



**HAL**  
open science

# Distributed methods for resource allocation: a passivity based approach

German Dario Obando Bravo

## ► To cite this version:

German Dario Obando Bravo. Distributed methods for resource allocation: a passivity based approach. Automatic Control Engineering. Ecole des Mines de Nantes; Universidad de los Andes (Bogotá), 2015. English. NNT: 2015EMNA0174 . tel-01235329

**HAL Id: tel-01235329**

**<https://theses.hal.science/tel-01235329>**

Submitted on 30 Nov 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Thèse de Doctorat

**Germán Darío OBANDO BRAVO**

*Mémoire présenté en vue de l'obtention du  
grade de Docteur de l'École des Mines de Nantes*

*Docteur de l'Universidad de los Andes, Bogota, Colombie*

*sous le label de L'Université Nantes Angers Le Mans*

**École doctorale :** *STIM*

**Discipline :** *Automatique, Productique*

**Spécialité :** *Automatique et Informatique Appliquée*

**Unité de recherche :** *IRCCyN et Universidad de los Andes, Bogota, Colombia*

**Soutenue le** *23 octobre 2015*

**Thèse N° :** *2015 EMNA 0174*

## **Distributed Methods for Resource Allocation: A Passivity-Based Approach**

### **JURY**

Président :	<b>Alexandre DOLGUI</b> , Professeur, École des Mines de Nantes
Rapporteurs :	<b>Carlos OCAMPO-MARTINEZ</b> , Assistant Professor, UPC Barcelona <b>Hervé GUEGUEN</b> , Professeur, Supelec Rennes
Directeur de Thèse Colombie :	<b>Nicanor QUIJANO</b> , Professor, Universidad de Los Andes, Bogota
Directeur de Thèse France :	<b>Jean-Jacques LOISEAU</b> , Directeur de Recherche, IRCCyN
Co-encadrant Colombie :	<b>Alain GAUTHIER</b> , Professor, Universidad de Los Andes, Bogota
Co-encadrant France :	<b>Naly RAKOTO-RAVALONTSALAMA</b> , Maitre Assistant, IRCCyN/EMN

# Distributed Methods for Resource Allocation: A Passivity–Based Approach

Germán Darío Obando Bravo

Universidad de Los Andes, Ecole des Mines de Nantes

A thesis submitted for the degree of

*Philosophiæ Doctor (PhD)*

2015 September

---

Members of Jury:

M. Alexandre DOLGUI (Ecole des Mines de Nantes), Président

M. Nicanor QUIJANO (Universidad de Los Andes), Directeur de These Colombie

M. Alain GAUTHIER (Universidad de Los Andes), co-encadrant Colombie

M. Jean-Jacques LOISEAU (IRCCyN), Directeur de These France

M. Naly RAKOTO (IRCCyN/EMN), co-encadrant France

M. Carlos OCAMPO-MARTINEZ (Universitat Politècnica de Catalunya), rapporteur

M. Hervé GUEGUEN (Supelec Rennes), rapporteur

## Abstract

Since the complexity and scale of systems have been growing in the last years, distributed approaches for control and decision making are becoming more prevalent. This dissertation focuses on an important problem involving distributed control and decision making, the dynamic resource allocation in a network. To address this problem, we explore a consensus-based algorithm that does not require any centralized computation, and that is capable to deal with applications modeled either by dynamical systems or by memoryless functions. The main contribution of our research is to prove, by means of graph theoretical tools and passivity analysis, that the proposed controller asymptotically reaches an optimal solution without the need of full information.

In order to illustrate the relevance of our main result, we address several engineering applications including: distributed control for energy saving in smart buildings, management of the customers of an aggregating entity in a smart grid environment, and development of an exact distributed optimization method that deals with resource allocation problems subject to lower-bound constraints.

Finally, we explore resource allocation techniques based on classic population dynamics models. In order to make them distributed, we introduce the concept of non-well-mixed population dynamics. We show that these dynamics are capable to deal with constrained information structures that are characterized by non-complete graphs. Although the proposed non-well-mixed population dynamics use partial information, they preserve similar properties of their classic counterpart, which uses full information. Specifically, we prove mass conservation and convergence to Nash equilibrium.

---

# Contents

<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Resource Allocation among Networks of Dynamical Systems . . . . .	3
1.2 Resource Allocation Problems with Lower–Bound Constraints . . . . .	5
1.3 Distributed Population Dynamics . . . . .	7
1.4 Research Questions and Outline of the Contributions . . . . .	8
1.4.1 Chapter 2 . . . . .	8
1.4.2 Chapter 3 . . . . .	8
1.4.3 Chapter 4 . . . . .	9
1.4.4 Chapter 5 . . . . .	10
1.4.5 Chapter 6 . . . . .	10
1.4.6 Other Contributions . . . . .	10
<b>2 Preliminaries</b>	<b>13</b>
2.1 Graph Theory . . . . .	13
2.2 Passivity . . . . .	16
2.2.1 Memoryless Systems . . . . .	16
2.2.2 Dynamical Systems . . . . .	16
<b>3 Distributed Resource Allocation Among Dynamic Networks</b>	<b>19</b>
3.1 Network Dynamics . . . . .	19
3.2 Control Objective and Resource Allocation Dynamics . . . . .	21
3.2.1 Control Objective . . . . .	21

## CONTENTS

---

3.2.2	Resource Allocation Dynamics . . . . .	22
3.2.3	Convergence to Output Consensus . . . . .	23
3.2.4	Extension to Memoryless Systems . . . . .	27
3.2.5	Discussion . . . . .	31
3.3	Distributed Building Temperature Control Under Power Constraints . . .	33
3.3.1	Motivation . . . . .	33
3.3.2	Thermal Model of a Building . . . . .	35
3.3.2.1	Thermal Model for a Wall . . . . .	36
3.3.2.2	Thermal Model for a Room . . . . .	37
3.3.2.3	Unified Model . . . . .	37
3.3.3	Problem Statement . . . . .	40
3.3.4	Distributed Control Strategy . . . . .	42
3.3.4.1	Power Constraints . . . . .	46
3.3.4.2	Equilibrium Points . . . . .	46
3.3.5	Simulation Results . . . . .	48
3.3.6	Discussion . . . . .	50
3.4	Optimal Resource Allocation in a Network of Agents . . . . .	51
3.4.1	Inclusion of Inequality Constraints by means of Barrier Functions .	52
3.4.2	Initial Feasible Solution . . . . .	53
3.4.3	Application to Electric Load Aggregation for Demand Response .	54
3.4.4	Discussion . . . . .	57
<b>4</b>	<b>Optimal Resource Allocation with Lower-Bound Constraints</b>	<b>59</b>
4.1	Problem Statement . . . . .	59
4.2	Center-Free Resource Allocation Algorithm . . . . .	61
4.2.1	Resource Allocation Among a Subset of Nodes in a Graph . . . . .	61
4.2.1.1	Feasibility . . . . .	63
4.2.1.2	Equilibrium Point . . . . .	63
4.2.1.3	Convergence Analysis . . . . .	64
4.2.2	Optimal Resource Allocation with Lower Bounds . . . . .	66
4.3	Simulation Results and Comparison . . . . .	71
4.3.1	Distributed Interior Point (DIP) Method . . . . .	71
4.3.2	Distributed Replicator Dynamics (DRD) . . . . .	72



4.3.3	Distributed Interior Point Method with Exact Barrier Functions (DIPE) . . . . .	72
4.3.4	Comparison . . . . .	73
4.4	Application to Optimal Chiller Loading . . . . .	75
4.5	Discussion . . . . .	78
<b>5</b>	<b>Distributed Population Dynamics<sup>1</sup></b>	<b>81</b>
5.1	Motivation . . . . .	81
5.2	Preliminaries . . . . .	83
5.2.1	Graph Representation of Population Interactions . . . . .	83
5.2.2	Population Games . . . . .	83
5.3	General dynamics on graphs . . . . .	85
5.3.1	Mean Dynamics . . . . .	87
5.3.2	Distributed Population Dynamics . . . . .	88
5.3.2.1	Distributed Replicator Dynamics (DRD) . . . . .	88
5.3.2.2	Distributed Smith Dynamics (DSD) . . . . .	88
5.3.2.3	Distributed Logit Dynamics (DLD) . . . . .	89
5.3.2.4	Distributed Projection Dynamics (DPD) . . . . .	89
5.4	Invariant Set and Stability Analysis . . . . .	89
5.4.1	Invariant Set Analysis . . . . .	89
5.4.2	Stability Analysis . . . . .	91
5.5	Illustrative Examples . . . . .	95
5.5.1	Solving a Distributed Optimization Problem . . . . .	96
5.5.2	Classic Population Games: Bad Rock–Paper–Scissors with a Twin	99
5.5.3	Distributed Control of Dynamical Systems . . . . .	100
5.6	Discussion . . . . .	103
<b>6</b>	<b>Conclusions and Future Directions</b>	<b>105</b>
6.1	Distributed Resource Allocation Among Dynamic Networks . . . . .	105
6.2	Optimal Resource Allocation with Lower–Bound Constraints . . . . .	106
6.3	Distributed Population Dynamics . . . . .	106
	<b>References</b>	<b>109</b>

---

<sup>1</sup>G. Obando and J. Barreiro–Gómez contributed equally to this chapter. Results are reported in [1].

## CONTENTS

---

# List of Figures

3.1	A large-scale system controlled by a network of agents. . . . .	20
3.2	Feedback interconnection of systems (3.2) and (3.4). . . . .	24
3.3	$n$ rooms arranged in a row. . . . .	36
3.4	Schematic representation of a room and a wall. The leftmost figure illustrates the cross section of a wall divided into layers, while the rightmost one shows the top view of a room surrounded by walls. . . . .	36
3.5	a. Ambient temperature during a day (7h–20h). b. Temperatures of the building’s rooms. Rooms 1–17 (red), rooms 18–34 (yellow), rooms 35–50 (blue). c. Heating Power allocated in the building’s rooms. d. Variance of the error across the rooms. Distributed controller (solid line), PI controllers (dashed line). . . . .	49
3.6	Cost function for each aggregator’s client. . . . .	56
3.7	Communication network with single-path topology. . . . .	57
3.8	Evolution of the decentralized resource allocation algorithm. . . . .	58
3.9	Evolution of the total cost perceived by the aggregator’s clients. . . . .	58
4.1	Single path topology for $n$ nodes. . . . .	73
4.2	Decoupled chilled-water plant with $n$ chillers (adapted from [2]). . . . .	75
4.3	<b>a.</b> $P_i-T_i$ curves for each chiller, <b>b.</b> Evolution of supply temperatures and total power consumed by the chillers, $C_L = 8862.8$ kW, <b>c.</b> Evolution of supply temperatures and total power consumed by the chillers, $C_L = 5908.6$ kW. . . . .	79
5.1	a. Well-mixed population. b. Non-well-mixed population. . . . .	86

## LIST OF FIGURES

---

5.2	Graph representation of: a. The well-mixed population in Figure 5.1a. b. The non-well-mixed population in Figure 5.1b. Nodes 1,2, and 3 correspond, respectively, to strategies “circle,” “triangle,” and “square” of the proposed example. . . . .	86
5.3	Non-complete graph for the distributed optimization example. . . . .	96
5.4	Evolution under different distributed population dynamics: states (1st row), fitness functions (2nd row), objective function (3rd row), equality constraint (4th row). . . . .	98
5.5	Distributed (up) and classic (bottom) population dynamics applied to bad rock(R)–paper(P)–scissors(S) with a twin(T). . . . .	100
5.6	Simple drinking water system with a unique resource and unknown demands.	101
5.7	Demand profile ( $d_1, d_2, d_3$ and $d_4$ ) during 2 days for the 4 tanks case. . . . .	101
5.8	System states evolution for a controller with full information. . . . .	102
5.9	System states evolution for a distributed population dynamics based controller. . . . .	103

# List of Tables

- 3.1 Utility functions' parameters . . . . . 57
- 4.1 Distributed Algorithms' Performance . . . . . 75
- 4.2 Chillers' parameters . . . . . 78
- 5.1 Some revision protocols and their corresponding population dynamics [3] . 85

## LIST OF TABLES

---

# 1

## Introduction

Control and decision making are required in many engineering problems, especially in those that seek to optimize certain criteria, such as increasing economic benefits, improving the quality of service, or reducing energy consumption. In several applications, traditional centralized approaches for control and decision making are no longer appropriate since the complexity and scale of systems have increased in the last years. Therefore, distributed schemes are becoming more prevalent [4]. The growing interest in these techniques has led to the development of a large number of distributed methods (e.g., see [5, 6, 7, 8, 9, 10] and the references therein), among which we find the algorithms based on multi-agent systems, where the agents<sup>1</sup> make decisions based on local information and coordinate with each other to obtain a desirable global behavior; for instance, the optimization of an overall objective function. An important issue within this field is the dynamic resource allocation over networks of agents. This problem arises when we have a limited amount of a certain resource (electric power, computing capacity, execution time), and it is necessary to establish an optimal distribution policy between some entities (e.g., loads, processors, controllers) that are managed by a set of agents connected by a communication network. Resource allocation is widely used (a comprehensive survey on this topic can be found in [11]), and it has been applied in several fields, such as economics [12], smart energy systems [13], distributed computation [14], communication systems [15], and virus spread mitigation [16].

An extensive literature describes distributed methods for resource allocation based on multi-agent systems: more traditional approaches use decomposition techniques [17],

---

<sup>1</sup>Entities or elements that are capable to make decisions.

## 1. INTRODUCTION

---

where the basic idea is to exploit the structure of the overall problem in order to divide it into subproblems that are then assigned to each agent in the network. Other classical techniques are the so-called incremental gradient schemes [18], where the solution moves following the direction of one component of the objective function's gradient by iteration, and this search direction is alternate by using a token ring scheme. We can also find some methodologies based on heuristics, such as those inspired by market mechanisms [19], [20], which use concepts from economic theory to model the interactions between the participants of the resource allocation process. More recently, there has been a widespread interest in game theoretical approaches to solve this kind of problems. Current developments on this field can be found in [21], [22] for the non-cooperative case, and in [23] for the cooperative one. Among all these methods, the appropriate coordination of agents is a crucial issue because it avoids converging to suboptimal solutions. To ensure this coordination, some methods require the inclusion of a centralized agent; for instance, in classical decomposition techniques, the Lagrange multiplier related to the "price" of the resource is centrally adjusted to get the optimum. Nonetheless, in some real situations, especially when we have a large number of agents and the amount of communications allowed between them is strongly limited, the implementation of a centralized coordinator is impractical, too expensive, or even infeasible [7]. By contrast, other methods are fully decentralized (e.g., [24], [25]). These decentralized methods exploit the communication capabilities of each agent to coordinate its decisions based on the information received from its neighbors. Moreover, decentralized methodologies have important advantages, among which we highlight the increased autonomy and resilience of the whole system in the face of possible failures.

This research focuses on distributed resource allocation methods among networks of agents that do not require a central coordinator. Specifically, we are interested in consensus-based approaches since their low computational requirements and appealing results on robustness, convergence rate, and flexibility (see, e.g., [26, 27, 28, 29, 30]). Regarding consensus-based algorithms for solving resource allocation problems, several works have been conducted. Most of them deals with static optimization, where the objective is to minimize a global cost function that depends directly on the resource allocated to the entities that comprise the whole system. Results on optimality and convergence of these resource allocation methods are not new. Indeed, the first reported algorithm, due to Ho et al., dates back to 1980 [24]. Since then, a large number of



variations of the cited algorithm and contributions for analyzing the proposed methods have been introduced in the literature. Most of these results are largely influenced by the progress of consensus theory [28, 31, 32, 33, 34]. We highlight the work in [35], where the authors use linear matrix inequalities in order to derive sufficient conditions that guarantee the convergence of the algorithm proposed in [24] to the optimal solution of a given problem. These conditions have been recently relaxed in [36] by means of convex optimization theory. Other related results can be found in [37] for non-smooth objective functions, and in [38] for non-convex objective functions. Although resource allocation for static problems is a well studied area, there still exists a research gap when dealing with dynamic problems, especially if the dynamics are modeled by non-linear differential equations.

### 1.1 Resource Allocation among Networks of Dynamical Systems

For the static case, a resource allocation process is equivalent to an equalization mechanism among the marginal costs associated with each entity that conforms the entire system (this equivalence follows from the application of the Karush–Kuhn–Tucker first order necessary conditions). Therefore, if we model each entity as a memoryless system whose input is the resource allocated in the entity, and whose output is equal to the associated marginal cost, a resource allocation algorithm can be viewed as an output-consensus protocol. Following the same ideas, if we replace the memoryless model by a dynamical model, it is possible to extend the resource allocation concept to dynamic problems, i.e., problems in which the desired objective depends on the outputs of a collection of dynamical systems that are influenced by the resource allocated to the entities. We can find a large number of applications that fits into this framework. For instance, building temperature regulation [39], control of water distribution systems [40], bandwidth allocation [41] and access control [42, 43] in communication networks, dispatch of distributed generators for frequency regulation [44], energy saving in lighting systems [22], control for urban drainage systems [45], and so forth.

From the above description, we formulate the resource allocation problem addressed in this dissertation as follows: *given a set of dynamical systems, design a distributed*

## 1. INTRODUCTION

---

*resource allocation policy that drives the systems' outputs to consensus.* Thus, our problem is twofold. On the one hand, we require an output-consensus algorithm. On the other hand, this algorithm must satisfy a given resource constraint (i.e., the constraint associated with the limited resource that has to be allocated). Although the literature on output-consensus algorithms is rich (see, e.g., [46, 47, 48, 49, 50, 51, 52, 53], and the references therein), few of these algorithms address the required resource constraint. Among those that take into account the considered constraint, we highlight the methods based on population dynamics [54, 55, 56, 57], where the resource is related to an invariant population mass. Nevertheless, these methods require a specific communication architecture (either the inclusion of a centralized coordinator, or the existence of non-trivial cliques (i.e., complete sub-graphs)). Other algorithms are based on distributed receding horizon techniques [58, 59, 60], where the treatment of the problem dynamics is made by taking a finite sequence of static problems. Even though this formulation is very flexible (e.g., it allows us to include transient performance criteria and other constraints different from the resource limitation), receding horizon algorithms impose a high computational burden since an optimization problem is solved at each algorithm's step. Finally, we find heuristic methods. For instance, market-based control [19, 61, 62, 63]. These heuristic methods generally lack a theoretical support that guarantees the proper achievement of the required goals.

To tackle the resource allocation problem among networks of dynamical systems, we use a continuous-time version of the *center-free* algorithm first introduced in [24] and extended in [35]. This technique belongs to the class of fully-distributed algorithms studied in the seminal work of Tsitsiklis [64]. In the framework proposed by Tsitsiklis, each system of the network is associated with an agent that manages the allocated resource taking into account a given objective. Starting with a feasible allocation, each agent updates the corresponding resource by following a given function that depends on the decisions of neighboring agents. The simple idea behind the method makes it appropriate for applications where the agents have elementary computational capabilities. Moreover, the use of a continuous-time approach leads to a natural interaction of the considered method with dynamical systems modeled by differential equations.

