_2^T x_1}{\|x_2\|^2} + \frac{x_2^T x_1}{\|x_1\| \|x_2\|} \quad (4.23)$$

$$= \left(\frac{\|x_1\|}{\|x_2\|} - 1 \right) (1 - \cos \theta) \geq 0 \quad (4.24)$$

where we have used $\cos \theta = \frac{x_2^T x_1}{\|x_1\| \|x_2\|}$.

The proof of the second inequality is more immediate, since $\frac{\gamma_2}{\gamma} \geq 1$ we have $\|a + b\| \leq \left\| \frac{x_1}{\gamma} - \frac{x_2}{\gamma} \right\|$. A graphical interpretation of this lemma is presented in Figure 4.4. $\square$

Now we use Lemma 10 to prove Theorem 12.

Proof of Theorem 12. It is clear by the projection operator $\Pi_{X_h}(\cdot)$ that $T(\mathbf{x})$ is a self-map onto itself, so a) is immediate.

To prove part b) we need to show that $T(\mathbf{x})$ is a contraction:

$$\|T(\mathbf{x}_1) - T(\mathbf{x}_2)\| \leq q \|\mathbf{x}_1 - \mathbf{x}_2\| \quad (4.25)$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in X$, and some $q \in (0, 1)$. We have

$$\begin{aligned} \left\| \sqrt{\alpha} \frac{\mathbf{a} + A\mathbf{x}_1}{\|\mathbf{a} + A\mathbf{x}_1\|} - \sqrt{\alpha} \frac{\mathbf{a} + A\mathbf{x}_2}{\|\mathbf{a} + A\mathbf{x}_2\|} \right\| &\stackrel{(i)}{\leq} \sqrt{\alpha} \left\| \frac{\mathbf{a} + A\mathbf{x}_1}{\gamma} - \frac{\mathbf{a} + A\mathbf{x}_2}{\gamma} \right\| \\ &\leq \frac{\sqrt{\alpha}}{\gamma} \|A\| \|\mathbf{x}_1 - \mathbf{x}_2\| = \frac{\sqrt{\alpha}}{\gamma} (|\mathcal{D}| - 1) \|\mathbf{x}_1 - \mathbf{x}_2\| \end{aligned}$$

where in step (i) we have applied the result from Lemma 10 and defined $\gamma = \min_{\mathbf{x} \in X} \|\mathbf{a} + A\mathbf{x}\|$.

At this point, we can infer that (4.25) is satisfied with $q = \frac{\sqrt{\alpha}}{\gamma} (|\mathcal{D}| - 1)$ and hence, $T(\mathbf{x})$ is a contraction mapping in region X if $\frac{\sqrt{\alpha}}{\gamma} (|\mathcal{D}| - 1) < 1$ is satisfied. In order to prove this last condition, we analyze the quadratic form $\|A\mathbf{x} + \mathbf{a}\| > \sqrt{\alpha} (|\mathcal{D}| - 1)$, which is depicted in Figure 4.5. This inequality forms a paraboloid with minimum point in $\mathbf{x} = -A^{-1}\mathbf{a}$ and curvature specified by its Hessian $(A^T A)^{1/2}$. This Hessian matrix has maximum eigenvalue $\lambda_{\max} = (|\mathcal{D}| - 1)$ with associated eigenvector $\mathbf{v}_{\max} = \frac{\mathbf{1}}{\|\mathbf{1}\|}$, and the rest of eigenvalues are equal with value $\lambda_{\min} = 1$. The level curve corresponding to the value of $\sqrt{\alpha} (|\mathcal{D}| - 1)$ is therefore an ellipsoid that grows faster in the direction of $\mathbf{v}_{\max}$. Then, one sufficient condition to satisfy that the mapping T is always a contraction, is to construct the region X outside of the ellipsoid. We define X as the upper halfspace limited by the hyperplane with normal vector

$v_{\max}$ and most distant point $x = A^{-1}\mathbf{a} + \sqrt{\alpha} \frac{\mathbf{1}}{\|\mathbf{1}\|}$. Such region is defined in equation (4.18) and, since X is convex, the euclidean projection of T is well defined. This proves *b*).