The main contribution of this dissertation is to obtain sufficient conditions that guarantee asymptotical convergence of the distributed resource allocation algorithm to a desired solution, even if the decisions of each agent influence the behavior of a nonlinear

dynamical system. Our results are inspired by the ideas developed in [32], due to the close relationship between the *center-free* algorithm and the continuous-time consensus protocols analyzed in [32]. We notice that convergence conditions can be directly derived from [46, 48] for systems with linear dynamics. Nonetheless, the nonlinear case is more challenging. In order to address the nonlinear case, we use passivity theory [65, 66]. Specifically, we prove that it is possible to drive a passive system (conceptually, a passive system is a multi-port system that can store but cannot generate energy) to a desired state by using a resource-allocation-based controller. Similar results are derived in [49, 67, 68, 69]. However, in these works, resource limitation is not taken into account (a more comprehensive discussion regarding the differences of our contributions and the results given in the cited works is developed in Section 3.2.5).

To illustrate the applicability of our results, we present two engineering problems related to the management of smart energy systems. The first application deals with the control of heating, cooling, and air conditioning (HVAC) systems of large buildings that are not over-designed. In this case, there exists a trade-off between the comfort of the building's occupants and the available heating/cooling power at critical load hours. We propose a distributed approach that maximizes the comfort of the building's occupants under several power constraints. The second application is related to load aggregation for demand response [70]. This problem is motivated by the fact that active participation of demand is a central issue in the smart grid concept. In the proposed approach, participants are encouraged to stay involved in demand response programs, by reducing the impact that they perceive when it is necessary to curtail their electric load. In all the cases, we provide simulation results that illustrate the efficiency of the proposed method.

## 1.2 Resource Allocation Problems with Lower-Bound Constraints

Once the key contribution on the convergence of the distributed resource allocation algorithm is set, we propose a novel technique based on a passive dynamical system, that optimally allocates a certain resource among a subset of nodes in a connected graph. Then, this technique is applied to solve static resource allocation problems with lower bound-constraints. The inclusion of lower-bound constraints is crucial in a large number of practical applications, e.g., in [12, 15, 22], where it is required to capture the

## 1. INTRODUCTION

---

non-negativity of the resource allocated to the different entities that are involved in the problem formulation. We use a passivity-based analysis that relies on the tools previously developed in order to prove that the proposed algorithm asymptotically converges to the optimal solution under some mild assumptions related to the convexity of the cost function, and the connectivity of the graph that represents the communication topology. In order to illustrate our theoretical results, we perform some simulations and compare our method with other techniques reported in the literature.

Although there exists an extensive literature regarding distributed methods for solving static resource allocation problems, this field still attracts considerable research attention [16, 36, 54, 71, 72]. A large number of solution methods belong to the general class of NUM algorithms (a survey can be found in [17]). However, most of these methods require either the inclusion of a centralized agent or the use of restrictive information structures (as it is pointed out in [73]). Other approaches that solve the static resource allocation problem are inspired on the algorithm presented in [35]. The key difference of the approaches based on [35] compared to our technique is that they do not allow the explicit inclusion of lower bounds on the decision variables, unless these lower bounds are added by means of barrier functions (either logarithmic or exact [36]). The problem of using barrier functions is that they can adversely affect the convergence time (in the case of using exact barrier functions) and the accuracy of the solution (in the case of using classic logarithmic barrier functions), especially for large-scale problems [74]. There are other methods that consider lower bound constraints in the problem formulation. For instance, the authors in [72, 75], have developed a decentralized technique based on broadcasting and consensus to optimally distribute a resource considering capacity constraints on each entity in the network. Nonetheless, compared to our algorithm, the approach in [72, 75] is only applicable to quadratic cost functions. On the other hand, authors in [22] propose a novel methodology based on population dynamics. The main drawback of this technique is that its performance is seriously degraded when the number of communication links decreases. We point out the fact that other distributed optimization algorithms can be applied to solve static resource allocation problems, as those presented in [76], [77], and [37]. Nevertheless, the underlying idea in these methods is different from the one used in our work, i.e., the authors in [76], [77], and [37] use consensus steps to refine an estimation of the system state, while in our approach, consensus is used to equalize a quantity that depends on both the marginal cost perceived

by each agent in the network, and the Karush–Kuhn–Tucker (KKT) multiplier related to the corresponding resource’s lower bound. In this regard, it is worth noting that the method studied in this paper requires less computational effort than the methods mentioned above. Finally, there are other techniques based on game theory and mechanism design [78, 79] that decompose and solve the original problem. Nonetheless, they need that each agent broadcasts a variable to all the other agents, i.e., a communication topology given by a complete graph is required. In contrast, the method developed in our dissertation only uses a communication topology given by a connected graph, which generally requires lower infrastructure.

Newly, in order to illustrate the performance of the proposed technique, we present an application that focuses on energy efficiency in smart buildings. More precisely, we present a distributed solution to the optimal chiller loading problem in multiple chiller systems [80], which are widely used in large air-conditioning systems. The goal is to distribute the cooling load among the chillers that comprise the plant to minimize the total amount of power used by them.

### 1.3 Distributed Population Dynamics

Finally, this dissertation explores resource allocation methods based on population dynamics. Population dynamics have been widely used in the design of learning and control systems for networked engineering applications, where the information dependency among elements of the network has become a relevant issue. Classic population dynamics (e.g., replicator, logit choice, Smith, and projection) require full information to evolve to the solution (Nash equilibrium). The main reason is that classic population dynamics are deduced by assuming well-mixed populations, which limits the applications where this theory can be implemented. In this research, we extend the concept of population dynamics for non-well-mixed populations in order to deal with distributed information structures that are characterized by non-complete graphs. Although the distributed population dynamics proposed in this document use partial information, they preserve similar characteristics and properties of their classic counterpart. Specifically, we prove mass conservation and convergence to Nash equilibrium. To illustrate the performance of the proposed dynamics, we show some applications in the solution of optimization problems, classic games, and the design of distributed controllers.

### 1.4 Research Questions and Outline of the Contributions

The purpose of this doctoral thesis can be summarized by the following research questions:

- Q1.** Can consensus-based resource allocation algorithms be extended to networks of nonlinear dynamical systems? Under what conditions, classic results on distributed resource allocation hold for this kind of networks?
- Q2.** How to address lower-bound constraints in static resource allocation problems by using distributed exact algorithms?
- Q2.** Is it possible to modify classic resource allocation methods based on population dynamics in order to avoid the need of full information? What conditions must be satisfied to guarantee that modified population dynamics that use local information preserve the optimality characteristics exhibited by their classic counterpart?

The above questions are tackled in this document as follows.

#### 1.4.1 Chapter 2

In this chapter, we present some basic definitions and preliminary results on algebraic graph theory and passivity theory. We also introduce an original lemma, related to some properties of Laplacian matrices of connected graphs (Lemma 2.1.1), that is a key tool in subsequent analyses.

#### 1.4.2 Chapter 3

**Q1** is addressed in Chapter 3. First, we propose a generalization of the static resource allocation problem for networks of dynamical systems that are modeled by nonlinear differential equations. Then, we show the design process of resource allocation rules based on consensus for this kind of networks. Our main contribution, which is stated in Theorems 3.2.1, 3.2.2, and 3.2.3, is to provide sufficient conditions that guarantee that the set of systems attain a desired global behavior using only local information. These conditions are related to the passivity of the considered systems (either dynamical or memoryless), and the connectivity of the graph that describes the local information structure. Finally, in this chapter, we present some applications of our main result in engineering problems.

### Related Publications:

- G. Obando, N. Quijano, and N. Rakoto–Ravalontsalama, “Distributed Resource Allocation among Dynamical Networks: An Application in Energy Efficient Buildings,” In preparation to be submitted to IET Control Theory and Applications, 2015.
- G. Obando, A. Pantoja, and N. Quijano, “Building Temperature Control based on Population Dynamics,” IEEE Transactions on Control Systems Technology, vol. 22, no. 1, pp. 404–412, 2014.
- G. Obando, N. Quijano, and N. Rakoto–Ravalontsalama, “Distributed building temperature control with power constraints,” In Proceedings of the 13th European Control Conference, pp. 2857–2862, 2014.
- G. Obando, N. Quijano, and N. Rakoto–Ravalontsalama, “Distributed Resource Allocation over Stochastic Networks: An Application in Smart Grids,” To appear in Proceedings of the 2nd Colombian Conference on Automatic Control, 2015.
- G. Obando, N. Quijano, and A. Pantoja, “Population dynamics applied to building energy efficiency,” In Proceedings of the 50th IEEE Conference on Decision and Control, pp. 4777–4782, 2011.

### 1.4.3 Chapter 4

Chapter 4 tackles **Q2**. The first part of this chapter is devoted to develop a distributed protocol (cf., Equation (4.4)) to allocate a certain resource among a subset of nodes in a connected graph. Optimality of this protocol is proven in Proposition 4.2.2 even if the subset of nodes does not define a connected subgraph. The second part of the chapter uses the developed protocol to propose a novel distributed algorithm that solves resource allocation problems with lower-bound constraints (cf., Algorithm 1). Convergence of the output of this algorithm to the optimal solution of the considered problems is guaranteed in Theorem 4.2.1. Finally, we perform a comparison between our algorithm and other state-of-the-art methods. This comparison shows the advantages of using exact approaches (as the one proposed in our algorithm) instead of solutions based on barrier functions.

## 1. INTRODUCTION

---

### Related Publications:

- G. Obando, N. Quijano, and N. Rakoto-Ravalontsalama, “A Center-Free Approach for Resource Allocation with Lower Bounds,” Submitted to the International Journal of Control, 2015.

### 1.4.4 Chapter 5

**Q3** is answered in Chapter 5, where we extend the results in [81] in order to avoid the need of full information of classic population dynamics. Our main contribution is the design of a general method that allows us to deduce several distributed population dynamics (i.e., population dynamics that evolve using only local information). To illustrate our methodology, we present a distributed version of the fundamental population dynamics (those obtained by applying classic revision protocols), i.e., the distributed replicator dynamics, the distributed Smith dynamics, the distributed logit dynamics, and the distributed projection dynamics. Finally, we prove that, although distributed population dynamics do not use full information, they preserve similar properties as their classic counterpart, i.e., simplex invariance (cf., Theorem 5.4.1 and Proposition 5.4.1), and convergence to Nash equilibria (cf., Theorems 5.4.2 and 5.4.3).

### Related Publications:

- J. Barreiro-Gómez, G. Obando, N. Quijano, “Distributed Population Dynamics: Optimization and Control Applications,” Submitted to the IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015.

### 1.4.5 Chapter 6

Conclusions and future directions are discussed in Chapter 6.

### 1.4.6 Other Contributions

The following publications had a significant influence on some of the thesis contributions, but they are not covered in this document.

- G. Obando, J. I. Poveda, and N. Quijano, “Replicator Dynamics Under Perturbations and Time Delays,” Submitted to Mathematics of Control, Signal, and Systems, 2015.



## 1.4 Research Questions and Outline of the Contributions

---

- A. Pantoja, N. Quijano, and G. Obando. “Bioinspired approach for a multizone temperature control” (Book Chapter). Handbook of Biomimetics and Bioinspiration (ISBN 978-9814354950) pp. 817-860. World Scientific.
- J. Barreiro-Gómez, G. Obando, C. Ocampo-Martinez, and N. Quijano, “Making Non-Centralized a Model Predictive Control Scheme by Using Distributed Smith Dynamics”. To appear in proceedings of 5th IFAC Conference on Nonlinear Model Predictive Control, 2015.
- J. Barreiro-Gómez, G. Obando, G. Riano-Briceno, N. Quijano, and C. Ocampo-Martinez, “Decentralized Control for Urban Drainage Systems Via Population Dynamics: Bogota Case Study”. In proceedings of the 14th European Control Conference, 2015.

## 1. INTRODUCTION

---

## 2

# Preliminaries

First, we describe the notation used throughout the document and presents some preliminary results on graph theory and passivity that are used in the proofs of our main contributions.

### 2.1 Graph Theory

In the multi-agent framework considered in this dissertation, we use a graph to model the communication network that allows the agents to coordinate their decisions. A graph is mathematically represented by the triplet  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ , where  $\mathcal{V} = \{1, \dots, n\}$  is the set of nodes,  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges connecting the nodes, and the adjacency matrix  $\mathcal{A}$  is an  $n \times n$  nonnegative matrix that satisfies:  $a_{ij} = 1$  if and only if  $(i, j) \in \mathcal{E}$ , and  $a_{ij} = 0$  if and only if  $(i, j) \notin \mathcal{E}$ . Each node of the graph corresponds to an agent of the multi-agent system, and the edges represent the available communication channels (i.e.,  $(i, j) \in \mathcal{E}$  if and only if agents  $i$  and  $j$  can share information). We assume that there is no edges connecting a node with itself, i.e.,  $a_{ii} = 0$ , for all  $i \in \mathcal{V}$ ; and that the communication channels are bidirectional, i.e.,  $a_{ij} = a_{ji}$ . The last assumption implies that  $\mathcal{G}$  is undirected. Additionally, we denote by  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ , the set of neighbors of node  $i$ , i.e., the set of nodes that are able to receive/send information from/to node  $i$ .

Let us define the  $n \times n$  matrix  $L(\mathcal{G}) = [l_{ij}]$ , known as the graph Laplacian of  $\mathcal{G}$ , as

## 2. PRELIMINARIES

---

follows:

$$l_{ij} = \begin{cases} \sum_{j \in \mathcal{V}} a_{ij} & \text{if } i = j \\ -a_{ij} & \text{if } i \neq j. \end{cases} \quad (2.1)$$

Properties of  $L(\mathcal{G})$  are related to connectivity characteristics of  $\mathcal{G}$  as shown in the following theorem. We remark that a graph  $\mathcal{G}$  is said to be connected if there exists a path connecting any pair of nodes.

**Theorem 2.1.1.** *(Adapted from [82]) An undirected graph  $\mathcal{G}$  of order  $n$  is connected if and only if  $\text{rank}(L(\mathcal{G})) = n - 1$ .*

From Equation (2.1), it can be verified that

$$L(\mathcal{G})\mathbf{1} = \mathbf{0}, \quad (2.2)$$

where  $\mathbf{1} = [1, \dots, 1]^\top$ ,  $\mathbf{0} = [0, \dots, 0]^\top$ . A consequence of this fact is that  $L(\mathcal{G})$  is a singular matrix. However, we can modify  $L(\mathcal{G})$  to obtain a nonsingular matrix as shown in the following lemma, which is the first result of our dissertation.

**Lemma 2.1.1.** *Let  $L^{kr}(\mathcal{G}) \in \mathbb{R}^{(n-1) \times n}$  be the submatrix obtained by removing the  $k$ th row of the graph Laplacian  $L(\mathcal{G})$ , and let  $L^k(\mathcal{G}) \in \mathbb{R}^{(n-1) \times (n-1)}$  be the submatrix obtained by removing the  $k$ th column of  $L^{kr}(\mathcal{G})$ . If  $\mathcal{G}$  is connected, then  $L^k(\mathcal{G})$  is positive definite. Furthermore, the inverse matrix of  $L^k(\mathcal{G})$  satisfies  $(L^k(\mathcal{G}))^{-1} \mathbf{1}_k^{kr} = -\mathbf{1}$ , where  $\mathbf{1}_k^{kr}$  is the  $k$ th column of the matrix  $L^{kr}(\mathcal{G})$ .*

*Proof.* First, notice that  $L(\mathcal{G})$  is a symmetric matrix because  $\mathcal{G}$  is an undirected graph. Moreover, notice that according to Equation (2.1),  $L(\mathcal{G})$  is diagonally dominant with non-negative diagonal entries. The same holds for  $L^k(\mathcal{G})$  since this is a sub-matrix obtained by removing the  $k$ th row and column of  $L(\mathcal{G})$ . Thus, to show that  $L^k(\mathcal{G})$  is positive definite, it is sufficient to prove that  $L^k(\mathcal{G})$  is nonsingular.

According to Theorem 2.1.1, since  $\mathcal{G}$  is connected,  $L(\mathcal{G})$  has exactly  $n - 1$  linearly independent columns (resp. rows). Let us show that the  $k$ th column (resp. row) of  $L(\mathcal{G})$  can be obtained by a linear combination of the other columns (resp. rows), i.e., the  $k$ th column (resp. row) is not linearly independent of the rest of the columns (resp. rows).

Since  $L(\mathcal{G})\mathbf{1} = \mathbf{0}$ , notice that  $l_{ik} = -\sum_{j \in \mathcal{V}, j \neq k} l_{ij}$ , for all  $i \in \mathcal{V}$ , i.e., the  $k$ th column can be obtained by a linear combination of the rest of the columns. Furthermore, since  $L(\mathcal{G})$  is a symmetric matrix, the same occurs with the  $k$ th row. Therefore, the submatrix  $L^k(\mathcal{G})$  is nonsingular since its  $n - 1$  columns (resp. rows) are linearly independent.

Now, let us prove that  $(L^k(\mathcal{G}))^{-1} \mathbf{1}_k^{k_r} = -\mathbf{1}$ . In order to do so, we use the fact that  $(L^k(\mathcal{G}))^{-1} L^k(\mathcal{G}) = I$ , where  $I$  is the identity matrix. Hence, by the definition of matrix multiplication, we have that

$$\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \quad (2.3)$$

where  $l_{ij}^k$  and  $\bar{l}_{ij}^k$  are the elements located in the  $i$ th row and  $j$ th column of the matrices  $L(\mathcal{G})$  and  $(L^k(\mathcal{G}))^{-1}$ , respectively. Thus,

$$\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mi}^k = 1, \text{ for all } i = 1, \dots, n-1. \quad (2.4)$$

Let  $l_{k_m}^{k_r}$  be the  $m$ th entry of the vector  $\mathbf{1}_k^{k_r}$ . Notice that, according to the definition of  $L^k(\mathcal{G})$  and since  $L(\mathcal{G})\mathbf{1} = \mathbf{0}$ ,  $l_{mi}^k = -\sum_{j=1, j \neq i}^{n-1} l_{mj}^k - l_{k_m}^{k_r}$ . Replacing this value in Equation (2.4), we obtain

$$-\sum_{j=1, j \neq i}^{n-1} \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k - \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{k_m}^{k_r} = 1, \text{ for all } i = 1, \dots, n-1.$$

According to Equation (2.3),  $\sum_{j=1, j \neq i}^{n-1} \sum_{m=1}^{n-1} \bar{l}_{im}^k l_{mj}^k = 0$ . This implies that  $\sum_{m=1}^{n-1} \bar{l}_{im}^k l_{k_m}^{k_r} = -1$ , for all  $i = 1, \dots, n-1$ . Therefore,  $(L^k(\mathcal{G}))^{-1} \mathbf{1}_k^{k_r} = -\mathbf{1}$ .  $\square$

Theorem 2.1.1 and Lemma 2.1.1 will be used in the analysis of the methods proposed in this dissertation, which must satisfy the information constraint imposed by the graph  $\mathcal{G}$ . This constraint can be formally stated by means of the concept of distributed map. A distributed map over the graph  $G$  is defined as follows [27].

**Definition 2.1.1.** *Given two sets  $\mathcal{R}, \mathcal{Z}$ , a function  $O : \mathcal{R}^n \rightarrow \mathcal{Z}^n$  is distributed over  $\mathcal{G}$  if there exist functions  $\tilde{\mathcal{O}}_1, \dots, \tilde{\mathcal{O}}_n : \mathcal{R} \times \mathcal{B}(\mathcal{R}) \rightarrow \mathcal{Z}$  ( $\mathcal{B}(\mathcal{R})$  denotes the collection of all subsets of  $\mathcal{R}$ ), such that*

$$\mathcal{O}_i(r_1, \dots, r_n) = \tilde{\mathcal{O}}_i(r_i, \{r_j | j \in \mathcal{N}_i\}),$$

for all  $(r_1, \dots, r_n) \in \mathcal{R}^n$ , and all  $i = 1, \dots, n$ .

Roughly speaking, we say that a map is distributed over the graph  $\mathcal{G}$  if it can be computed by each node using only local information.

### 2.2 Passivity

Passivity theory [66] is a useful tool in the analysis of nonlinear systems. In this section we provide some definitions regarding this field.

#### 2.2.1 Memoryless Systems

Let us consider the following memoryless system

$$y = \vartheta(x), \quad (2.5)$$

where  $\vartheta : \mathbb{R}^n \mapsto \mathbb{R}^n$  is continuous,  $x \in \mathbb{R}^n$  is the system input, and  $y \in \mathbb{R}^n$  is the system output.

**Definition 2.2.1.** (Adapted from [83]) *We say that the memoryless system given in Equation (2.5) is passive if  $x^\top y \geq 0$ , for all  $x \in \mathbb{R}^n$ .*

Passivity concept can be extended to dynamical systems as follows.

#### 2.2.2 Dynamical Systems

Let us consider a dynamical system of the form

$$\begin{aligned} \dot{\xi} &= \zeta(\xi, x) \\ y &= \vartheta(\xi, x), \end{aligned} \quad (2.6)$$

where  $\xi \in \mathbb{R}^m$  is the system state;  $x \in \mathbb{R}^n$  is the system input;  $y \in \mathbb{R}^n$  is the system output;  $\zeta : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}^m$  is locally Lipschitz; and  $\vartheta : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}^n$  is continuous. Furthermore,  $\zeta(0, 0) = 0$  and  $\vartheta(0, 0) = 0$ . Notice that we assume that the number of inputs and outputs of the system is the same.

**Definition 2.2.2.** (Adapted from [83]). *We say that the dynamical system given in Equation (2.6) is passive if there exists a continuously differentiable storage function  $V(\xi) \geq 0$  such that*

$$x^\top y \geq \dot{V}(\xi), \text{ for all } (\xi, x) \in \mathbb{R}^m \times \mathbb{R}^n.$$

*Additionally, we say that this system is:*

- *Lossless if  $x^\top y = \dot{V}(\xi)$ , for all  $(\xi, x) \in \mathbb{R}^m \times \mathbb{R}^n$ .*
- *Strictly passive if  $x^\top y \geq \dot{V}(\xi) + \psi(\xi)$ , for some positive definite function  $\psi$ , and for all  $(\xi, x) \in \mathbb{R}^m \times \mathbb{R}^n$ .*

Passivity theory and Lyapunov stability are closely related. Indeed, there are some results that link passivity characteristics of systems connected in feedback configurations to stability of the closed loop response. In this regard, we notice that the definitions of storage function and Lyapunov function are similar. A key difference is that Lyapunov functions have to be positive definite whereas storage functions only need to be positive semidefinite. Nonetheless, there exists a result due to Krasovskii and LaSalle that relaxes the positive definiteness requirement on Lyapunov functions. This result is described as follows.

Consider the system

$$\dot{\xi} = \varrho(\xi) \tag{2.7}$$

where  $\varrho : \mathbb{R}^m \mapsto \mathbb{R}^m$  is a locally Lipschitz map.

**Theorem 2.2.1. *Invariance Principle*** (Adapted from [83]). *Let  $\Omega \subset \mathbb{R}^m$  be a compact set that is positively invariant with respect to (2.7). Let  $V : \mathbb{R}^m \mapsto \mathbb{R}$  be a continuously differentiable function such that  $\dot{V}(\xi) \leq 0$  in  $\Omega$ . Let  $\mathcal{S}$  be the set of all points in  $\Omega$  where  $\dot{V}(\xi) = 0$ . Let  $\mathcal{M}$  be the largest invariant set in  $\mathcal{S}$ . Then every solution of (2.7) starting in  $\Omega$  approaches  $\mathcal{M}$  as  $t \rightarrow \infty$ .*

In this document, we use passivity theory and the invariance principle of Theorem 2.2.1 to analyze the stability properties of local control laws.

## 2. PRELIMINARIES

---



# 3

## Distributed Resource Allocation Among Dynamic Networks

### 3.1 Network Dynamics

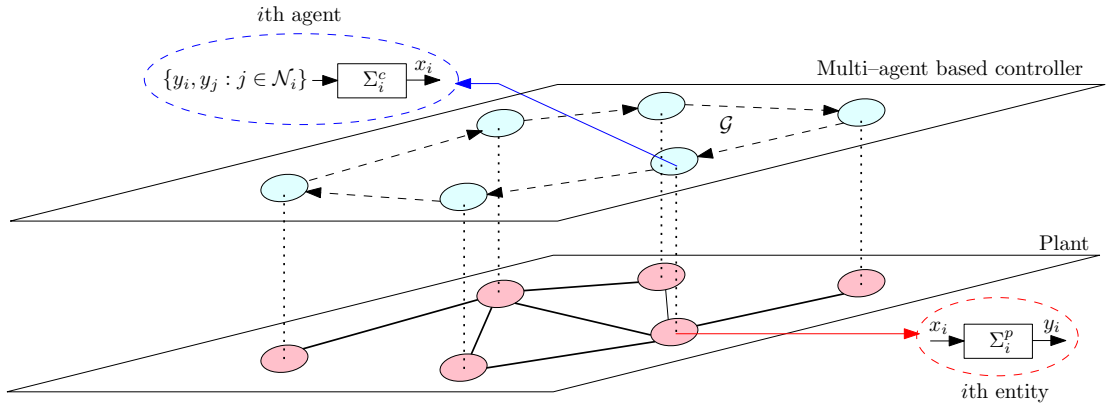
In several engineering problems, we have to deal with large-scale systems, which are composed by dynamical entities that interact with each other. We can find a large number of examples of this kind of systems, especially in the emergent field of smart grids and smart energy systems. In many of those problems, the main goal is associated with certain control criteria that depend on the state of the overall system. This overall-state dependency is challenging since, in some real situations, the geographical distribution of the entities, the limited power computation, or some privacy issues make the implementation of traditional centralized controllers (i.e., controllers that know the information of the whole system) not viable. Under these scenarios, an information constraint emerges. Hence, it is necessary to seek for distributed solutions as the one represented by the multi-agent approach depicted in Figure 3.1.

The bottom layer of Figure 3.1 shows a plant (e.g., a smart building) comprised by  $n$  dynamical entities or sub-systems, which can be modeled by the following set of differential equations,

$$\Sigma_i^p : \begin{cases} \dot{T}_i &= g_i(T, x_i) \\ y_i &= h_i(T, x_i) \end{cases}, \quad \text{for } i = 1, \dots, n, \quad (3.1)$$

where  $\Sigma_i^p$  denotes the  $i$ th sub-system;  $T_i \in \mathbb{R}^{m_i}$  is the sub-system state, which can represent a physical quantity (e.g., the power generated by a micro-grid, or the temper-

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS



**Figure 3.1:** A large-scale system controlled by a network of agents.

ature of a room in a building);  $T = [T_1^\top, \dots, T_n^\top]^\top$ ;  $x_i \in \mathbb{R}$  is the sub-system input; and  $y \in \mathbb{R}^n$  is the sub-system output. Furthermore,  $g_i : \mathbb{R}^{m_i} \times \mathbb{R} \mapsto \mathbb{R}^{p_i}$  is locally Lipschitz, and  $h : \mathbb{R}^{m_i} \times \mathbb{R} \mapsto \mathbb{R}$  is continuous. Notice that the overall system dynamics are not necessarily decoupled since  $g_i$  depends on the global state  $T$ . The set of sub-systems (3.1) can be written in compact form as follows,

$$\Sigma^p : \begin{cases} \dot{T} &= g(T, x) \\ y &= h(T, x) \end{cases}, \quad (3.2)$$

where  $x = [x_1, \dots, x_n]^\top$ ,  $y = [y_1, \dots, y_n]^\top$ ,  $g = [g_1^\top, \dots, g_n^\top]^\top$ , and  $h = [h_1, \dots, h_n]^\top$ .

On the other hand, the upper layer of Figure 3.1 shows the distributed control scheme based on a multi-agent architecture. In this class of scheme, we have a set of agents (i.e., entities that are capable to make decisions [84]. Physically, they can be processors, computers, or embedded systems) that are connected by a communication network characterized by the weighted graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ . The set of nodes of  $\mathcal{G}$  represents the set of agents (therefore, the terms ‘agent’ and ‘node’ are used interchangeably in this document), and the set of edges of  $\mathcal{G}$  represents the available communication channels, i.e., agents  $i$  and  $j$  can share information if and only if  $(i, j) \in \mathcal{E}$ . Although, the positive elements of the weighted adjacency matrix  $\mathcal{A} = [a_{ij}]$  are sometimes used to describe the intensity of the communication links, we generally assume that they have a value of one, i.e.,  $a_{ij} = 1$  if and only if  $(i, j) \in \mathcal{E}$ , and  $a_{ij} = 0$  if and only if  $(i, j) \notin \mathcal{E}$ . Additionally, in this document, we consider bidirectional communications, i.e.,  $a_{ij} = a_{ji}$ . This implies that the graph  $\mathcal{G}$  is undirected.

---

## 3.2 Control Objective and Resource Allocation Dynamics

The goal of the agents is to drive the plant to a desired global state. To do this, each agent is responsible for managing one sub-system using only partial information. Specifically, the  $i$ th agent has information on the output of its sub-system and the outputs of its neighboring sub-systems, i.e., the  $i$ th agent knows the values  $\mathcal{Y}_{\mathcal{N}_i} := (y_i, \{y_j : j \in \mathcal{N}_i\})$ , where  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$ . These values can be obtained by the  $i$ th agent by means of measurements, or by using the information received from its neighbors via the communication network  $\mathcal{G}$ . Available information is employed by the agents to take decisions, which are characterized by the control outputs  $x_1, \dots, x_n$ . Thus, the multi-agent network is mathematically described by the following set of differential equations

$$\Sigma_i^c : \dot{x}_i = u_i(\mathcal{Y}_{\mathcal{N}_i}), \quad \text{for } i = 1, \dots, n, \quad (3.3)$$

where  $u_i : \mathbb{R}^{|\mathcal{N}_i|+1}$  is locally Lipschitz. In compact form, (3.3) is denoted by the following system,

$$\Sigma^c : \dot{x} = u(y), \quad (3.4)$$

where  $u = [u_1, \dots, u_n]^\top$  is a distributed map over the graph  $\mathcal{G}$  (cf., Definition 2.1.1), i.e.,  $u_i$  can be computed by the  $i$ th agent employing only local information.

## 3.2 Control Objective and Resource Allocation Dynamics

### 3.2.1 Control Objective

Several applications require that the outputs of the sub-systems that comprise the whole system reach the same value. These applications include, but are not limited to,...

Problems requiring that the sub-systems outputs reach the same value are called output consensus problems. Formally, output consensus is defined as follows (adapted from [67]),

**Definition 3.2.1.** *Consider the set of sub-systems given in Equation (3.2). We say that output consensus is reached if  $\lim_{t \rightarrow \infty} |y_i(t) - y_j(t)| = 0$ , for all  $i, j = 1, \dots, n$ , where  $y_i(t)$  is the output of the sub-system  $\Sigma_i^p$  at time  $t$ .*

In this document, we study the case when output consensus has to be achieved under a multi-variable constraint on the control inputs  $x_1, \dots, x_n$ . This constraint is associated

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

with the preservation of a certain resource, and it is formulated as follows,

$$\sum_{i=1}^n x_i = X, \quad (3.5)$$

where  $X \in \mathbb{R}$  is the available resource. Notice that, if  $x_i$  is the energy provided by the  $i$ th controller, then (3.5) is related to a general limitation of the total energy that can be applied to the plant. The inclusion of resource constraints is frequent in many control and decision-making problems. For instance, building temperature regulation [39], control of water distribution systems [40], bandwidth allocation [41] and access control [42, 43] in communication networks, dispatch of distributed generators for frequency regulation [44], energy saving in lighting systems [22], control for urban drainage systems [45], and so forth.

Summarizing, the control goal of the multi-agent network (3.4) is twofold:

**O1.** Satisfying the constraint (3.5).

**O2.** Driving (3.2) to output consensus.

#### 3.2.2 Resource Allocation Dynamics

In order to reach the objectives described above, it is necessary to design a set of local control laws,  $u_1, \dots, u_n$ , that can be applied by the multi-agent network (3.4). In this regard, we propose to use the following resource allocation algorithm

$$u_i(\mathcal{Y}_{\mathcal{N}_i}) = \sum_{j \in \mathcal{N}_i} a_{ij}(y_j - y_i), \text{ for all } i = 1, \dots, n. \quad (3.6)$$

We recall that  $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$  is the set of neighbors of node  $i$ , and  $a_{ij} > 0$  is the weight of the link that connects nodes  $i$  and  $j$ .

Before starting the analysis of this algorithm, it is important to notice the fact that the right hand side of Equation (3.6) is a distributed map over the graph  $\mathcal{G}$  since the  $i$ th agent only requires local information to compute the quantity  $\sum_{j \in \mathcal{N}_i} a_{ij}(y_j - y_i)$ . We also notice the fact that the resource allocation algorithm corresponds to a classic consensus protocol, whose effectiveness has been proven in the literature. For instance, there exists a large number of results regarding the use of (3.6) to solve consensus problems when  $y_i$  is the output of a memoryless system that directly depends on  $x_i$  (cf.,

## 3.2 Control Objective and Resource Allocation Dynamics

---

Equation(2.5)) [27, 28, 32, 35, 36]. However, our research deals with the dynamic case (3.2).

We are interested in proving that our control goals **O1** and **O2** can be achieved by using (3.6). First, let us focus on **O1**, i.e., satisfaction of the resource constraint (3.5).

**Lemma 3.2.1.** *Assume that the multi-agent network (3.4) uses the resource allocation algorithm (3.6). If the initial conditions of (3.4) satisfy the resource constraint (3.5), i.e.,  $\sum_{i=1}^n x_i(0) = X$ , then  $x(t)$  satisfies (3.5) for all  $t \geq 0$ .*

*Proof.* It is sufficient to prove that the quantity  $\rho := \sum_{i=1}^n x_i$  is positively invariant. To do that, let us calculate the time derivative of this quantity, which is given by  $\dot{\rho} = \sum_{i=1}^n \dot{x}_i$ . Replacing  $\dot{x}_i$  by the dynamics of the multi-agent network (3.4) when the resource allocation algorithm (3.6) is used, we have that  $\dot{\rho} = \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} a_{ij}(y_j - y_i)$ . In matrix form, this expression is given by  $\dot{\rho} = -\mathbf{1}^\top L(\mathcal{G})y$ , where  $\mathbf{1}$  is the column vector of ones. Notice that, since the graph Laplacian  $L(\mathcal{G})$  is symmetric,  $\mathbf{1}^\top L(\mathcal{G}) = L(\mathcal{G})\mathbf{1} = \mathbf{0}$  (cf., Equation (2.2)). Therefore,  $\dot{\rho} = 0$ . This implies that the quantity  $\rho = \sum_{i=1}^n x_i$  is positively invariant.  $\square$

Therefore, the resource allocation algorithm attains **O1** under a proper initialization. Notice that this result holds regardless of the topology of the communication network  $\mathcal{G}$  and the class of sub-systems dynamics (3.2). In the next section, we provide conditions on the graph  $\mathcal{G}$  and on the dynamics of the sub-systems to guarantee that **O2** is also attained by applying (3.6).

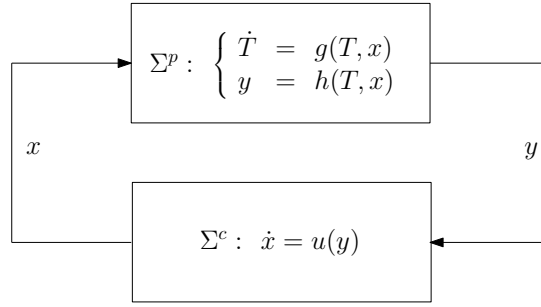
### 3.2.3 Convergence to Output Consensus

The set of sub-systems (3.2) controlled via the multi-agent network (3.4) can be viewed as the feedback interconnection shown in Figure 3.2. If the agents apply the resource allocation algorithm described in Equation (3.6), any equilibrium point  $(T^*, x^*)$  of this feedback interconnection must satisfy the following property.

**Proposition 3.2.1.** *Let  $(T^*, x^*)$  be an equilibrium point of the feedback connection shown in Figure 3.2, and let  $y^* = h(T^*, x^*)$  be the steady state output of  $\Sigma^p$ . If  $u(y)$  is given by the resource allocation algorithm (3.6) and the communication graph  $\mathcal{G}$  is connected, then  $y_i^* = y_j^*$ , for all  $i, j = 1, \dots, n$ , where  $y_i^*$  is the  $i$ th element of the vector  $y^*$ .*

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---



**Figure 3.2:** Feedback interconnection of systems (3.2) and (3.4).

*Proof.* If  $u(y)$  is given by (3.6), then the dynamics of  $\Sigma^c$  can be written in matrix form as  $\dot{x} = -L(\mathcal{G})y$ . Besides, since  $(T^*, x^*)$  is an equilibrium point of the closed loop system, we have that  $-L(\mathcal{G})y^* = \mathbf{0}$ . This implies that  $y^*$  belongs to the null space of the graph Laplacian  $L(\mathcal{G})$ . On the other hand, combining the result stated in Theorem 2.1.1, the property given in Equation (2.2), and the fact that  $\mathcal{G}$  is connected by assumption, we can conclude that the null space of  $L(\mathcal{G})$  is equal to the linear span of  $\mathbf{1}$ , which is given by  $\text{span}(\mathbf{1}) = \{\xi \in \mathbb{R}^n : \xi = \kappa \mathbf{1}\}$ , where  $\kappa \in \mathbb{R}$  is a constant. Therefore,  $y^* = \kappa \mathbf{1}$ . Hence,  $y_i^* = y_j^*$ , for all  $i, j = 1, \dots, n$ .  $\square$

According to Definition 3.2.1, the statement in Proposition 3.2.1 implies that if the equilibrium point  $(T^*, x^*)$  is asymptotically stable, then output consensus is reached. In order to analyze the stability of  $(T^*, x^*)$ , let us express the dynamics of  $\Sigma^p$  and  $\Sigma^c$  in error coordinates.