The convergence property of the iterative fixed-point equation directly follows from the Banach fixed point theorem [26, Th.11.1.6.], which states that the mapping admits a unique fixed point $\mathbf{x}^* = T(\mathbf{x}^*)$ in $\mathbf{x} \in X$, and that such point can be found as the limit point of the sequence $\mathbf{x}^{k+1} = T(\mathbf{x}^k)$, for any starting point $\mathbf{x}^0 \in X$. This proves *c*).

Finally, to prove *d*), it is sufficient to show that the following problem **P** is feasible:

$$\begin{aligned} & \text{find } \mathbf{x} \\ \mathbf{P} : & \quad \text{s.t. } \mathbf{x} = \sqrt{\alpha} \frac{\mathbf{a} + A\mathbf{x}}{\|\mathbf{a} + A\mathbf{x}\|} \\ & \quad \mathbf{1}^T \mathbf{x} + \frac{1}{|\mathcal{D}|-1} \mathbf{a}^T \mathbf{1} - \sqrt{\alpha|\mathcal{D}|} \geq 0, \end{aligned}$$

which would prove the existence of a point that solves (4.15) and is inside region X . An equivalent formulation of **P** is

$$\begin{aligned} & \text{find } \mathbf{x}, z \\ & \quad \text{s.t. } \sqrt{\alpha} \mathbf{1}^T (\mathbf{a} + A\mathbf{x}) + \left(\frac{1}{|\mathcal{D}|-1} \mathbf{a}^T \mathbf{1} - \sqrt{\alpha|\mathcal{D}|} \right) z \geq 0 \\ & \quad \|\mathbf{A}\mathbf{x} + \mathbf{a}\| = z, \quad z > 0. \end{aligned}$$

If $\frac{1}{|\mathcal{D}|-1} \mathbf{a}^T \mathbf{1} - \sqrt{\alpha|\mathcal{D}|} = 0$ the previous problem is unconditionally feasible. Otherwise, it can be transformed as follows

$$\begin{aligned} & \text{find } \mathbf{y}, z \\ & \quad \text{s.t. } z \geq \frac{-\sqrt{\alpha}}{\frac{1}{|\mathcal{D}|-1} \mathbf{a}^T \mathbf{1} - \sqrt{\alpha|\mathcal{D}|}} \mathbf{1}^T \mathbf{y} \end{aligned} \tag{4.26}$$

$$\|\mathbf{y}\| = z, \quad z > 0. \tag{4.27}$$

Equation (4.26) represents a halfspace defined by a non vertical and non horizontal hyperplane that passes through the origin. Equations in (4.27) represent a positive cone. Since the intersection of these spaces is necessarily nonempty the problem is feasible, and so is **P**. This proves *d*) and concludes the proof of the theorem. $\square$

Fig. 4.4 Graphical interpretation of Lemma 10

Fig. 4.5 Graphical interpretation of Theorem 10

Chapter 5

Conclusions

In [chapter 2](#) we analyzed the conditions that allow QPs with non-overlapping constraints exhibit the $\mathcal{S}$ -property. Our results encompass both QPs with inequality as well as equality quadratic constraints, and admits any number of constraints as long as they do not overlap. To the best of our knowledge, this is the first time in the literature that we can find results affecting the presence of the $\mathcal{S}$ -property for more than a single equality constraint. Interestingly, this may help analyze problems that consider polynomial minimization, boolean regression or cardinality minimization, where all of them include many equality constraints.

The road-map followed in our proof is also novel where we derived new theorems and lemmas applying to semidefinite programs. In particular, we considered the existence of solutions of rank 1 in several problem formulations where the matrices are simultaneously diagonalizable via congruence and, furthermore, we also consider a special case when they are not. We related the existence of these rank constrained solutions to the fulfillment of the $\mathcal{S}$ -property, and established strong duality. The steps and procedure we followed may have an influence in other problems of interest to come.