The set of sub-systems  $\Sigma^p$  given in Equation (3.2) can be written in error coordinates as follows,

$$\begin{aligned} \dot{e}_T &= g^e(e_T, e_x) \\ e_y &= h^e(e_T, e_x), \end{aligned} \tag{3.7}$$

where  $e_T = T - T^*$ ,  $e_x = x - x^*$ , and  $e_y = y - y^*$ . Moreover,  $g^e(e_T, e_x) = g(T, x) - g(T^*, x^*)$  and  $h^e(e_T, e_x) = h(T, x) - h(T^*, x^*)$ , for all  $(T, x) \in \mathbb{R}^m \times \mathbb{R}^n$ . Notice that  $h^e(0, 0) = 0$ , and  $g^e(0, 0) = 0$  since  $(T^*, x^*)$  is an equilibrium point of (3.2). Let us make the following assumption on (3.7).

**Assumption 3.2.1.** Consider the dynamical system (3.7). If  $g^e(0, e_x) = 0$ , then  $e_x = 0$ .

This assumption prevents the original system (3.2) to have other rest points of the form  $(T^*, \tilde{x}^*)$ , where  $\tilde{x}^* \neq x^*$ . It is worth noting that systems satisfying Assumption 3.2.1

### 3.2 Control Objective and Resource Allocation Dynamics

---

are common in several applications, including mobile-sensors networks, multi-machine power systems, and social swarms (see e.g., [68]).

On the other hand, the dynamics of the multi-agent network  $\Sigma^c$  (3.4) that applies the resource allocation algorithm (3.6) is expressed in error coordinates as follows (using the compact matrix form),

$$\dot{e}_x = -L(\mathcal{G})e_y. \quad (3.8)$$

This is true given the fact that  $L(\mathcal{G})(y-y^*) = L(\mathcal{G})y$ . The next original result establishes an important property of this system.

**Proposition 3.2.2.** *Assume that  $x^*$  satisfies the resource constraint (3.5), i.e.,  $\sum_{i=1}^n x_i^* = X$ . If  $x(0)$  satisfies (3.5) and  $\mathcal{G}$  is connected, then the multi-agent systems in error coordinates given in Equation (3.8) is passive and lossless from the input  $e_y$  to the output  $-e_x$ .*

*Proof.* Since  $\mathcal{G}$  is connected, we can use the following storage function,

$$V_x(e_x) = \frac{1}{2} e_x^{k\top} \left( L^k(\mathcal{G}) \right)^{-1} e_x^k, \quad (3.9)$$

where  $e_x^k$  is the vector obtained by removing the  $k$ th element from vector  $e_x$ , and  $L^k(\mathcal{G})$ , is defined in Lemma 2.1.1. Although the inverse matrix of  $L^k(\mathcal{G})$  is positive definite according to Lemma 2.1.1, the storage function  $V_x(e_x)$  is positive semidefinite since it does not depend on the  $k$ th element of  $e_x$ . The derivative of  $V_x(e_x)$  along the trajectories of (3.8) is given by

$$\dot{V}_x(e_x) = e_x^{k\top} \left( L^k(\mathcal{G}) \right)^{-1} \dot{e}_x^k. \quad (3.10)$$

Let us express  $\dot{e}_x^k$  in terms of the matrix  $L^k(\mathcal{G})$  as follows,

$$\dot{e}_x^k = -L^k(\mathcal{G})e_y^k - \mathbf{I}_k^{k_r} e_{y_k}, \quad (3.11)$$

where  $e_{y_k}$  is the  $k$ th element of the vector  $e_y$ ,  $e_y^k$  is the vector obtained by removing the  $k$ th element from vector  $e_y$ , and  $\mathbf{I}_k^{k_r}$  is defined in Lemma 2.1.1. Replacing (3.11) in (3.10), we have that

$$\dot{V}_x(e_x) = -e_x^{k\top} e_y^k - e_x^{k\top} (L(\mathcal{G}))^{-1} \mathbf{I}_k^{k_r} e_{y_k}.$$

Taking into account that  $(L^k(\mathcal{G}))^{-1} \mathbf{I}_k^{k_r} = -\mathbf{1}$  (cf., Lemma 2.1.1), we obtain

$$\dot{V}_x(e_x) = -e_x^{k\top} e_y^k + \mathbf{1}^\top e_x^k e_{y_k}. \quad (3.12)$$

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

Notice that  $\mathbf{1}^\top e_x^k = \sum_{i \in \nu, i \neq k} e_{x_i}$ , where  $e_{x_i}$  is the  $i$ th element of the vector  $e_x$ . On the other hand, according to Lemma 3.2.1,

$$e_{x_k}(t) = - \sum_{i \in \nu, i \neq k} e_{x_i}(t), \text{ for all } t \geq 0. \quad (3.13)$$

This property holds since both  $x^*$  and  $x(0)$  satisfy (3.5) by assumption. Using (3.13), Equation (3.12) can be expressed as follows

$$\begin{aligned} \dot{V}_x(e_x) &= -e_x^{k\top} e_y^k - e_{x_k} e_{y_k} \\ &= -e_x^\top e_y. \end{aligned}$$

Therefore, (3.8) is lossless. □

Passivity of (3.8) is important since the feedback interconnection of two passive systems generally leads to the emergence of stable rest points. Thus, we take advantage of this property to establish requirements on (3.7) that ensure output consensus under the configuration shown in Figure 3.2. These requirements are summarized in the following contribution.

**Theorem 3.2.1.** *Let  $(T^*, x^*)$  be an equilibrium point of the feedback interconnection of systems (3.2) and (3.4), where  $u(y)$  is given by the resource allocation algorithm (3.6). Assume that the following conditions hold:*

- A1.** *The communication graph  $\mathcal{G}$  of the multi-agent system (3.4) is connected.*
- A2.**  *$x^*$  and  $x(0)$  satisfy the resource constraint (3.5).*
- A3.** *The system (3.2) expressed in error coordinates with respect to  $(T^*, x^*)$  satisfies Assumption 3.2.1. Moreover it is strictly passive from the input  $e_x$  to the output  $e_y$  with radially unbounded storage function.*

*Then (3.2) reaches output consensus.*

*Proof.* Using the result in Proposition 3.2.1, it is sufficient to show that  $(T^*, x^*)$  is asymptotically stable, or equivalently that the origin of (3.7)–(3.8) is asymptotically stable. To do this, we use the invariance principle given in Theorem 2.2.1 as follows.

Let us take  $V(e_x, e_T) = V_x(e_x) + V_T(e_T)$ , where  $V_x(e_x)$  is the storage function of (3.8) that is given in Equation (3.9), and  $V_T(e_T)$  is the storage function of (3.7). Notice that, since conditions of Proposition 3.2.2 hold and (3.7) is strictly passive, we have that

$$\begin{aligned} \dot{V}(e_x, e_T) &= \dot{V}_x(e_x) + \dot{V}_T(e_T) \\ &\leq -e_x^\top e_y + e_x^\top e_y - \psi(e_T) \\ &= -\psi(e_T), \end{aligned}$$



## 3.2 Control Objective and Resource Allocation Dynamics

---

where  $\psi(e_T)$  is a positive definite function. Furthermore, let us consider the compact set  $\Omega_c = \{(e_x, e_T) : e_{x_k} = -\sum_{i \in \mathcal{V}, i \neq k} e_{x_i}, V_x(e_x) + V_T(e_T) \leq c\}$ , where  $c$  is any positive real number. Notice that  $\Omega_c$  is bounded since  $V_T(e_T)$  is positive definite (this is true since (3.7) is strictly passive) and radially unbounded, and  $V_x(e_x)$  is positive definite with respect to  $\{e_{x_i} : i \in \mathcal{V}, i \neq k\}$  and radially unbounded. The set  $\Omega_c$  is also positively invariant with respect to (3.7)–(3.8) since  $\dot{V}_x(e_x) + \dot{V}_T(e_T) \leq 0$  and  $\sum_{i \in \mathcal{V}} e_{x_i}(t) = 0$ , for all  $t \geq 0$  (cf., Lemma 3.2.1). Additionally, we define the set  $\mathcal{S}$  as the set where  $\dot{V}(e_x, e_T) = 0$ , i.e.,  $\mathcal{S} = \{(e_x, e_T) : e_T = 0\}$ . Let  $\mathcal{M}$  be an invariant set of  $\mathcal{S}$ . Notice that if  $(e_T, e_x) \in \mathcal{M}$ , then  $\dot{e}_T = 0$ . This implies that  $g^e(0, e_x) = 0$ . Hence,  $e_x = 0$  (given the fact that Assumption 3.2.1 holds). Thus, the only invariant set of  $\mathcal{S}$  is  $\mathcal{M} = \{(0, 0)\}$ . Since Theorem 2.2.1 states that every solution of (3.7)–(3.8) starting in  $\Omega_c$  approaches  $\mathcal{M}$  as  $t \rightarrow \infty$ , we can conclude that the origin of (3.7)–(3.8) is asymptotically stable.  $\square$

The result in Theorem 3.2.1 establishes sufficient conditions on the communication graph **(A1)**, initialization of the local control laws **(A2)**, and characteristics of the controlled plant **(A3)** that guarantee that the control objective **O2** is reached by using the resource allocation algorithm (3.6). We notice the fact that, if the controlled plant does not satisfy the requirement given in **A3**, then it is possible to implement a first-level controller that makes the plant passive.

Summarizing, we have designed a set of local control laws (3.6) that allows passive systems to attain the control goals **O1** and **O2**.

### 3.2.4 Extension to Memoryless Systems

We can extend the result in Theorem 3.2.1 to deal with passive memoryless systems. To this end, we consider that, in the resource allocation algorithm (3.6),  $y = [y_1, \dots, y_n]^\top$  is the output of the following memoryless system

$$y = \vartheta(x). \tag{3.14}$$

where  $\vartheta : \mathbb{R}^n \mapsto \mathbb{R}^n$  is a continuous function. In order to simplify the analysis, let us write the above system in error coordinates as follows,

$$e_y = \vartheta^e(e_x). \tag{3.15}$$

where  $e_x = x - x^*$ ,  $\vartheta^e(e_x) = \vartheta(x) - \vartheta(x^*)$ , and  $x^*$  is an equilibrium point of the closed loop system given by Equations (3.4) and (3.14). Similarly to the dynamic case, we make the following assumption on (3.15).

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

**Assumption 3.2.2.** Consider the memoryless system (3.15). If  $e_x^\top \vartheta^e(e_x) = 0$ , then  $e_x = 0$ .

This requirement is satisfied if, for instance, the vector  $\vartheta$  is comprised of monotonic functions.

The following theorem states that the results obtained for the dynamic case also hold for the memoryless case.

**Theorem 3.2.2.** Let  $x^*$  be an equilibrium point of the system given by (3.4) and (3.14). Assume that the following conditions hold:

- A1.** The communication graph  $\mathcal{G}$  of the multi-agent system (3.4) is connected.
- A2.**  $x^*$  and  $x(0)$  satisfy the resource constraint (3.5).
- A3.** The memoryless system (3.14) expressed in error coordinates with respect to  $x^*$  satisfies Assumption 3.2.2. Moreover it is passive from the input  $e_x$  to the output  $e_y$ .

Then **O1** is attained, and the output  $y$  of the memoryless system (3.14) reaches consensus.

*Proof.* We proceed as follows,

- **O1:** The proof of the resource constraint satisfaction (i.e.,  $\sum_{i=1}^n x_i(t) = X$ , for all  $t \geq 0$ ) follows from Lemma 3.2.1.
- *Output consensus:* To prove that the system output  $y$  reaches consensus, it is sufficient to show that  $x^*$  is asymptotically stable under the system given by (3.4) and (3.14) (this equivalence follows from the fact that any equilibrium point of ((3.4),(3.14)) must satisfy output consensus (cf., Proposition 3.2.1)), or equivalently that the origin of ((3.8),(3.15)) is asymptotically stable.

Let us take  $V_x(e_x)$  equal to the storage function given in Equation (3.9). Since (3.8) is lossless from  $e_y$  to  $-e_x$  (cf., Proposition 3.2.2), we have that  $\dot{V}_x(e_x) = -e_x^\top e_y$ . Replacing  $e_y$  by the output of the memoryless system (3.15), we obtain  $\dot{V}_x(e_x) = -e_x^\top \vartheta^e(e_x)$ . Since (3.15) is passive and satisfies Assumption 3.2.2, we conclude that  $\dot{V}_x(e_x)$  is negative definite. Finally, we define the positively invariant compact set  $\Omega_c = \{e_x : e_{x_k} = -\sum_{i \in \nu, i \neq k} e_{x_i}, V_x(e_x) \leq c\}$ , where  $c$  is any positive real number, and employ the invariance principle given in Theorem 2.2.1 to prove that any solution of ((3.8),(3.15)) starting in  $\Omega_c$  asymptotically converges to the origin.

## 3.2 Control Objective and Resource Allocation Dynamics

---

□

The statement in Theorem 3.2.2 is not surprising since the steady state behavior of passive dynamical systems is similar to the behavior of passive memoryless functions. In fact, we can take advantage of this similarity to derive a general case that merges the results obtained in Theorems 3.2.1 and 3.2.2. This general case is formulated as follows.

Consider the function  $f : \mathbb{R}^m \times \mathbb{R}^n \mapsto \mathbb{R}^n$  defined by

$$f(T, x) = h(T, x) + \vartheta(x), \quad (3.16)$$

where  $h(T, x)$  is the output of the dynamical system (3.2), and  $\vartheta(x)$  is the output of the memoryless system (3.14). Assume that our problem is to attain the objective **O1** and reach consensus on  $f(T, x)$ . To address this problem, we propose to use the multi-agent based controller (3.4) with the following resource allocation algorithm.

$$u_i = \sum_{j \in \mathcal{N}_i} (f_j - f_i), \text{ for all } i = 1, \dots, n. \quad (3.17)$$

To analyze the performance of the proposed controller, we use a similar procedure to the one developed in Section 3.2.3. Thus, we begin by expressing the closed loop system ((3.2),(3.4)), with  $u$  defined in (3.17), in error coordinates with respect to the equilibrium point  $(T^*, x^*)$  as follows.

$$\begin{aligned} \dot{e}_T &= g^e(e_T, e_x) \\ \dot{e}_x &= -L(\mathcal{G}) \left( h^e(e_T, e_x) + \vartheta^e(e_x) \right). \end{aligned} \quad (3.18)$$

Let us make the following assumption on this system.

**Assumption 3.2.3.** *Consider the error dynamics (3.18). If  $g^e(0, e_x) = 0$  and  $\vartheta^e(e_x) = 0$ , then  $e_x = 0$ .*

Different from Assumption 3.2.1, Assumption 3.2.3 allows the system (3.2) to have other equilibrium points of the form  $(T^*, \tilde{x}^*)$ , where  $\tilde{x}^* \neq x^*$ , as long as these points do not satisfy the equation  $(\tilde{x}^* - x^*)^\top (\vartheta(\tilde{x}^*) - \vartheta(x^*)) = 0$ .

If (3.18) satisfies some conditions, then we can guarantee an appropriate performance of the proposed control scheme. This set of conditions is specified in the following result.

**Theorem 3.2.3.** *Let  $(T^*, x^*)$  be an equilibrium point of the feedback interconnection of systems (3.2) and (3.4), where  $u$  is given by the resource allocation algorithm (3.17). Assume that the following conditions hold:*

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

- A1.** *The communication graph  $\mathcal{G}$  of the multi-agent system (3.4) is connected.*
- A2.**  *$x^*$  and  $x(0)$  satisfy the resource constraint (3.5).*
- A3.** *The system (3.2) expressed in error coordinates with respect to  $(T^*, x^*)$  is strictly passive from the input  $e_x$  to the output  $h^e(e_T, e_x)$  with radially unbounded storage function.*
- A4.** *The memoryless system (3.14) expressed in error coordinates with respect to  $x^*$  is passive from the input  $e_x$  to the output  $\vartheta^e(e_x)$ .*
- A5.** *Assumption 3.2.3 holds.*

Then **O1** is attained, and the variable  $f(T, x)$  given in Equation (3.16) reaches consensus.

*Proof.* We proceed as follows:

- **O1:** The proof of the resource constraint satisfaction (i.e.,  $\sum_{i=1}^n x_i(t) = X$ , for all  $t \geq 0$ ) is straightforward by using the same steps described in the proof of Lemma 3.2.1.
- *Consensus on  $f(T, x)$ :* In order to prove that the variable  $f(T, x)$  reaches consensus, it is sufficient to show that  $(T^*, x^*)$  is asymptotically stable under the system ((3.2),(3.4)), with  $u$  defined in (3.17) (this equivalence follows from the fact that any equilibrium point of this system must satisfy consensus on  $f(T, x)$ ), or equivalently that the origin of (3.18) is asymptotically stable. To do this, we take the function  $V(e_x, e_T) = V_x(e_x) + V_T(e_T)$ , where  $V_x(e_x)$  is the storage function of (3.8) that is given in Equation (3.9), and  $V_T(e_T)$  is the storage function of (3.7). Then, we use the same steps as in the proof of Theorem 3.2.1 to obtain that the derivative of  $V(e_x, e_T)$  along the trajectories of (3.18) is given by

$$\dot{V}(e_x, e_T) = -\psi(e_T) - e_x^\top \vartheta^e(e_x),$$

where  $\psi(e_T) > 0$ , and  $e_x^\top \vartheta^e(e_x) \geq 0$  since **A4** holds.