We also analyzed the limit of QPs with non-overlapping constraints, which have a particular structure that allowed us to prove coercivity. This property is quite powerful to establish existence of solutions due to the famous Weirstrass theorem as we showed in the text, and this result comes regardless of the $\mathcal{S}$ -property role.

We developed two distributed algorithms that solve the QPs optimally whenever the $\mathcal{S}$ -property is satisfied. The first one is based on first order methods and has a straight forward convergence analysis. The second one is based on the NOVA decomposition framework and at the moment we have not proven convergence yet. We have simulated the proposal in a robust least squares case successfully. However, further theoretical results are needed to analyze its convergence properties. Overall, we presented novel theoretical results regarding QPs with separable constraints, and new algorithms to solve them.

In [chapter 3](#) we analyzed the problem of distributed localization using squared ranged measurements. We considered the TL problem and analyzed it from a centralized and distributed perspective. Our first proposal was based on the ADMM decomposition. The scheme solved a non-convex quadratic problem with single quadratic constraint per node and then exchanged the estimates with the whole network. The process was repeated until convergence.

ADMM has the inherent drawback that it requires average consensus updates whenever the network is not fully connected. Modern frameworks such as diffusion or NEXT have been designed to overcome this aspect. After realizing this limitation, we combined both ideas of having a non fully connected network and exact local computations to propose NEXT-Exact. It includes both the convergence speed and computational simplicity of ADMM-Exact, but also is more flexible for networks which are not fully connected.

However, both algorithms optimize using descent directions of the primal problem, and they may converge to suboptimal stationary solutions. Therefore, we decided to study the performance of dual methods that account for optimality. We proposed a centralized dual method as well as a distributed version using our contributions from [chapter 2](#). The main drawback of this approach is that the $\mathcal{S}$ -property is a strong requirement that cannot be guaranteed in advance. As a future line of research, it would be interesting to learn how to overcome this limitation.

Finally, in [chapter 4](#) we presented a demand-side management model that takes into account deviations in the estimated consumption profiles, and analyzed the worst-case effects of these terms. We proposed a scenario where these deviations are calculated by the energy users (yielding a min-max game), which introduced a non-concave maximization problem for all users. This formulation is novel in the sense that it did not assume any kind of monotonicity in the underlying game.

Our method provides estimates of the worst-case energy deviations and aggregate loads of the network, and further considers robust pricing as an effect of both global energy demand, as well as uncertain real-time demand costs of the supply authority. The individual problems can be solved optimally due to the quadratic structure with a single quadratic objective and quadratic constraint. We are able to prove existence of the NE, as well as to propose a novel convergent distributed algorithm to find the equilibrium solution. The proposed algorithm has very little computational complexity, and provides significant monetary savings. We further illustrated the benefit of our robust model giving flexibility to the users within certain margins while providing a robust production price to the energy supplier.

Regarding future lines of research, nonconvex QPs could be further used in game theory, generalizing current results about non-monotone games. It seems that achieving optimality

in games may be easier since only local optimality is required, and the $\mathcal{S}$ -property may not play such an important role as it does in nonlinear programming. These considerations may have implications in zero-sum games, min-max problems and team NEPs with quadratic cost functions, where different agents compete with opposing cost functions and where monotonicity is not present.

It goes without saying, that QPs are still a very active line of research with huge implications in optimization theory and many applications of interest. Current research lines explore better bounds than semidefinite relaxations from quadratic problems using semi-Lagrangian formulations. For the time being, these include only nonconvex inequality constraints, but some work to include equality constraints may be useful since they appear naturally in many problems as we have already discussed.

Finally, we already commented on the relation that exists between centralized formulations and distributed equivalent problems. It would be very useful to find equivalent representations of nonconvex problems that could be solved using a distributed form. We observed that even though two problems may be completely equivalent, sharing the same stationary points and optimal solutions, one form may exhibit the $\mathcal{S}$ -property while the other one may not. Our idea consists of understanding better this circumstance in order to design schemes that would not have this drawback.

References

- [1] James Aspnes, David Goldenberg, and Y. Richard Yang, “On the computational complexity of sensor network localization,” in *Algorithmic Aspects of Wireless Sensor Networks*. July 2004, pp. 32–44, Springer, doi:10.1007/978-3-540-27820-7_5.
- [2] Italo Atzeni, Luis G. Ordóñez, Gesualdo Scutari, Daniel P. Palomar, and Javier Rodríguez Fonollosa, “Day-ahead bidding strategies for demand-side expected cost minimization,” in *IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, Nov 2012, pp. 91–96, doi:10.1109/SmartGridComm.2012.6485965.
- [3] Italo Atzeni, Luis G. Ordóñez, Gesualdo Scutari, Daniel P. Palomar, and Javier Rodríguez Fonollosa, “Demand-side management via distributed energy generation and storage optimization,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 866–876, June 2013, doi:10.1109/TSG.2012.2206060.
- [4] Amir Beck and Yonina C. Eldar, “Strong duality in nonconvex quadratic optimization with two quadratic constraints,” *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 844–860, 2006, doi:10.1137/050644471.
- [5] Amir Beck, Peter Stoica, and Jian Li, “Exact and approximate solutions of source localization problems,” *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, May 2008, doi:10.1109/TSP.2007.909342.
- [6] Amir Beck and Marc Teboulle, “Global optimality conditions for quadratic optimization problems with binary constraints,” *SIAM Journal on Optimization*, vol. 11, no. 1, pp. 179–188, 2000, doi:10.1137/S1052623498336930.
- [7] Mats Bengtsson and Björn Ottersten, “Optimal and suboptimal transmit beamforming,” in *Handbook of Antennas in Wireless Communications*, Lal Chand Godara, Ed., pp. 568–600. CRC Press, 2001, doi:10.1201/9781420042146.
- [8] Dimitri P. Bertsekas, *Nonlinear programming*, Athena Scientific, 2nd edition edition, Sept. 1999, <http://athenasc.com/nonlinbook.html>.
- [9] Dimitri P. Bertsekas, *Convex Optimization Theory*, Athena Scientific, 2009, <http://www.athenasc.com/convexduality.html>.
- [10] Dimitri P. Bertsekas and John N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena scientific optimization and computation series. Athena Scientific, 1997, <http://athenasc.com/pdcbook.html>.

- [11] Pascal Bianchi and Jérémie Jakubowicz, “Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization,” *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, Feb 2013, doi:10.1109/TAC.2012.2209984.
- [12] Pratik Biswas, Tzu-Chen Lian, Ta-Chung Wang, and Yinyu Ye, “Semidefinite programming based algorithms for sensor network localization,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 188–220, 2006, doi:10.1145/1149283.1149286.
- [13] Immanuel M. Bomze, “Copositive relaxation beats lagrangian dual bounds in quadratically and linearly constrained quadratic optimization problems,” *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1249–1275, 2015, doi:10.1137/140987997.
- [14] Stephen Boyd, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends in Machine Learning*, vol. 3, no. 1, 2010, doi:10.1561/22000000016.
- [15] Stephen Boyd and Lieven Vandenbergh, *Convex optimization*, Cambridge University Press, 2004, doi:10.1017/CBO9780511804441.
- [16] Stephen Boyd, Lin Xiao, and Almir Mutapcic, “Subgradient methods,” Technical report, Stanford University, Oct. 2003, https://web.stanford.edu/class/ee392o/subgrad_method.pdf (accessed 2017-06-01).
- [17] Louis Brickman, “On the field of values of a matrix,” *Proceedings of the American Mathematical Society*, vol. 12, no. 1, pp. 61–66, 1961, doi:10.2307/2034125.
- [18] Loris Cannelli, Francisco Facchinei, Vyacheslav Kungurtsev, and Gesualdo Scutari, “Asynchronous parallel algorithms for nonconvex big-data optimization. Part I: Model and convergence,” July 2016, arXiv:1607.04818.
- [19] Maria Rosa Celis, John E. Dennis, and Richard A. Tapia, “A trust region strategy for equality constrained optimization,” *SIAM Conference on Numerical optimization*, June 1984, https://www.researchgate.net/publication/235172977_A_Trust_Region_Strategy_for_Equality_Constrained_Optimization (accessed 2017-06-01).
- [20] Jianshu Chen and Ali H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012, doi:10.1109/TSP.2012.2198470.
- [21] Jianshu Chen, Zaid J. Towfic, and Ali H. Sayed, “Dictionary learning over distributed models,” *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, Feb. 2015, doi:10.1109/TSP.2014.2385045.
- [22] Antonio J. Conejo, Juan M. Morales, and Luis Baringo, “Real-time demand response model,” *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 236–242, Dec 2010, doi:10.1109/TSG.2010.2078843.
- [23] Jose A. Costa, Neal Patwari, and Alfred O. Hero III, “Distributed weighted-multidimensional scaling for node localization in sensor networks,” *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 39–64, Feb. 2006, doi:10.1145/1138127.1138129.