Furthermore, let us consider the compact set

$$\Omega_c = \{(e_x, e_T) : e_{x_k} = - \sum_{i \in \nu, i \neq k} e_{x_i}, V_x(e_x) + V_T(e_T) \leq c\},$$

where  $c$  is any positive real number. Notice that  $\Omega_c$  is bounded since  $V_T(e_T)$  is positive definite (this is true since (3.7) is strictly passive) and radially unbounded,

### 3.2 Control Objective and Resource Allocation Dynamics

---

and  $V_x(e_x)$  is positive definite with respect to  $\{e_{x_i} : i \in \mathcal{V}, i \neq k\}$  and radially unbounded. The set  $\Omega_c$  is also positively invariant with respect to (3.18) since  $\dot{V}_x(e_x) + \dot{V}_T(e_T) \leq 0$  and  $\sum_{i \in \mathcal{V}} e_{x_i}(t) = 0$ , for all  $t \geq 0$ . Additionally, we define the set  $\mathcal{S}$  as the set where  $\dot{V}(e_x, e_T) = 0$ , i.e.,  $\mathcal{S} = \{(e_x, e_T) : e_T = 0, e_x^\top \vartheta^e(e_x) = 0\}$ . Let  $\mathcal{M}$  be an invariant set of  $\mathcal{S}$ . Notice that if  $(e_T, e_x) \in \mathcal{M}$ , then  $\dot{e}_T = 0$ . This implies that  $g^e(0, e_x) = 0$ . Therefore, if  $(e_T, e_x) \in \mathcal{M}$ , then  $g^e(0, e_x) = 0$  and  $e_x^\top \vartheta^e(e_x) = 0$ . Hence,  $e_x = 0$  (given the fact that **A5** holds). Thus, the only invariant set of  $\mathcal{S}$  is  $\mathcal{M} = \{(0, 0)\}$ . Since Theorem 2.2.1 states that every solution of (3.18) starting in  $\Omega_c$  approaches  $\mathcal{M}$  as  $t \rightarrow \infty$ , we can conclude that the origin of (3.18) is asymptotically stable. □

In conclusion, we have shown that the resource allocation algorithm (3.17) allows the multi-agent based controller (3.4) to satisfy the resource constraint (3.5) and drive a specified quantity to consensus, as long as this quantity depends on the output of a passive system and/or the output of a passive memoryless function. The inclusion of memoryless terms is important; for instance, to capture actuator constraints as we will show in Section 3.3.4.

#### 3.2.5 Discussion

In the literature, significant research attention has been paid to the control of dynamical systems through consensus-based protocols (e.g., see [47, 67, 68, 69]). Most of this research effort is directed towards achieving **O2** whereas **O1** is somewhat disregarded. However, in a large number of applications **O1** plays a fundamental role since it can be directly linked to the satisfaction of technical constraints on the actuators. For instance, a limitation on the available energy used to control the plant. Some works address **O1** by assuming that the dynamics of the controlled plant are linear. In this case, conditions that guarantee a proper performance of the closed-loop system can be derived by employing linear matrix inequalities [85, 86].

For the nonlinear case, passivity based theory is used in [67, 68, 69]. Nonetheless, these papers do not deal with the same problem studied in this dissertation. Once again, the key difference is that they do not take into account the control goal **O1**, i.e., the resource constraint (3.5). We highlight the work in [67], where the authors propose a

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

control scheme similar to the one discussed in our research. They analyze the closed-loop response of systems of the following form

$$\text{Plant: } \begin{cases} \dot{T} &= g(T, x) \\ y &= h(T) \end{cases}, \quad \text{Controller: } \begin{cases} \dot{x} &= \sum_{j \in \mathcal{N}_i} (y_j - y_i). \end{cases} \quad (3.19)$$

We notice that the main difference with the distributed resource allocation algorithm ((3.3),(3.6)) is that in (3.19), the control law does not have internal dynamics, but it is the output of a memoryless system. The authors in [67] require similar conditions to those provided in Theorem 3.2.1 to guarantee that output consensus is reached.

Comparing the proposed control scheme ((3.2),(3.4)) with the one shown in (3.19), it can be noticed that the latter can be adapted to take the form ((3.2),(3.4)). For instance, let us consider the following transformation

$$\text{Plant: } \begin{cases} \dot{T} &= g(T, \tilde{x}) \\ \dot{\tilde{x}} &= x \\ y &= h(T) \end{cases}, \quad (3.20)$$

where we have augmented the state of the original plant by adding the vector  $\tilde{x}$ . Hence, one might think that our result in Theorem 3.2.1 is covered by the result in [67]. This is not true since the transformation shown in (3.20) does not preserve passivity, i.e., we cannot guarantee that if the plant in (3.19) is passive, then the augmented plant in (3.20) is also passive. To clarify this statement, we propose the following example.

**Example 3.2.1.** *Let us consider a linear system and its corresponding transformation given by*

$$\text{Plant: } \begin{cases} \dot{T} &= -T + x \\ y &= T \end{cases}, \quad (3.21)$$

$$\text{Augmented Plant: } \begin{cases} \dot{T} &= -T + \tilde{x} \\ \dot{\tilde{x}} &= x \\ y &= T \end{cases},$$

where  $T, x, \tilde{x} \in \mathbb{R}$ . The original plant is strictly passive from the input  $x$  to the output  $y$  since the time derivative of the storage function  $V(T) = \frac{1}{2}T^2$  is equal to  $\dot{V}(T) = -T^2 + xy$ . To verify passivity of the augmented plant, let us employ the following lemma and the fact that a minimal time invariant realization (as the one that describes the augmented plant in (3.21)) is passive if and only if it is positive real (see [87])

### 3.3 Distributed Building Temperature Control Under Power Constraints

---

**Lemma 3.2.2. Positive Real** (adapted from [83]). Let  $G(s)C(sI-A)^{-1}B$  be a transfer function matrix, where  $(A, B, C)$  is a minimal realization. Then  $G(s)$  is positive real if and only if there exist matrices  $P = P^\top > 0$ , and  $L$  such that

$$PA + A^\top P = -L^\top L \quad (3.22a)$$

$$PB = C^\top \quad (3.22b)$$

If we apply this lemma to the augmented plant in (3.21), we have that any matrix  $P$  must have the following form (using (3.22b))

$$P = \begin{bmatrix} p & 1 \\ 1 & 0 \end{bmatrix},$$

where  $p \in \mathbb{R}$ . However,  $\det(P) = -1$ , i.e.,  $P$  is not positive definite, which contradicts the requirements in Lemma 3.2.2. Thus, the augmented plant is not positive real. In this way, we have shown that, although the original plant is passive, the augmented plant does not preserve this characteristic. In conclusion, this example shows that the result given in Theorem 3.2.1, cannot be obtained directly from the contribution developed in [67].

The first part of this chapter has presented the general framework studied in our dissertation, and the design of local control laws (3.6) that allow passive systems (dynamical or memoryless) to attain the objectives **O1** and **O2**. To illustrate the relevance of our results, in the next sections we present two application examples. In the first example, we address the distributed temperature control in large-scale buildings whose heating systems are not over-designed (i.e., energy efficient buildings). The second example deals with a widely studied problem involving distributed decision making, the optimal resource allocation in a network of agents. Specifically, we present an application related to the distributed management of the customers of an aggregating entity in a smart grid environment.

## 3.3 Distributed Building Temperature Control Under Power Constraints

### 3.3.1 Motivation

Due to a widespread interest on environmental and economic issues, energy efficiency is becoming a research trending topic in the last time; an example of this fact is the

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

increasing number of scientific publications on that subject (e.g., see [88] and the references therein). In this regard, several studies have shown that most of the total energy consumption is done by commercial and residential buildings, e.g., according to [89], buildings in Europe account for around 40% of the total energy, and over 70% for the electricity usage. Additionally, buildings are the main responsible for greenhouse gas emissions (almost 40% of the total [90]), causing harmful effects on the environment. Therefore, it is not surprising to find in the literature new strategies, guidelines and standards that tend to maximize the buildings energy efficiency. Most of them focus on the optimization of heating, ventilating, and air conditioning (HVAC) systems, given the fact that these systems consume the largest amount of energy in a building. For instance, in [91] a model predictive control (MPC) that uses the rooms' thermal storage capacity is presented, where the main idea is to pre-heat (cool) the rooms depending on the occupation pattern to maintain the temperature in a desirable range, while the cost of the energy is minimized. Similarly, the authors in [60] use MPC to manage the energy in residential buildings. Several constraints and models of the buildings' equipment and energy market are considered. In addition, a hierarchical structure is proposed in order to reduce the underlying computational complexity. Other applications can be found in [92] and [93].

The applications mentioned above consider that the available power in the corresponding HVAC system is always enough to reach the control goals. This is a reasonable assumption, since HVAC systems are, in general, designed to handle worst case loads. However, this over-design of HVAC equipment is one of the main reasons for energy inefficiency [90]. Recent standards (e.g., [94]) are tending to solve the problem by means of the implementation of new guidelines, such as the net-zero energy concept (where buildings do not demand energy from the electrical grid, but they are capable to supply their load by means of distributed generators). Additionally, demand response policies are becoming more common in building management systems [89]. One of those strategies, which leads to the reduction of energy costs, is the limitation of the power consumption during hours of peak electricity demand. Hence, it is possible that at certain hours of the day, some building's services or comfort levels are affected.

For these reasons, it is important to consider scenarios in which the available power of the HVAC system is constrained. In this regard, we can cite the works in [95] and [61], where the authors address the problem using a centralized dynamic resource allocation



### **3.3 Distributed Building Temperature Control Under Power Constraints**

approach based on population dynamics and market mechanisms, respectively. Centralized strategies have shown an appropriate performance to control buildings' HVAC systems under power constraints. However, since a building is a large scale system comprised by a large number of rooms (e.g., typical commercial buildings have more than one hundred rooms), the implementation of a centralized controller is not efficient, or even it could be unfeasible. One possibility is to use distributed methods with multiple controllers that make decisions in a coordinated way, where the coordination is made by means of a communication network, instead of a centralized agent. Thus, we propose to use a distributed methodology, which is based on the multi-agent scheme described in (3.1), to regulate the temperature inside the rooms of a building that has a central heating system (which is a common configuration in several residential buildings). We prove that the proposed control method optimally allocates the available heating power among the rooms, even when that power is strongly limited and the setpoints of the rooms are unreachable. In order to do that, we use the passivity-based analysis given in Theorem 3.2.1. For analysis purposes, before describing the controller design, let us introduce the building thermal model.

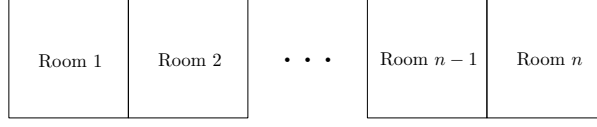
#### **3.3.2 Thermal Model of a Building**

In general terms, the mathematical description of the thermal behavior of building systems is complex since the temperature in the zones of a building is not homogeneous. However, the problem is usually simplified by dividing the system into sections in which a uniform temperature is assumed, and heat transfer occurs only from section to section.

In [96, 97], the authors outline a general model that considers the two most important components that constitute a building: rooms and walls. We use the ideas in [96, 97] in order to model the thermal performance of a building consisting of  $n$  rooms, where each of them is enclosed by a certain number of walls that can be arranged according to different topologies (i.e., the spatial location of rooms is, in general, arbitrary). Figure 3.3 shows an example of the configuration of a building in which the rooms are arranged in a row. This type of configuration is common in several types of structures, as in livestock buildings.

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

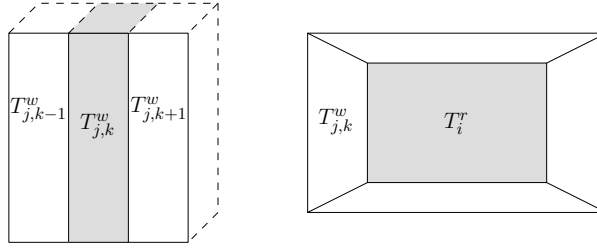


**Figure 3.3:**  $n$  rooms arranged in a row.

#### 3.3.2.1 Thermal Model for a Wall

A wall can be divided into layers with uniform temperature (as shown in Figure 3.4).

The temperature  $T_{j,k}^w$  of the layer  $k$  within the wall  $j$  is given by



**Figure 3.4:** Schematic representation of a room and a wall. The leftmost figure illustrates the cross section of a wall divided into layers, while the rightmost one shows the top view of a room surrounded by walls.

$$(\rho_{j,k}^w c_{j,k}^w V_{j,k}^w) \dot{T}_{j,k}^w = K_{k,k+1}^j (T_{j,k+1}^w - T_{j,k}^w) + K_{k,k-1}^j (T_{j,k-1}^w - T_{j,k}^w) \quad (3.23)$$

where  $\rho_{j,k}^w, c_{j,k}^w, V_{j,k}^w$  are, respectively, the density, specific heat, and volume of the layer  $k$  (the term  $\rho_{j,k}^w c_{j,k}^w V_{j,k}^w$  is associated with the thermal capacitance of the layer). The thermal conductance  $K_{k,k\pm 1}^j$  associated with the link between layers  $k$  and  $k \pm 1$ , can be estimated by harmonic mean as:

$$K_{k,k\pm 1}^j = \frac{A_j}{(L_{j,k\pm 1}/2)/\lambda_{j,k\pm 1} + (L_{j,k}/2)/\lambda_{j,k}}$$

where  $A_j$  is the area of the  $j$ th wall,  $L_{j,k}$  denotes the thickness of the layer  $k$ , and  $\lambda_{j,k}$  its thermal conductivity. Subscripts  $k + 1$  and  $k - 1$  are related to the adjacent layers.

For the layers which are in direct contact with a room or the external environment, there are some conditions that can be described as follows:

- For a layer  $k$  within the wall  $j$  that is in contact with the room  $i$  and does not have an adjacent layer  $k \pm 1$ , the term  $K_{k,k\pm 1}^j (T_{j,k\pm 1}^w - T_{j,k}^w)$  in Equation (3.23) must

### 3.3 Distributed Building Temperature Control Under Power Constraints

be replaced with  $h_i^r A_j (T_i^r - T_{j,k}^w)$ .  $T_i^r$  represents the temperature of the  $i$ th room, and  $h_i^r$  its coefficient of convective heat transfer.

- For a layer  $k$  within the wall  $j$  that is in contact with the external environment and does not have an adjacent layer  $k \pm 1$ , the term  $K_{k,k\pm 1}^j (T_{j,k\pm 1}^w - T_{j,k}^w)$  must be replaced with  $h_a A_j (T_{j,k}^a - T_{j,k}^w)$ .  $T_{j,k}^a$  represents the ambient temperature perceived by the layer  $k$  of the wall  $j$ , and  $h_a$  the external coefficient of convective heat transfer.

#### 3.3.2.2 Thermal Model for a Room

A schematic representation of a room surrounded by walls is shown in Figure 3.4. In our model, we assume a uniform temperature in each room. Moreover, in order to formulate an appropriate thermal control strategy, we consider that each of the rooms has a sensor and an actuator (heater). With these assumptions, the temperature  $T_i^r$  of the  $i$ th room can be modeled as

$$(\rho_a c_a V_i^r) \dot{T}_i^r = \sum_{j \in \Omega_i^r} h_i^r A_j (T_{j,k}^w - T_i^r) + x_i + d_i, \quad (3.24)$$

where  $\Omega_i^r$  is the set of walls adjacent to room  $i$ ,  $T_{j,k}^w$  is the temperature of the layer within the wall  $j$  that is in direct contact with the room  $i$ ;  $A_j$  is the area of wall  $j$ ;  $\rho_a, c_a$  are, respectively, the density and the specific heat of the air;  $V_i^r$  is the volume of the  $i$ th room, and  $h_i^r$  its coefficient of convective heat transfer;  $x_i$  is the heating power supplied by the actuator; and  $d_i$  is a thermal disturbance (e.g., due to opening a window or because of the presence of people who are generating heat). Similarly to Equation (3.23), the terms  $\rho_a c_a V_i^r$  and  $h_i^r A_j$  in Equation (3.24) are associated with the thermal capacitance of the room  $i$  and with the thermal conductance of the junction between the room  $i$  and the wall  $j$ , respectively.

#### 3.3.2.3 Unified Model

If we define the state vector

$$\mathbf{T} = [T_1^r, \dots, T_N^r, T_{1,1}^w, \dots, T_{1,m_1}^w, \dots, T_{M,1}^w, \dots, T_{M,m_M}^w]^\top \quad (3.25)$$

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

where  $N$  is the number of rooms of the building,  $M$  is the number of walls, and  $m_j$  is the number of layers of the  $j$ th wall, Equations (3.23) and (3.24) can be unified into a single expression as follows [98].

$$\theta_i \dot{T}_i = \sum_{j=1}^W \alpha_{i,j} (T_j - T_i) + \alpha_{i,a} (T_i^a - T_i) + v_i (x_i + d_i), \text{ for } i = 1, \dots, W, \quad (3.26)$$

where  $T_i$  is the  $i$ th element of the vector  $\mathbf{T}$  ( $T_i$  corresponds to the temperature of a component of the building, that can be either a room or a wall's layer);  $\theta_i > 0$  is its thermal capacitance,  $\alpha_{i,j} \geq 0$  is the thermal conductance of the junction between elements  $i$  and  $j$  (we notice that  $\alpha_{i,j} = 0$  if the junction between elements  $i$  and  $j$  does not exist. Moreover,  $\alpha_{i,j} = \alpha_{j,i}$ );  $\alpha_{i,a} \geq 0$  is the thermal conductance of the junction between the  $i$ th element and the outside environment (again,  $\alpha_{i,a} = 0$  if this junction does not exist); the variable  $v_i$  takes the values of 1 if  $i = 1, \dots, n$ , and 0 otherwise; and  $T_i^a$  is the ambient temperature perceived by the  $i$ th element when it is in direct contact with the outside environment, if the  $i$ th element is not in direct contact with the outside environment, then  $T_i^a = 0$ . Finally,  $W$  denotes the dimension of the vector  $\mathbf{T}$ , i.e.,  $W$  is the total number of elements (rooms/layers) that comprise the building.

Since in the next sections we plan to use an analysis based on the result given in Theorem 3.2.1, it is convenient to write the dynamics of the building thermal system given in Equation (3.26) in error coordinates, i.e., expressing (3.26) in the form (3.7). To do this, let  $T^* = [T_1^*, \dots, T_W^*]^\top$ ,  $x^* = [x_1^*, \dots, x_N^*]^\top$  be an equilibrium point of the referred system (considering fixed values for  $T_1^a, \dots, T_W^a, d_1, \dots, d_n$ ). Moreover, in order to simplify the notation, let us define the weighted graph  $\mathcal{G}_b = \{\mathcal{V}_b, \mathcal{E}_b, \mathcal{A}_b\}$  that represents the connections between the elements that comprise the building, where the set of nodes  $\mathcal{V}_b = \{1, \dots, W\}$  is related to the components of the building, following the order established in Equation (3.25) (e.g., node 1 corresponds to the first room, node  $n+1$  corresponds to the first layer of the wall 1, etc.). The set of edges  $\mathcal{E}_b \subset \mathcal{V}_b \times \mathcal{V}_b$  is related to the thermal interactions between the building's components, i.e.,  $(i, j) \in \mathcal{E}_b$  if and only if  $\alpha_{i,j} > 0$ . The weighted adjacency matrix  $\mathcal{A}_b$  is formed by the thermal conductances of the junctions between the components of the building, i.e., the component that corresponds to the  $i$ th row and  $j$ th column of  $\mathcal{A}_b$  is given by  $\alpha_{i,j}$ . Using  $\mathcal{G}_b$ , we can express the

### 3.3 Distributed Building Temperature Control Under Power Constraints

building thermal model in error coordinates as follows,

$$\begin{aligned} \dot{e}_T = & - \begin{bmatrix} \frac{1}{\theta_1} & & \\ & \ddots & \\ & & \frac{1}{\theta_W} \end{bmatrix} \left( L(\mathcal{G}_b) + \begin{bmatrix} \alpha_{1,a} & & \\ & \ddots & \\ & & \alpha_{W,a} \end{bmatrix} \right) e_T \\ & + \begin{bmatrix} \frac{1}{\theta_1} & & \\ & \ddots & \\ & & \frac{1}{\theta_W} \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \\ \mathbf{0}_{(W-n) \times n} & \end{bmatrix} e_x \end{aligned} \quad (3.27)$$

where  $e_T = [T_1 - T_1^*, \dots, T_W - T_W^*]^\top$ ,  $e_x = [x_1 - x_1^*, \dots, x_n - x_n^*]^\top$ ,  $L(\mathcal{G}_b)$  is the graph Laplacian of  $\mathcal{G}_b$ ,  $\mathbf{I}_n$  is the  $n \times n$  identity matrix, and  $\mathbf{0}_{(W-n) \times n}$  is a  $(W - n) \times n$  matrix with all its entries being zero. We notice that  $L(\mathcal{G})$  is symmetric given the fact that  $\alpha_{i,j} = \alpha_{j,i}$ . Before stating an important property of this system, let us introduce the following assumption.