- [24] K. Cumanan, R. Krishna, Vinod Sharma, and S. Lambotharan, “Robust interference control techniques for multiuser cognitive radios using worst-case performance optimization,” in *Asilomar Conference on Signals, Systems and Computers*, Oct. 2008, doi:10.1109/ACSSC.2008.5074429.
- [25] Alexandre d’Aspremont and Stephen Boyd, “Relaxations and randomized methods for nonconvex QCQPs,” Technical report, Stanford University, 2003, <https://web.stanford.edu/class/ee364b/lectures/relaxations.pdf> (accessed 2017-06-01).
- [26] Kenneth R. Davidson and Allan P. Donsig, *Real Analysis and Applications: Theory in Practice*, Springer, Oct. 2010, doi:10.1007/978-0-387-98098-0.
- [27] Paolo Di Lorenzo and Gesualdo Scutari, “NEXT: In-network nonconvex optimization,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, June 2016, doi:10.1109/TSIPN.2016.2524588.
- [28] Lloyd L Dines, “On the mapping of quadratic forms,” *Bulletin of the American Mathematical Society*, vol. 47, no. 6, pp. 494–498, 1941, doi:10.1090/S0002-9904-1941-07494-X.
- [29] Laurence Charles Ward Dixon and George Philip Szegö, “The global optimization problem: an introduction,” in *Towards Global Optimisation 2*, Laurence Charles Ward Dixon and George Philip Szegö, Eds., pp. 1–15. North-Holland Publishing Company, Amsterdam, 1978.
- [30] Lance Doherty, Kristofer S. J. Pister, and Laurent El Ghaoui, “Convex position estimation in wireless sensor networks,” in *Twentieth Annual Joint Conference of the IEEE Computer and Communications Society*, 2001, vol. 3, pp. 1655–1663, doi:10.1109/INFCOM.2001.916662.
- [31] Laurent El Ghaoui and Hervé Lebret, “Robust solutions to least-squares problems with uncertain data,” *SIAM Journal on Matrix Analysis and Applications*, vol. 18, no. 4, pp. 1035–1064, 1997, doi:10.1137/S0895479896298130.
- [32] Tolga Eren, David K. Goldenberg, Walter Whiteley, Y. Richard Yang, A. Stephen Morse, Brian D. O. Anderson, and Peter N. Belhumeur, “Rigidity, computation, and randomization in network localization,” in *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, March 2004, vol. 4, pp. 2673–2684 vol.4, doi:10.1109/INFCOM.2004.1354686.
- [33] Tomaso Erseghe, “A distributed and maximum-likelihood sensor network localization algorithm based upon a nonconvex problem formulation,” *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 4, pp. 247–258, Dec. 2015, doi:10.1109/TSIPN.2015.2483321.
- [34] Francisco Facchinei and Christian Kanzow, “Generalized Nash equilibrium problems,” *Annals of Operations Research*, vol. 175, no. 1, pp. 177–211, Mar. 2010, doi:10.1007/s10479-009-0653-x.
- [35] Francisco Facchinei and Jong-Shi Pang, “Nash equilibria: the variational approach,” in *Convex optimization in signal processing and communications*. Cambridge University Press, Dec. 2009, doi:10.1017/CBO9780511804458.