**Assumption 3.3.1.** *For all  $i \in \mathcal{V}_b$ , there exists some  $j \in \mathcal{V}_b$  with  $\alpha_{j,a} > 0$  (i.e., the element that corresponds to node  $j$  is in direct contact with the external environment), such that there is a path between  $i$  and  $j$ .*

This implies that no component of the building is completely isolated from the outside environment. Without this assumption, one building's component could store energy indefinitely, which is not realistic. Taking into account this assumption, the system given in Equation (3.27) has the following property.

**Proposition 3.3.1.** *Let  $T^* = [T_1^*, \dots, T_W^*]^\top$ ,  $x^* = [x_1^*, \dots, x_n^*]^\top$  be an equilibrium point of the system stated in Equation (3.26). If Assumption 3.3.1 holds, then (3.27) is strictly passive from the input  $e_x = [x_1 - x_1^*, \dots, x_n - x_n^*]^\top$  to the output  $e_y = [T_1 - T_1^*, \dots, T_n - T_n^*]^\top$ .*

*Proof.* We use the following positive definite storage function

$$V_1(e_T) = \frac{1}{2} \sum_{i=1}^W \theta_i e_{T_i}^2 \quad (3.28)$$

where  $e_{T_i}$  is the  $i$ th element of the vector  $e_T$ , i.e.,  $e_{T_i} = T_i - T_i^*$ , for  $i = 1, \dots, W$ . The derivative of  $V_1$  along the trajectories of the system stated in Equation (3.27), is given by

$$\dot{V}_1(e_T) = - \left[ e_T^\top L(\mathcal{G}_b) e_T + \sum_{i=1}^W \alpha_{i,a} e_{T_i}^2 \right] + \sum_{i=1}^n e_{T_i} e_{x_i} \quad (3.29)$$

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

where  $e_{x_i}$  is the  $i$ th element of the vector  $e_x$ , i.e.,  $e_{x_i} = x_i - x_i^*$ , for  $i = 1, \dots, n$ . Let us prove that the expression in square brackets is positive definite. Since  $L(\mathcal{G}_b)$  is symmetric, we know that  $e_T^\top L(\mathcal{G}_b) e_T = \frac{1}{2} \sum_{i \in \mathcal{V}_b} \sum_{j \in \mathcal{V}_b} \alpha_{i,j} (e_{T_j} - e_{T_i})^2$  [32], which means that both  $e_T^\top L(\mathcal{G}_b) e_T$  and  $\sum_{i=1}^W \alpha_{i,a} e_{T_i}^2$  are positive semidefinite functions. Furthermore, let us note that:

1.  $e_T^\top L(\mathcal{G}_b) e_T = 0$  if and only if  $e_{T_i} = e_{T_j}$ , for all  $\alpha_{i,j} \neq 0$ . Hence,  $e_T^\top L(\mathcal{G}_b) e_T = 0$  implies that  $e_{T_i} = e_{T_k}$  if there exists a path between  $i$  and  $k$ .
2.  $\sum_{i=1}^W \alpha_{i,a} e_{T_i}^2 = 0$  if and only if  $e_{T_i} = 0$ , for all  $\alpha_{i,a} \neq 0$ .

Mixing conditions 1 and 2, and given the fact that Assumption 3.3.1 holds, we conclude that the expression in brackets is positive definite, i.e.,  $e_T^\top L(\mathcal{G}_b) e_T + \sum_{i=1}^W \alpha_{i,a} e_{T_i}^2 = \psi_1(e_T)$ , where  $\psi_1(e_T) > 0$ . Moreover, since  $\sum_{i=1}^n e_{T_i} e_{x_i} = e_x^\top e_y$ , we have that

$$\dot{V}_1(e_T) = -\psi_1(e_T) + e_x^\top e_y,$$

and the system is strictly passive (cf., Definition 2.2.2). □

Let us note that we have taken the error of the control signals as the input of our system, and the error of the rooms' temperatures as the output (this fact is since, according to Equation (3.25), the  $i$ th output of the system corresponds to the error between the temperature of the  $i$ th room and its respective setpoint, for  $i = 1, \dots, n$ ). We do not include the temperatures of the walls' layers in the output because they are not relevant for any control strategy. The property pointed out in Proposition 3.3.1 is appealing because passivity guarantees the stability of the closed loop response when we add the multi-agent based controller described in ((3.4),(3.6)). Indeed, we notice that the building thermal system (3.27) satisfies **A3**, i.e, it satisfies all the assumptions imposed on the controlled plant in Theorem 3.2.1.

#### 3.3.3 Problem Statement

The main goal of a building temperature control strategy is to keep the temperature of the building's rooms close to a reference, we denote the reference of the  $i$ th room by  $T_i^s$ . The setpoints are not necessarily the same for all the rooms, given the fact that they depend on the preferences of the occupants, the kind of room and its usage. In order to reach the aforementioned goal, we assume that each room of the building has

### 3.3 Distributed Building Temperature Control Under Power Constraints

an actuator (heater). These actuators have some constraints depending on the kind of heating system installed in the building.

A typical configuration, which is common in several buildings, is the central heating system. In this configuration, the total heating power  $Q$  is generated in one place (e.g., a furnace room) and then it is distributed to the  $N$  rooms that comprise the building. Hence, we have the following constraint,

$$\sum_{i=1}^n x_i + x_{n+1} = Q, \quad (3.30)$$

where  $x_i$  is the heating power consumed in the  $i$ th room, for  $i = 1, \dots, n$ ; and  $x_{n+1}$  is the remaining power, i.e., the amount of heating power that is not consumed in any room. We notice that the value of each  $x_i$  is nonnegative. Moreover, it must not exceed the capacity of the corresponding actuator. Hence, we have that

$$\begin{aligned} 0 \leq x_i \leq \bar{x}_i, \text{ for } i = 1, \dots, n \\ x_{n+1} \geq 0 \end{aligned} \quad (3.31)$$

where  $\bar{x}_i$  denotes the maximum capacity of the actuator of the  $i$ th room. If the amount of generated heating power is large enough, it is possible to reach all the temperature setpoints in the building's rooms by using individual controllers, for instance, we can use a proportional-integral (PI) controller per room. However, this strategy does not work properly if the available power is scarce, because the power allocation is not performed in an optimal way (e.g., see [19, 95]). This condition is rare but could occur in buildings with non over-designed heating systems, i.e., when the heating system is not designed to handle worst case loads.

In the latter situation, a good alternative is to use a dynamic resource allocation strategy, where the objective is to distribute the generated heating power among the rooms, in such a way that the building's occupants perceive the same welfare. A measure of the success of the allocation process is given in [19, 61, 95], where the authors propose to use the variance of the differences between the setpoints and the steady state temperatures across the rooms, as follows,

$$\text{Var} = \frac{1}{n} \sum_{i=1}^n ((T_i^s - T_i^*) - \langle T_i^s - T_i^* \rangle)^2, \quad (3.32)$$

where  $\langle \cdot \rangle$  denotes mean value. The smaller the variance, the more successful the allocation done by the control strategy.

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

We recall that buildings are generally large scale systems, which are comprised by a large number of rooms. Hence, the application of centralized control strategies is not suitable. In this regard, the distributed method proposed in Section 3.2 is an appealing alternative to address the proposed problem. Under this method, each room has a controller that makes decision in a coordinated way, and the coordination is made by means of a communication network (i.e., we do not require the inclusion of a centralized agent). In the following section, we provide a detailed description of the design of the distributed controller.

#### 3.3.4 Distributed Control Strategy

The performance criterion given in Equation (3.32), suggests that a good strategy to solve the stated problem is to lead an agreement among the temperature errors in the rooms. In fact, if we assume that all the heating power  $Q$  is supplied to the rooms (i.e.,  $\sum_{i=1}^n x_i = Q$ ), and if we omit the constraints (3.31), then our building temperature control problem is solved by applying the multi-agent approach ((3.4),(3.6)), with  $y_i = T_i - T_i^s$ , and initial conditions satisfying  $\sum_{i=1}^n x_i(0) = Q$ . In this case, notice that Theorem 3.2.1 guarantees that at equilibrium all the temperature errors  $T_i - T_i^s$  are equal; therefore, the variance (3.32) goes to zero. Nonetheless, the assumptions described above are not realistic. First, in situations where there is enough power to reach the temperature references in all the rooms, the power supplied to the rooms is generally lower than  $Q$ , i.e.,  $\sum_{i=1}^n x_i < Q$ . Second, constraints (3.31) cannot be removed since they are closely related to the operation of the actuators. Hence, we need to devise a control strategy that addresses all the requirements stated in the problem formulation. In order to do this, we propose to preserve the multi-agent based controller (3.4), but with some modifications on the resource allocation protocol (3.6).

The first modification is associated with the inclusion of constraints (3.31) by means of memoryless functions. To this end, we change the consensus variable from  $y_i = T_i - T_i^s$  to

$$f_i(T_i, x_i) = (T_i - T_i^s) + \epsilon \varphi_i(x_i), \text{ for all } i = 1, \dots, n, \quad (3.33)$$

where  $\epsilon > 0$  is a small positive constant, and  $\varphi_i(x_i)$  has the following properties:

- *i)*  $\varphi_i(x_i)$  is a monotonically increasing continuous function defined in  $(0, \bar{x}_i)$ ,
- *ii)*  $\varphi_i(x_i) \rightarrow -\infty$ , when  $x_i \rightarrow 0$ ,



### 3.3 Distributed Building Temperature Control Under Power Constraints

- *iii)*  $\varphi_i(x_i) \rightarrow +\infty$ , when  $x_i \rightarrow \bar{x}_i$ .

We can think of  $\varphi_i(x_i)$  as the derivative of a convex barrier function that penalizes the control signal when it tries to deviate outside its feasible domain (we refer the reader to [99] for a detailed description on barrier functions). In this regard, the constant  $\epsilon$  should be designed in such a way that the effect of the function  $\varphi_i(x_i)$  is minimized when the control signal is far from the boundary of its allowable limits.

The second modification allows the rooms to consume less heating power than  $Q$ , i.e.,  $\sum_{i=1}^n x_i \leq Q$ . This is done by including an additional agent ( $n + 1$ ) that manages the remaining power  $x_{n+1}$ . Thus, the modified multi-agent system is characterized by the graph  $\mathcal{G}_c = \{\mathcal{V}_c, \mathcal{E}_c, \mathcal{A}_c\}$ , where  $\mathcal{V}_c = \{1, \dots, n + 1\}$ ,  $\mathcal{E}_c \subset \mathcal{V}_c \times \mathcal{V}_c$ , and  $\mathcal{A}_c = [a_{ij}]$  is the adjacency matrix.

After these changes, we can rewrite the dynamics of the multi-agent based controller as follows,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{ij} (f_j - f_i), \text{ for all } i = 1, \dots, n + 1, \quad (3.34)$$

where  $\mathcal{N}_i = \{j \in \mathcal{V}_c : (i, j) \in \mathcal{E}_c\}$  is the neighborhood of node  $i$  in  $\mathcal{G}_c$ . We notice the fact that  $\mathcal{N}_i$  depends on the communication topology (given by  $\mathcal{G}_c$ ), and not on the building's structure (given by  $\mathcal{G}_b$ ). Furthermore, since we have included the additional node ( $n + 1$ ), we define the corresponding function  $f_{n+1}$  as follows,

$$f_{n+1}(x_{n+1}) = \epsilon \varphi_{n+1}(x_{n+1}) \quad (3.35)$$

where  $\varphi_{n+1}(x_{n+1})$  is a strictly increasing continuous function defined in  $(0, +\infty)$ , which tends to  $-\infty$  when  $x_{n+1}$  tends to 0. Notice that, since the consensus variables  $f_1, \dots, f_{n+1}$  depends on the output of the building thermal system, and on the memoryless term  $\varphi_i(x_i)$  that keeps the control signals feasible, the multi-agent controller (3.34) have the form ((3.4),(3.17)). Therefore, we can use the result in Theorem 3.2.3 to study the properties of the proposed control strategy.

**Proposition 3.3.2.** *Consider the building thermal system (3.26) controlled via the multi-agent network (3.34) with  $f = [f_1, \dots, f_{n+1}]$  given by Equations (3.33) and (3.35). If the initial conditions of the controller satisfy the power constraint (3.30), i.e.,  $\sum_{i=1}^{n+1} x_i = Q$ , and the communication graph  $\mathcal{G}_c$  is connected, then the variable  $f$  asymptotically reaches consensus. Moreover, the control signals satisfy (3.30) for all  $t \geq 0$ .*

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

*Proof.* It is sufficient to show that the closed-loop system ((3.26),(3.34)), with  $f = [f_1, \dots, f_{n+1}]$  given by Equations (3.33) and (3.35), meets the conditions of Theorem 3.2.3.

- First, let us show that ((3.26)(3.34)) has an equilibrium point  $(T^*, [x^{*\top}, x_{n+1}^*]^\top)$  that satisfies  $\sum_{i=1}^{n+1} x_i^* = Q$ : According to Equation (3.27), if  $(T^*, [x^{*\top}, x_{n+1}^*]^\top)$ , is a rest point of (3.26), then

$$0 = -\text{diag}\left(\frac{1}{\theta_1}, \dots, \frac{1}{\theta_n}\right) \underbrace{\left( L(\mathcal{G}_b) + \begin{bmatrix} \alpha_{1,a} & & \\ & \ddots & \\ & & \alpha_{W,a} \end{bmatrix} \right)}_{A_b} T^* \quad (3.36)$$

$$+ \text{diag}\left(\frac{1}{\theta_1}, \dots, \frac{1}{\theta_n}\right) \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0}_{(W-n) \times n} \end{bmatrix} x^* + \mathbf{d},$$

where  $\mathbf{d}$  is a vector of constants that describes the influence of the ambient temperature and the external disturbances on the building thermal system. Hence,

$$T^* = A_b^{-1} \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0}_{(W-n) \times n} \end{bmatrix} x^* + \underbrace{A_b^{-1} \text{diag}^{-1}\left(\frac{1}{\theta_1}, \dots, \frac{1}{\theta_n}\right) \mathbf{d}}_{\mathbf{d}'}. \quad (3.37)$$

The matrix  $A_b$  is invertible since it is positive definite (see the proof of Proposition 3.3.1).

On the other hand, the multi-agent network (3.34) must satisfy at equilibrium

$$-L(\mathcal{G}_c) f(T^*, [x^{*\top}, x_{n+1}^*]^\top) = 0, \quad (3.38)$$

where

$$f(T^*, [x^{*\top}, x_{n+1}^*]^\top) = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times (W-n)} \\ \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times (W-n)} \end{bmatrix} T^* - [T_1^s, \dots, T_n^s, 0]^\top + \epsilon [\varphi_1(x_1^*), \dots, \varphi_{n+1}(x_{n+1}^*)]^\top.$$

Since  $\mathcal{G}_c$  is connected by assumption, the null space of  $L(\mathcal{G}_c)$  is equal to the linear span of  $\mathbf{1}$  (cf., Theorem 2.1.1). Therefore, Equation (3.38) is satisfied if and only if

$$\xi \mathbf{1} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times (W-n)} \\ \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times (W-n)} \end{bmatrix} T^* - [T_1^s, \dots, T_n^s, 0]^\top + \epsilon [\varphi_1(x_1^*), \dots, \varphi_{n+1}(x_{n+1}^*)]^\top,$$

### 3.3 Distributed Building Temperature Control Under Power Constraints

where  $\xi \in \mathbb{R}$ . Replacing the value of  $T^*$  given in Equation (3.37), we obtain

$$\xi \mathbf{1} = \begin{bmatrix} (A_b^{-1})^n \\ \mathbf{0}_{1 \times n} \end{bmatrix} x^* + \epsilon [\varphi_1(x_1^*), \dots, \varphi_{n+1}(x_{n+1}^*)]^\top + \mathbf{d}'', \quad (3.39)$$

where  $(A_b^{-1})^n$  is the submatrix formed by the  $n$  first rows and the  $n$  first columns of matrix  $A_b$ , and  $\mathbf{d}''$  is a vector of constants, which is defined by

$$\mathbf{d}'' = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times (W-n)} \\ \mathbf{0}_{1 \times n} & \mathbf{0}_{1 \times (W-n)} \end{bmatrix} \mathbf{d}' - [T_1^s, \dots, T_n^s, 0]^\top.$$

According to the Karush–Kuhn–Tucker first order necessary conditions, it follows from (3.39), that finding an equilibrium point of ((3.26),(3.34)) that satisfies the power constraint (3.30) is equivalent to solve the following optimization problem

$$\begin{aligned} \min_{(x^*, x_{n+1}^*)} \quad & x^{*\top} (A_b^{-1})^n x^* + \mathbf{d}''^\top [x^{*\top}, 0]^\top + \epsilon \Phi(x^*, x_{n+1}^*) \\ \text{subject to} \quad & \sum_{i=1}^{n+1} x_i = Q \end{aligned} \quad (3.40)$$

where  $\Phi(x^*, x_{n+1}^*)$  satisfies  $\nabla \Phi = [\varphi_1(x_1^*), \dots, \varphi_{n+1}(x_{n+1}^*)]^\top$ , and  $(A_b^{-1})^n$  is positive definite since it is a strictly diagonally dominant matrix (see Equation (3.36)). Notice that, the objective function of (3.40) is strictly convex since  $(A_b^{-1})^n$  is positive definite and the gradient of  $\Phi$  is comprised of monotonically increasing functions. Therefore, the optimization problem (3.40) has a unique solution. This implies that ((3.26),(3.34)) has an equilibrium point that satisfies the power constraint (3.30). Moreover, this equilibrium point is unique.

- **A1** holds by assumption.
- **A2** holds since the closed loop system ((3.26),(3.34)) has an equilibrium point satisfying  $\sum_{i=1}^{n+1} x_i^* = Q$ , and the initial conditions satisfy  $\sum_{i=1}^{n+1} x_i(0) = Q$  by assumption.
- **A3**: The fact that (3.26) expressed in error coordinates is strictly passive, from the input  $[x_1 - x_1^*, \dots, x_{n+1} - x_{n+1}^*]^\top$  to the output  $[T_1 - T_1^*, \dots, T_n - T_n^*, 0]^\top$ , follows from Proposition 3.3.1. Moreover, the storage function is radially unbounded.
- **A4**: Notice that the memoryless term  $\varphi_i(x_i)$  satisfies the following property inside the feasible domain of  $x_i$ :

$$(\varphi_i(x_i) - \varphi_i(x_i^*))(x_i - x_i^*) \geq 0, \text{ for all } i = 1, \dots, n + 1. \quad (3.41)$$

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

This is true since  $\varphi_i(x_i)$  is monotonically increasing. Therefore, the memoryless function  $[(\varphi_1(x_1) - \varphi_1(x_1^*)), \dots, (\varphi_{n+1}(x_{n+1}) - \varphi_{n+1}(x_{n+1}^*))]^\top$  is passive with respect to the error vector  $[x_1 - x_1^*, \dots, x_{n+1} - x_{n+1}^*]^\top$ .

- **A5:** Since the  $\varphi_i(x_i)$  is monotonically increasing, for all  $i = 1, \dots, n + 1$ , then  $\varphi_i(x) - \varphi_i(x_i^*) = 0$  if and only if  $x_i = x_i^*$ . Thus, Assumption 3.2.3 holds.

In conclusion, all assumptions of Theorem 3.2.3 hold. This proves the statement in Proposition 3.3.2. □

Let us analyze the importance of Proposition 3.3.2.

#### 3.3.4.1 Power Constraints

A direct consequence of Proposition 3.3.2 is that the proposed resource allocation algorithm prevents the calculated control signal to violate the constraint given in Equation (3.30), i.e, the algorithm guarantees that the sum of the powers allocated in the rooms of the building plus the remaining power (which is not used to heat the rooms) is always equal to the generated heating power  $Q$ .

On the other hand,  $\varphi_i(\cdot)$  acts as a barrier function that forces the  $i$ th control signal  $x_i$  to satisfy the constraint (3.31). In order to explain the effect of  $\varphi_i(\cdot)$ , let us consider the following situation: if  $x_i$  is close enough to its upper bound  $\bar{x}_i$ , then the value of the corresponding function  $f_i$  is higher than the value of the other functions, this happens given the fact that  $\varphi_i(x_i)$  (which is a component of  $f_i$ ) is a monotonically increasing continuous function that tends to  $+\infty$  when  $x_i$  approximates to its upper bound. When the above occurs,  $\dot{x}_i$  is negative (according to Equation (3.34)), and therefore  $x_i$  decreases and does not exceed the upper bound  $\bar{x}_i$ . The opposite happens when  $x_i$  is close to its lower bound.

Summarizing, satisfaction of constraints (3.30) and (3.31) is guaranteed under the proposed control strategy.

#### 3.3.4.2 Equilibrium Points

In this subsection, we use the result given in Proposition 3.3.2 to characterize the equilibrium points of the building thermal system controlled via the algorithm proposed in

### 3.3 Distributed Building Temperature Control Under Power Constraints

Equation (3.34) under different conditions. Moreover, we show that these equilibrium points solve the problem stated in Section 3.3.3.

Proposition 3.3.2 entails that, at equilibrium, all the functions  $f_1, \dots, f_{n+1}$ , are equal. For a better understanding of the implications of this feature, we define  $\hat{f}_i(T_i) = f_i(T_i, x_i) - \epsilon\varphi_i(x_i) = T_i - T_i^s$ , as a function that only depends on the temperature of the  $i$ th room and its corresponding setpoint. Furthermore, following the same definition, we have that  $\hat{f}_{n+1} = f_{n+1}(x_{n+1}) - \epsilon\varphi_{n+1}(x_{n+1}) = 0$ . Using the auxiliary functions  $\hat{f}_1, \dots, \hat{f}_{n+1}$ , let us analyze the properties of the equilibrium point  $(T^*, [x^{*\top}, x_{n+1}^*]^\top)$ .

- *Normal conditions:* if for all  $i = 1 \dots, n + 1$ ,  $x_i^*$  is inside its feasible domain and far from its allowable limits (which are given in Equation (3.31)), then the value of  $\epsilon\varphi_i(x_i^*)$  can be neglected (given the fact that  $\epsilon$  is small). Therefore, we have that  $f_i \approx \hat{f}_i$ , for all  $i = 1 \dots, n + 1$ ; and according to Proposition 3.3.2,  $f_i(T_i^*, x_i^*) = f_{n+1}(x_{n+1}^*)$ , for all  $i = 1, \dots, n$ . Hence,  $\hat{f}_i(T_i^*) = T_i^* - T_i^s \approx 0$ , for all  $i = 1, \dots, n$ . This means that the temperature setpoints can be reached by using the proposed control strategy, as long as the power required in each room satisfies its corresponding constraints.
- *Scarce heating power:* another scenario emerges when, at steady state, all the control signals are inside their feasible domains and far from their allowable limits, except the remaining power, i.e.,  $x_{n+1}^* \rightarrow 0$ . Let us note that, in this case,  $\epsilon\varphi_i(x_i^*)$  can be neglected for  $i = 1, \dots, n$ , but not for  $i = n + 1$ , where we have that  $\epsilon\varphi_{n+1}(x_{n+1}^*) < 0$ . Therefore, applying Proposition 3.3.2, we can conclude that  $f_i = T_i^* - T_i^s \approx \epsilon\varphi_{n+1}(x_{n+1}^*) < 0$ , for all  $i = 1, \dots, n$ . This implies that, despite the temperatures of the rooms do not reach the corresponding setpoints due to lack of heating power, all the steady state errors ( $T_i^* - T_i^s$ , for each  $i = 1, \dots, n$ ) are almost equal, minimizing the variance given in Equation (3.32).
- *General case:* Finally, let us assume a general case, where in steady state, all the control signals are inside their feasible domains, but only a set of them, denoted by  $\mathcal{X}$ , is far from the allowable limits, i.e.,

$$\mathcal{X} = \{x_i^* : x_i^* \text{ is not close to } 0, \text{ and } x_i^* \text{ is not close to } \bar{x}_i\};$$

while other set of signals, denoted by  $\underline{\mathcal{X}}$ , is close to the lower bound, i.e.,

$$\underline{\mathcal{X}} = \{x_i^* : x_i^* \rightarrow 0\};$$

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

and other set of signals, denoted by  $\bar{\mathcal{X}}$ , is close to the upper bound, i.e.,

$$\bar{\mathcal{X}} = \{x_i^* : x_i^* \rightarrow \bar{x}_i\}.$$

As in the previous scenarios,  $\epsilon\varphi_i(x_i^*) \approx 0$  if  $x_i^* \in \mathcal{X}$ . However, this is not the case if  $x_i^*$  belongs to the other sets. Using the properties of the function  $\varphi_i(\cdot)$ , it can be verified that  $\epsilon\varphi_i(x_i^*) < 0$  if  $x_i^* \in \underline{\mathcal{X}}$ , while  $\epsilon\varphi_i(x_i^*) > 0$  if  $x_i^* \in \bar{\mathcal{X}}$ . Hence, by using the result in Proposition 3.3.2, we can conclude that the following relationships hold at equilibrium:

- $\hat{f}_i \approx \hat{f}_j$  if  $x_i^* \in \mathcal{X}$  and  $x_j^* \in \mathcal{X}$ .
- $\hat{f}_i < \hat{f}_j$  if  $x_i^* \in \mathcal{X}$  and  $x_j^* \in \underline{\mathcal{X}}$ .
- $\hat{f}_i > \hat{f}_j$  if  $x_i^* \in \mathcal{X}$  and  $x_j^* \in \bar{\mathcal{X}}$ .

Moreover, since  $\hat{f}_i(T_i^*) = T_i^* - T_i^s$ , for all  $i = 1, \dots, n$ , the above relationships imply that no heating power is assigned to the rooms that are warmer to the average, while the colder rooms receive the maximum allowable power. In this regard, the heating power is optimally allocated.

Summarizing, the equilibrium point of the closed loop system ((3.26),(3.34)) exhibits optimal properties under different scenarios of the building temperature control problem. Besides, since this equilibrium point is asymptotically stable according to Proposition 3.3.2, we can conclude that the multi-agent approach (3.34) properly solves the building temperature control problem described in Section 3.3.3.

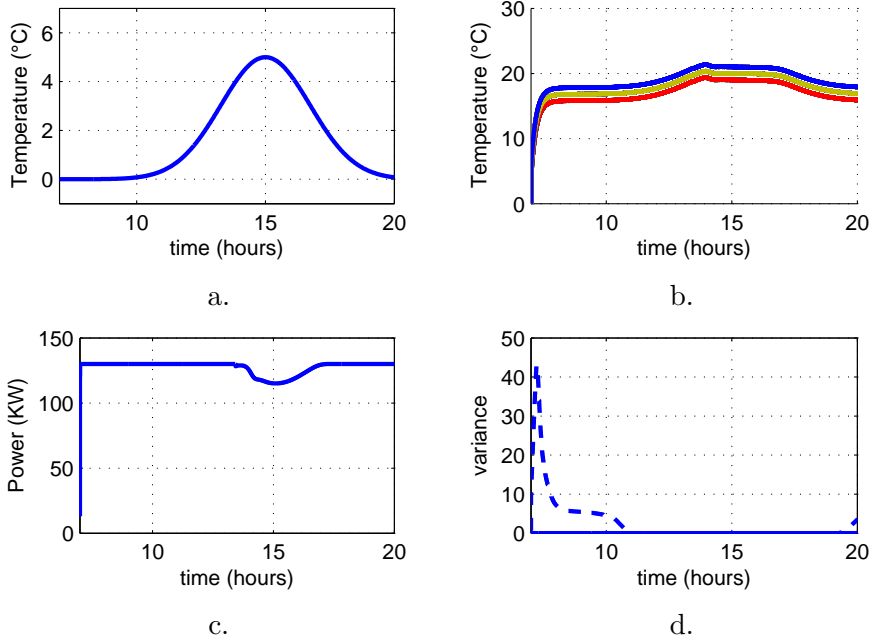
#### 3.3.5 Simulation Results

In order to illustrate the performance of the proposed control technique, we simulate the thermal behavior of a building comprised of 50 rooms that are arranged as in Figure 3.3. The building has a central heating system of 130 KW, with one actuator per room. The maximum power that each actuator can provide is 3.25 KW. Moreover, the temperature setpoints are: 19 °C for rooms 1–17, 20 °C for rooms 18–34, and 21 °C for rooms 34–50. In the simulation, we assume that the outdoor temperature changes according to Figure 3.5a.

The results obtained with the proposed controller and a communication network with ring topology, are shown in Figures 3.5b–3.5d. Figure 3.5b shows the evolution

### 3.3 Distributed Building Temperature Control Under Power Constraints

of the temperatures in each room from 7h to 20h, Figure 3.5c shows the total heating power used by the rooms, and Figure 3.5d shows the performance index related to the variance of the error across the rooms (cf., Equation (3.32)). It can be noticed that, from 7h–13h it is not possible to reach the setpoints despite the fact that all the available power is used to heat the rooms. However, the controller minimizes the variance of the error across the rooms to a value of zero, which implies that the building’s occupants are perceiving the same welfare (i.e., the *scarce power* situation described in Section 3.3.4.2). When the weather is warmer, i.e., around 13h–17h, the temperature in each room rises and reaches the corresponding setpoint, while the power consumption is reduced. In the last part of the simulation, we have a similar behavior as the one described in the first part, due to the decrease of the outdoor temperature.



**Figure 3.5:** a. Ambient temperature during a day (7h–20h). b. Temperatures of the building’s rooms. Rooms 1–17 (red), rooms 18–34 (yellow), rooms 35–50 (blue). c. Heating Power allocated in the building’s rooms. d. Variance of the error across the rooms. Distributed controller (solid line), PI controllers (dashed line).

Finally, we compare the performance of the proposed technique with the most widely used method in building temperature control, i.e., PI control [90]. In order to do that, we implement one PI controller per room in the same scenario described above. Each

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

PI controller seeks to regulate the temperature of its corresponding room without any information on the state of the other rooms, i.e., their decisions are not coordinated. Moreover, we notice the fact that the maximum power that each PI controller can use depends on both the actuator capacity and the consumption of the other rooms. The variance of the errors across the rooms with this technique is depicted in dashed line in Figure 3.5d. It can be noticed that individual PI controllers do not work properly when the outdoor temperature is lower and the available heating power is not enough to reach all the setpoints, i.e., from 7h–11h, and from 19h–20h; in that time intervals, the variance has a large value. This adverse effect is due to an inefficient power allocation.

#### 3.3.6 Discussion

When the generated heating power is enough to reach the temperature setpoints of the building’s rooms, the behavior of the closed loop system controlled via the proposed algorithm is the same as the behavior achieved with individual PI controllers, i.e., all the temperature references are reached without steady state error. However, under a scarce power scenario, PI controllers do not allocate the available resource in an optimal way, as shown in Figure 3.5d, where the variance of the temperature errors across the rooms has a high value at critical load hours. This fact is due to a lack of coordination among the controllers. The proposed algorithm does not have that problem because the communication network allows the agents to share information on the state of the neighboring rooms’ temperatures. In this way, each agent is able to make decisions that improve the performance of the overall system, rather than the performance of a single room.

Optimal results can also be obtained by using model predictive control (centralized or distributed). However, although that technique offers wide formulation flexibility and its predictive features can enhance the energy saving, it suffers from high computational requirements and model dependence, which reduces its scalability, especially for large buildings with not negligible thermal coupling among rooms [100]. In this regard, the distributed methodology that we propose has the advantage of being formulated as a set of “quasi-linear” differential equations, whose implementation does not demand high computation resources. Hence, its scalability is substantially higher than that provided by receding horizon techniques.



### 3.4 Optimal Resource Allocation in a Network of Agents

The classical resource allocation problem arises when we have a scarce resource, and it is necessary to distribute it among a set of entities in order to optimize a given objective function. We talk about resource allocation over networks of agents when the resource allocated to each entity is managed by an agent that only has partial information of the overall optimization problem. In order to reach the optimal allocation, the agents coordinate their decisions by using a communication network with a given topology. If we have a set of  $n$  entities and we assign one agent per entity, the aforementioned problem can be expressed mathematically as follows [11],

$$\begin{aligned} \min \quad & \phi(x) \\ \text{subject to} \quad & \sum_{i=1}^n x_i = X \end{aligned} \tag{3.42}$$

where  $\phi : \mathbb{R}^n \mapsto \mathbb{R}$  is a strictly convex and differentiable cost function,  $X \in \mathbb{R}$  is a given constant related to the total amount of resource,  $x_i$  is the amount of resource allocated to the  $i$ th entity, and  $x = [x_1, \dots, x_n]^\top$ . In the literature on distributed optimization [7], it is common to assume that the  $i$ th agent knows the marginal cost associated with its corresponding entity, i.e.,  $\frac{\partial \phi}{\partial x_i}$ . Moreover, this information can be shared by the agents using the available communication network, which is modeled by the graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ .

The local information structure imposed by the graph  $\mathcal{G}$  must be considered as part of the problem statement. Therefore, the main objective is to formulate a distributed algorithm that can be implemented by each agent to solve (3.42) using only its available information. In this case, notice that the available information of the  $i$ th agent is given by  $\left( \frac{\partial \phi}{\partial x_i}, \left\{ \frac{\partial \phi}{\partial x_j} : j \in \mathcal{N}_i \right\} \right)$ .

A well-known method to address (3.42) that satisfies the local information constraint is formulated as follows [24, 35].

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} a_{i,j} \left( \frac{\partial \phi}{\partial x_j} - \frac{\partial \phi}{\partial x_i} \right), \text{ for all } i = 1, \dots, n. \tag{3.43}$$

Although the optimality of (3.43) has been proved in [36], we are interested in analyzing this method from the perspective of passive memoryless systems given in Section 3.2.4.

Notice that (3.43) is equivalent to the resource allocation algorithm given by Equations (3.4) and (3.6), where  $y = \nabla \phi(x)$ . Let us show that the memoryless system

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

$y = \nabla\phi(x)$  is passive from the input  $x - x^*$  to the output  $y - y^*$ , where  $x^*$  is the global optimum of (3.42),  $y^* = \nabla\phi(x^*)$ , and  $x$  satisfies the resource constraint, i.e.,  $\sum_{i=1}^n x_i = X$ . To this end, we use the following facts:

- Since  $\phi$  is strictly convex, then this satisfies the inequality  $\phi(x^*) > \phi(x) + \nabla\phi(x)(x^* - x)$ , for all  $x \neq x^*$  (see, e.g., [101]). Given the fact that  $x^*$  is the global optimum of (3.42), then  $\nabla\phi(x)(x - x^*) > 0$ , for all  $x \neq x^*$  that satisfies  $\sum_{i=1}^n x_i = X$ .
- According to the Karush–Kuhn–Tucker first order necessary conditions,  $\nabla\phi(x^*)$  belongs to the linear span of  $\mathbf{1}$ , i.e.,  $\nabla\phi(x^*) = \lambda\mathbf{1}$ , where  $\lambda \in \mathbb{R}$ . Thus,  $\nabla\phi(x^*)(x - x^*) = \lambda(\mathbf{1}^\top x - \mathbf{1}^\top x^*)$ . Since both  $x$  and  $x^*$  satisfy the resource constraint, we can conclude that  $\nabla\phi(x^*)(x - x^*) = 0$ .

Combining these two facts, we obtain  $(\nabla\phi(x) - \nabla\phi(x^*))(x - x^*) > 0$ , for all  $x \neq x^*$ . Thus, the considered memoryless system is passive. Furthermore, notice that this system satisfies Assumption 3.2.2 since  $(\nabla\phi(x) - \nabla\phi(x^*))(x - x^*) = 0$  only if  $x = x^*$ .

In conclusion, the result in Theorem 3.2.2 can also be used to guarantee the convergence of (3.43) to the optimum of (3.42) as long as  $\sum_{i=1}^n x_i(0) = X$ .