- [36] Francisco Facchinei, Jong-Shi Pang, and Gesualdo Scutari, “Non-cooperative games with minmax objectives,” *Computational Optimization and Applications*, vol. 59, no. 1-2, pp. 85–112, Mar. 2014, doi:10.1007/s10589-014-9642-3.
- [37] Francisco Facchinei, Veronica Piccialli, and Marco Sciandrone, “Decomposition algorithms for generalized potential games,” *Computational Optimization and Applications*, vol. 50, no. 2, pp. 237–262, 2011, doi:10.1007/s10589-010-9331-9.
- [38] Francisco Facchinei, Gesualdo Scutari, and Simone Sagratella, “Parallel selective algorithms for nonconvex big data optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 7, pp. 1874–1889, April 2015, doi:10.1109/TSP.2015.2399858.
- [39] Eric Feron, “Nonconvex quadratic programming, semidefinite relaxations and randomization algorithms in information and decision systems,” in *System Theory: Modeling, Analysis and Control*, Theodore E. Djaferis and Irvin C. Schick, Eds., pp. 255–274. Springer, 2000, doi:10.1007/978-1-4615-5223-9_19.
- [40] Baris Fidan, Soura Dasgupta, and Brian D. O. Anderson, “Guaranteeing practical convergence in algorithms for sensor and source localization,” *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4458–4469, 2008, doi:10.1109/TSP.2008.924138.
- [41] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman & Co., 1979, <http://dl.acm.org/citation.cfm?id=574848>.
- [42] Michael Grant and Stephen Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [43] Magnus R. Hestenes, “Pairs of quadratic forms,” *Linear Algebra and its Applications*, vol. 1, no. 3, pp. 397–407, 1968, doi:10.1016/0024-3795(68)90016-5.
- [44] Magnus R. Hestenes, “Multiplier and gradient methods,” *Journal of Optimization Theory and Applications*, vol. 4, pp. 302–320, 1969, doi:10.1007/bf00927673.
- [45] Jean-Baptiste Hiriart-Urruty, “Global optimality conditions in maximizing a convex quadratic function under convex quadratic constraints,” *Journal of Global Optimization*, vol. 21, no. 4, pp. 443–453, Dec. 2001, doi:10.1023/A:1012752110010.
- [46] Jean-Baptiste Hiriart-Urruty and Mounir Torki, “Permanently going back and forth between the “quadratic world” and the “convexity world” in optimization,” *Applied Mathematics and Optimization*, vol. 45, no. 2, pp. 169–184, 2002, doi:10.1007/s00245-001-0034-6.
- [47] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn, “Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems,” *SIAM Journal on Optimization*, vol. 26, no. 1, pp. 337–364, 2016, doi:10.1137/140990309.
- [48] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, 2012, doi:10.1017/CBO9781139020411.

- [49] Yongwei Huang and Daniel P. Palomar, "Rank-constrained separable semidefinite programming with applications to optimal beamforming," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 664–678, Feb 2010, doi:10.1109/TSP.2009.2031732.
- [50] David H. Jacobson, "A generalization of finsler's theorem for quadratic inequalities and equalities," *Quaestiones Mathematicae*, vol. 1, no. 1, pp. 19–28, 1976, doi:10.1080/16073606.1976.9632513.
- [51] Vaithilingam Jeyakumar, G. M. Lee, and Guoyin Y. Li, "Alternative theorems for quadratic inequality systems and global quadratic optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 983–1001, 2009, doi:10.1137/080736090.
- [52] Vaithilingam Jeyakumar, Alexander M. Rubinov, and Z.Y. Wu, "Non-convex quadratic minimization problems with quadratic constraints: global optimality conditions," *Mathematical Programming*, vol. 110, no. 3, pp. 521–541, Sept. 2007, doi:10.1007/s10107-006-0012-5.
- [53] Lanxin Lin and H.C. So, "Best linear unbiased estimator algorithm for received signal strength based localization," in *European Signal Processing Conference*, 2011, pp. 1989–1993, <http://ieeexplore.ieee.org/document/7074178/>.
- [54] Kenneth Wing Kin Lui, Wing-Kin Ma, H. C. So, and Frankie Kit Wing Chan, "Semidefinite programming algorithms for sensor network node localization with uncertainties in anchor positions and/or propagation speed," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 752–763, Feb. 2009, doi:10.1109/TSP.2008.2007916.
- [55] A.-H. Mohsenian-Rad, V.W.S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010, doi:10.1109/TSG.2010.2089069.
- [56] Jorge J Moré, "Generalizations of the trust region problem," *Optimization methods and Software*, vol. 2, no. 3-4, pp. 189–209, 1993, doi:10.1080/10556789308805542.
- [57] Angelia Nedic and Asuman Ozdaglar, "Cooperative distributed multi-agent," in *Convex Optimization in Signal Processing and Communications*, Daniel P. Palomar and Yonina C. Eldar, Eds. Cambridge University Press, 2009, doi:10.1017/CBO9780511804458.
- [58] Angelia Nedic, Asuman Ozdaglar, and Pablo A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010, doi:10.1109/TAC.2010.2041686.
- [59] Thang Van Nguyen, Youngmin Jeong, Hyundong Shin, and Moe Z. Win, "Least square cooperative localization," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1318–1330, April 2015, doi:10.1109/TVT.2015.2398874.
- [60] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani, *Algorithmic game theory*, Cambridge University Press, Sept. 2007.
- [61] Kazuo Nishimura and James Friedman, "Existence of Nash equilibrium in n person games without quasi-concavity," *International Economic Review*, vol. 22, no. 3, pp. 637–648, Oct. 1981, doi:10.2307/2526164.

- [62] Imre Pólik and Tamás Terlaky, “A survey of the S-lemma,” *SIAM review*, vol. 49, no. 3, pp. 371–418, 2007, doi:10.1137/S003614450444614X.
- [63] Boris T. Polyak, “Convexity of quadratic transformations and its use in control and optimization,” *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 553–583, 1998, doi:10.1023/A:1021798932766.
- [64] Ting Kei Pong, “Edge-based semidefinite programming relaxation of sensor network localization with lower bound constraints,” *Computational Optimization and Applications*, vol. 53, no. 1, pp. 23–44, 2012, doi:10.1007/s10589-011-9447-6.
- [65] Michael J. D. Powell, “A method for nonlinear constraints in minimization problems,” in *Optimization*, R. Fletcher, Ed. Academic Press, 1969.
- [66] W. Saad, Z. Han, H.V. Poor, and T. Basar, “Game-theoretic methods for the smart grid: An overview of microgrid systems, demand-side management, and smart grid communications,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 86–105, Sept. 2012, doi:10.1109/MSP.2012.2186410.
- [67] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V.W.S. Wong, and J. Jatskevich, “Optimal real-time pricing algorithm based on utility maximization for smart grid,” in *International Conference on Smart Grid Communications (SmartGridComm)*, Gaithersburg, MD, Oct 2010, pp. 415–420, doi:10.1109/SMARTGRID.2010.5622077.
- [68] Ali H. Sayed, Sheng-Yuan Tu, Jianshu Chen, Xiaochuan Zhao, and Zaid J. Towfic, “Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, May 2013, doi:10.1109/MSP.2012.2231991.
- [69] Gesualdo Scutari, Francisco Facchinei, Lorenzo Lampariello, and Peiran Song, “Distributed methods for constrained nonconvex multi-agent optimization-part I: Theory,” Jan. 2016, arXiv:1410.4754.
- [70] Gesualdo Scutari, Francisco Facchinei, Peiran Song, Daniel P. Palomar, and Jong-Shi Pang, “Decomposition by partial linearization: Parallel optimization of multi-agent systems,” *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 641–656, Feb. 2014, doi:10.1109/TSP.2013.2293126.
- [71] Gesualdo Scutari, Daniel P. Palomar, Francisco Facchinei, and Jong-Shi Pang, “Monotone games for cognitive radio systems,” in *Distributed Decision Making and Control*, Rolf Johansson and Anders Rantzer, Eds., pp. 83–112. Springer, 2012, doi:10.1007/978-1-4471-2265-4_4.
- [72] Xiufang Shi, Guoqiang Mao, Brian D. O. Anderson, Zaiyue Yang, and Jiming Chen, “Robust localization using range measurements with unknown and bounded errors,” Jan. 2017, arXiv:1701.00900.
- [73] Andrea Simonetto and Geert Leus, “Distributed maximum likelihood sensor network localization,” *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1424–1437, March 2014, doi:10.1109/TSP.2014.2302746.

- [74] Anthony Man-Cho So and Yinyu Ye, “Theory of semidefinite programming for sensor network localization,” *Mathematical Programming*, vol. 109, no. 2, pp. 367–384, 2007, doi:10.1007/s10107-006-0040-1.
- [75] Claudia Soares, João Xavier, and João Gomes, “Simple and fast convex relaxation method for cooperative localization in sensor networks using range measurements,” *IEEE Transactions on Signal Processing*, vol. 63, no. 17, pp. 4532–4543, Sept. 2015, doi:10.1109/TSP.2015.2454853.
- [76] Seshan Srirangarajan, Ahmed H. Tewfik, and Zhi-Quan Luo, “Distributed sensor network localization using socp relaxation,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4886–4895, Dec. 2008, doi:10.1109/T-WC.2008.070241.
- [77] Kunal Srivastava and Angelia Nedic, “Distributed asynchronous constrained stochastic optimization,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 772–790, 2011, doi:10.1109/JSTSP.2011.2118740.
- [78] Paula Tarrío, Ana M. Bernardos, Juan A. Besada, and Jose R. Casar, “A new positioning technique for RSS-based localization based on a weighted least squares estimator,” in *IEEE International Symposium on Wireless Communication Systems*, 2008, <http://ieeexplore.ieee.org/document/4726133/>.
- [79] Wayes Tushar, Bo Chai, Chau Yuen, David B. Smith, and H. Vincent Poor, “Energy management for a user interactive smart community: A Stackelberg game approach,” in *IEEE Innovative Smart Grid Technologies*, May 2014, pp. 709–714, doi:10.1109/ISGT-Asia.2014.6873879.
- [80] Jiaheng Wang, Gesualdo Scutari, and Daniel P. Palomar, “Robust MIMO cognitive radio via game theory,” *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1183–1201, 2011, doi:10.1109/TSP.2010.2092773.
- [81] Yu Wang, Wotao Yin, and Jinshan Zeng, “Global convergence of ADMM in nonconvex nonsmooth optimization,” Nov. 2015, arXiv:1511.06324.
- [82] Yunpeng Wang, Walid Saad, Narayan B. Mandayam, and H. Vincent Poor, “Integrating energy storage into the smart grid: A prospect theoretic approach,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 7779–7783, doi:10.1109/ICASSP.2014.6855114.
- [83] Zizhuo Wang, Song Zheng, Yinyu Ye, and Stephen Boyd, “Further relaxations of the semidefinite programming approach to sensor network localization,” *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 655–673, 2008, doi:10.1137/060669395.
- [84] Lin Xiao, Stephen Boyd, and Seung-Jean Kim, “Distributed average consensus with least-mean-square deviation,” *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007, doi:10.1016/j.jpdc.2006.08.010.
- [85] Valery Yakubovich, “S-procedure in nonlinear control theory,” *Vestnik Leningrad University*, vol. 1, pp. 62–77, 1971.
- [86] Yinyu Ye and Shuzhong Zhang, “New results on quadratic minimization,” *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 245–267, 2003, doi:10.1137/S105262340139001X.