#### 3.4.1 Inclusion of Inequality Constraints by means of Barrier Functions

We can take advantage of the convexity of  $\phi$  to include in (3.42) inequality constraints of the form

$$\underline{x}_i \leq x_i \leq \bar{x}_i, \text{ for all } i = 1, \dots, n, \quad (3.44)$$

where  $\underline{x}_i$  and  $\bar{x}_i$  are, respectively, the lower and upper bound of  $x_i$ . This is done by using convex barrier functions that keep the decision variables strictly inside the feasible set. Barrier functions are continuous functions defined inside the feasible domain whose value increases to infinity when the variable approaches to the boundary of its feasible domain. In our specific case, we use a strictly convex barrier function for each inequality constraint (3.44), as follows [99]:

$$b_i(x_i) = -\varepsilon(\ln(x_i - \underline{x}_i) + \ln(\bar{x}_i - x_i)), \forall i = 1, \dots, n, \quad (3.45)$$

where  $\varepsilon > 0$  is a constant that minimizes the effect of the barrier function when the solution is far from the boundary of the feasible set. These barrier functions are added

to the objective function  $\phi(x)$  in Equation (3.42), i.e., our new objective function becomes

$$\phi_b(x) = \phi(x) - \varepsilon \sum_{i=1}^n (\ln(x_i - \underline{x}_i) + \ln(\bar{x}_i - x_i)).$$

Notice that, if  $\phi$  is strictly convex,  $\phi_b$  remains strictly convex. Moreover, although the solution of the optimization problem that includes barrier functions is suboptimal for the original problem, this solution can be sufficiently close to the optimum depending on the value of  $\varepsilon$ . Furthermore, since  $b_i$  only depends on  $x_i$ , the algorithm (3.43) remains distributed if we replace  $\frac{\partial \phi}{\partial x_i}$  by  $\frac{\partial \phi_b}{\partial x_i}$ , for all  $i = 1, \dots, n$ .

#### 3.4.2 Initial Feasible Solution

The algorithm given in (3.43) needs an initial feasible solution of the corresponding optimization problem. Although finding this starting point is straightforward if the optimization problem does not include the inequality constraints (3.44) (in this case, we can take  $(x_k(0) = X, x_i(0) = 0, \text{ for all } i = 1 \dots, n, i \neq k)$  as the initial feasible solution), it is not trivial if they are considered. If this is the case, a preliminary phase that provides us the required feasible starting point is required. To implement the preliminary phase, we use a continuous-time version of the the distributed constraint satisfaction algorithm described in [70].

In order to maintain the distributed nature of our resource allocation problem, we assume that only one agent knows the total amount of available resource  $X$ . The main idea of the preliminary phase is to solve two auxiliary problems that use local information, and then combining the solutions to get the starting point for the main problem ((3.42), (3.44)). Let  $\beta_{1_i}$  and  $\beta_{2_i}$ , two auxiliary variables associated with the  $i$ th node. Using these variables, we establish the following auxiliary problems,

$$\begin{aligned} \text{Problem 1 : } \min \quad & \phi_1(\beta_1) = \beta_1^\top \beta_1, \\ \text{s. t.} \quad & \mathbf{1}^\top \beta_1 = d - \sum_{i=1}^n \underline{x}_i \\ \\ \text{Problem 2 : } \min \quad & \phi_2(\beta_2) = \beta_2^\top \beta_2 \\ \text{s. t.} \quad & \mathbf{1}^\top \beta_2 = \sum_{i=1}^n (\bar{x}_i - \underline{x}_i) \end{aligned} \tag{3.46}$$

where  $\beta_1 = [\beta_{1_1}, \dots, \beta_{1_n}]^\top$  and  $\beta_2 = [\beta_{2_1}, \dots, \beta_{2_n}]^\top$ . It can be shown that the optimal solutions of Problems 1 and 2 are,  $\beta_1^* = \frac{d - \sum_{i=1}^n \underline{x}_i}{n} \mathbf{1}$  and  $\beta_2^* = \frac{\sum_{i=1}^n (\bar{x}_i - \underline{x}_i)}{n} \mathbf{1}$ , respectively.

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

Moreover, since  $\phi_1$  and  $\phi_2$  are strictly convex, these optimal solutions can be obtained by using the distributed algorithm described in Equation (3.43) with the following initial feasible solutions:

$$\beta_{1_i}(0) = \begin{cases} d - \underline{x}_i & \text{if } i = k \\ -\underline{x}_i & \text{otherwise} \end{cases},$$

$$\beta_{2_i}(0) = \bar{x}_i - \underline{x}_i, \quad \text{for all } i = 1, \dots, n.$$

where we assume, without loss of generality, that the  $k$ th agent knows the value  $X$ . We notice the fact that the agents only require local information to obtain  $\beta_1(0)$  and  $\beta_2(0)$ . Using the optimal solutions of Problems 1 and 2 in Equation (3.46), we can obtain a feasible solution of the main problem ((3.42), (3.44)), as follows,

$$x_i(0) = \underline{x}_i + (\bar{x}_i - \underline{x}_i) \begin{pmatrix} \frac{\beta_{1,i}^*}{\beta_{2,i}^*} \end{pmatrix}, \text{ for all } i = 1, \dots, n.$$

Let us note that if  $0 \leq \frac{\beta_{1,i}^*}{\beta_{2,i}^*} \leq 1$ , then this initial solution satisfies all the constraints of our resource allocation problem ((3.42), (3.44)). Moreover if  $\frac{\beta_{1,i}^*}{\beta_{2,i}^*} < 0$  or  $\frac{\beta_{1,i}^*}{\beta_{2,i}^*} > 1$ , the problem is infeasible.

#### 3.4.3 Application to Electric Load Aggregation for Demand Response

In this section, we present an application of the optimal resource allocation in networks of agents. The considered application deals with the problem of electric load aggregation for demand response [102]. This problem is motivated by the fact that the active participation of demand is a central issue in the smart grid concept, in this regard, aggregators play a fundamental role. Aggregators are mediators between consumers, who provide their demand flexibilities, and the other participants of electricity markets (e.g., distribution/transmission system operators, retailers, electricity brokers). So, when a demand deviation is required (e.g., due to imbalances in the distribution network), aggregators can provide it by acting on the loads of their clients. In return to these possible load curtailments, aggregators' clients pay lower electricity prices. Moreover, since each aggregator generally has a large number of clients, distributed management strategies are becoming more relevant.

The operation of an aggregator can be viewed as a resource allocation process, where the total amount of load deviation that the aggregator has to provide at certain time,

### 3.4 Optimal Resource Allocation in a Network of Agents

---

needs to be split between their clients (consumers). One possibility is to do an allocation of the necessary load curtailment considering only technical constraints as it is described in [70]. However, a better opportunity emerges given the fact that the aggregator can take advantage of the aforementioned allocation process to generate other benefits. For instance, it is important to encourage the clients to stay involved in demand response programs by minimizing the impact that they perceive due to the load curtailments. To formally describe this situation, let us assume that the aggregator manages a set of  $n$  consumers. The authors in [62], propose to model the benefit perceived by the  $i$ th consumer as follows,

$$u_i(p_i) = k_{i,1} - k_{i,2}e^{-k_{i,3}p_i} - k_{i,4}p_i, \quad (3.47)$$

where  $p_i$  is the electric power being used by the  $i$ th consumer,  $k_{i,1}, k_{i,2}, k_{i,3}$ , and  $k_{i,4}$  are positive parameters that depend on the consumer's preferences. Moreover, if the consumer suffers a load curtailment  $x_i$ , the new benefit that the consumer perceives is  $u_i(p_i - x_i)$ . Therefore, the cost of the load curtailment for the  $i$ th consumer is given by

$$c_i(x_i) = u_i(p_i) - u_i(p_i - x_i). \quad (3.48)$$

It can be shown that  $c_i$  is a strictly convex function of  $x_i$ .

In this regard, the aggregator must satisfy the required load deviation  $D_L$ , minimizing at the same time, the total cost perceived by its clients, i.e.,  $\phi(x) = \sum_{i=1}^n c_i(x_i)$ . Mathematically, this problem is formulated as follows

$$\begin{aligned} \min \quad & \phi(x) = \sum_{i=1}^n c_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^n x_i = D_L \\ & 0 \leq x_i \leq p_i, \text{ for all } i = 1, \dots, n, \end{aligned} \quad (3.49)$$

where  $x_i$  is the load curtailed from the  $i$ th consumer. The inequality constraints model the fact that the aggregator cannot curtail more load than that used by the consumer.

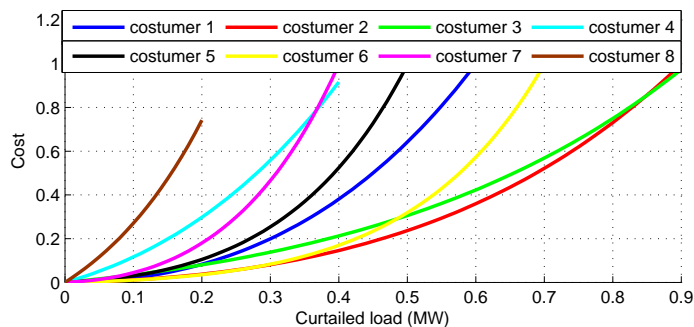
As previously mentioned, an aggregator can have a large number of clients, e.g., small consumers. Hence, a centralized operation is not the best option. Moreover, in a smart grid environment, it is assumed that each consumer has computation and communication capabilities. For these reasons, the optimization problem given in Equation (3.49) can be appropriately approached by means of a multi-agent system that uses the distributed protocol given in Equation (3.43). In order to do that, we assign one agent per client. Each agent knows (or estimates) the function describing the benefit of the corresponding

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---

client, and its current consumption. The coordination of the agents' decisions is done through a communication network with a topology given by a connected graph  $\mathcal{G}$ .

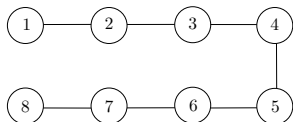
To illustrate the performance of the distributed resource allocation algorithm, we simulate the operation of an aggregator that manages 8 clients, and uses a communication network with single-path topology (see Figure 3.7). Let us suppose that at certain time, the power consumptions of the clients are:  $p_1 = 0.6$  MW,  $p_2 = 0.9$  MW,  $p_3 = 0.9$  MW,  $p_4 = 0.4$  MW,  $p_5 = 0.5$  MW,  $p_6 = 0.7$  MW,  $p_7 = 0.4$  MW,  $p_8 = 0.2$  MW. Furthermore, at that time, the aggregator has to provide a load deviation (i.e., load curtailment)  $D_L = 2.7$  MW. The parameters of the benefit function (3.47) for each client are given in Table 4.2, and the corresponding cost functions (3.48) are depicted in Figure 3.6. Let us note that the cost functions for clients 1 and 4 (5 and 8) are different despite the fact that their corresponding benefit functions have the same parameters. This happens due to the different power consumption levels. Moreover, in order to address the inequality constraints of the problem, we use logarithmic barrier functions, as it was explained in Section 3.4.1. The evolution of the distributed resource allocation algorithm (3.43) is depicted in Figure 3.8. In this case, we use the following initial feasible point:  $x_1(0) = 0.35$  MW,  $x_2(0) = 0.53$  MW,  $x_3(0) = 0.53$  MW,  $x_4(0) = 0.23$  MW,  $x_5(0) = 0.30$  MW,  $x_6(0) = 0.41$  MW,  $x_7(0) = 0.23$  MW,  $x_8(0) = 0.12$  MW. It can be noticed that the steady state solution satisfies the technical constraints of the problem, i.e.,  $\sum_{i=1}^n x_i^* = D_L = 2.7$  MW, and  $0 \leq x_i^* \leq p_i$ , for  $i = 1, \dots, n$ . Furthermore, Figure 3.9 shows that the total cost function decreases and finally reaches a reduction of 13.7%. In this regard, it is worth noting that the impact of load curtailment on the aggregator's clients is reduced by minimizing the cost function stated in Equation (3.48). This can be seen as an incentive for clients to keep involved in demand response programs.



**Figure 3.6:** Cost function for each aggregator's client.

**Table 3.1:** Utility functions' parameters

	$k_{i,1}$	$k_{i,2}$	$k_{i,3}$	$k_{i,4}$
$i = 1$	2.95	2.95	2.00	1.77
$i = 2$	1.77	1.77	2.00	0.53
$i = 3$	1.31	1.31	2.00	0.13
$i = 4$	2.95	2.95	2.00	1.77
$i = 5$	1.30	1.30	5.00	0.39
$i = 6$	1.11	1.11	5.00	0.11
$i = 7$	1.60	1.60	5.00	0.96
$i = 8$	1.30	1.30	5.00	0.39



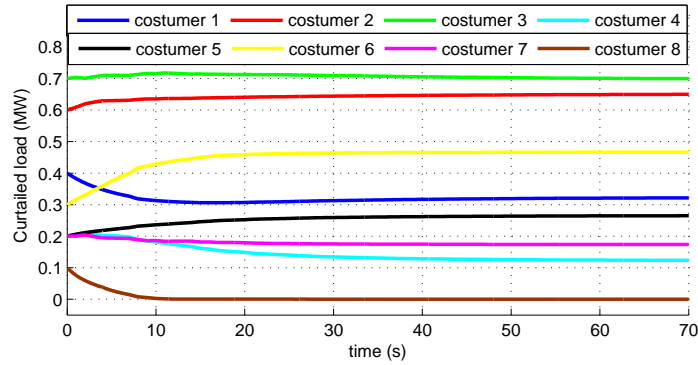
**Figure 3.7:** Communication network with single-path topology.

### 3.4.4 Discussion

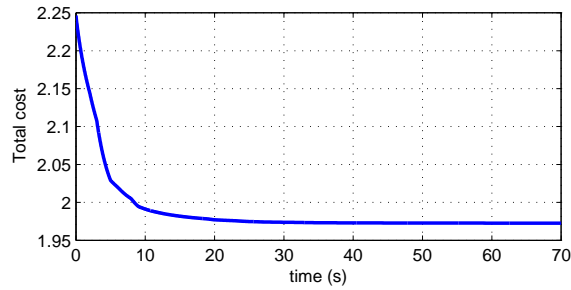
The classic resource allocation problem (3.42) have been widely studied in the literature on optimization (see e.g., [11] and the references therein). In fact, the properties of the algorithm stated in Equation (3.43) are well-known [24, 35, 36]. Therefore, the optimality analysis of (3.43) cannot be considered as an original contribution of Section 3.4. The purpose of this section has been to show a novel perspective of the classic resource allocation problem from passivity theory. We have proven that there exists an equivalence between (3.42) and a passive memoryless system that is given by the gradient of the cost function as long as this cost function is convex. Thus, we have shown that the result given in Theorem 3.2.2 is applicable to the solution of the optimal resource allocation problem via the algorithm (3.43). On the other hand, Section 3.4.2 have described a constraint satisfaction algorithm that can be employed to initialize (3.43) when inequality constraints of the form (3.44) are added to the problem formulation. Notice that, the same initialization phase is required in the multi-agent scheme proposed to address the building temperature control problem (see Section 3.3). Regarding the

### 3. DISTRIBUTED RESOURCE ALLOCATION AMONG DYNAMIC NETWORKS

---



**Figure 3.8:** Evolution of the decentralized resource allocation algorithm.



**Figure 3.9:** Evolution of the total cost perceived by the aggregator's clients.

inequality constraints (3.44), it has been shown that they can be satisfied by introducing convex barrier functions. However, this approach decreases the accuracy of the solution. Indeed, in the next chapter, we will show that the adverse effect caused by the influence of barrier function is more pronounced in large-scale applications. A solution to this problem will also be proposed in the next chapter.

Finally, we have described an application to the distributed management of the customers of an aggregating entity in a demand response environment. We notice the fact that demand response programs, where consumers allow an aggregator to curtail part of their load, are not exclusively applied by commercial and industrial users nowadays. They have been extended to residential consumers. This fact has greatly increased the number of clients that aggregators need to manage. In this regard, the development and analysis of distributed techniques, as the one shown in Section 3.4.3, is important to properly deal with large number of users. Hence, these techniques facilitate the inclusion of new participants in demand response programs.



## 4

# Optimal Resource Allocation with Lower-Bound Constraints

In this chapter, we go deeper into the study of the optimal resource allocation problem stated in Section 3.4. Specifically, we focus our attention on solution methods for problems that include lower-bound constraints in the resource allocated to each entity. The capability of handling this kind of constraints is crucial in a large number of practical applications, e.g., in [12, 15, 16, 36, 54, 71, 72], where it is required to capture the non-negativity of the resource allocated to each entity.

The algorithm proposed in this chapter is motivated by the fact that, although lower-bound constraints can be addressed by using barrier functions (cf., Section 3.4.1), the inclusion of these functions affect the accuracy of the solution, especially for large-scale problems [74]. Therefore, it is necessary to develop other methods whose accuracy does not depend on the problem scale.

## 4.1 Problem Statement

We recall that the resource allocation problem arises when there is a limited amount of a certain resource (e.g., electric power, computing capacity, or execution time), and it is necessary to establish an optimal distribution policy between some entities (e.g., loads, processors, or controllers). If we impose lower-bound constraints to the resource

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

allocated to each entity, the problem is formulated as follows [11],

$$\min_x \phi(x) := \sum_{i=1}^n \phi_i(x_i) \quad (4.1a)$$

$$\text{subject to } \sum_{i=1}^n x_i = X \quad (4.1b)$$

$$x_i \geq \underline{x}_i, \text{ for all } i = 1, \dots, n, \quad (4.1c)$$

where  $x_i \in \mathbb{R}$  is the resource allocated to the  $i$ th entity;  $x = [x_1, \dots, x_n]^\top$ ;  $\phi_i : \mathbb{R} \mapsto \mathbb{R}$  is a strictly convex and differentiable cost function;  $X$  is the available resource; and  $\underline{x}_i$  is the lower bound of  $x_i$ , i.e., the minimum amount of resource that has to be allocated to the  $i$ th entity.

Given the fact that we are interested in distributed algorithms to solve the problem stated in Equation (4.1), we consider a multi-agent network, where the  $i$ th agent is responsible for managing the resource allocated to the  $i$ th entity. Moreover, we assume that the agents have limited communication capabilities, so they can only share information with their neighbors. This constraint can be represented by a graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$  as it was explained in the previous chapter.

Avoiding the lower-bound constraints (4.1c), Karush–Kuhn–Tucker (KKT) conditions establish that at the optimal solution  $x^* = [x_1^*, \dots, x_n^*]^\top$  of the problem given in Equation (4.1a–4.1b), the marginal costs  $\phi'_i(x_i) = \frac{d\phi_i}{dx_i}$  must be equal, i.e.,  $\phi'_i(x_i^*) = \lambda$ , for all  $i = 1, \dots, n$ , where  $\lambda \in \mathbb{R}$ . Hence, a valid alternative to solve (4.1a–4.1b) is the use of consensus methods. For instance, we can employ the algorithm presented in Section 3.4, which is described as follows:

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} (\phi'_j(x_j) - \phi'_i(x_i)), \text{ for all } i \in \mathcal{V}. \quad (4.2)$$

This algorithm has two main properties: *i)* at equilibrium,  $\phi'_i(x_i^*) = \phi'_j(x_j^*)$  if the nodes  $i$  and  $j$  are connected by a path; *ii)*  $\sum_{i=1}^n x_i^* = \sum_{i=1}^n x_i(0)$ , where  $x_i(0)$  is the initial condition of  $x_i$ . Therefore, if the graph  $\mathcal{G}$  is connected and the initial condition is feasible (i.e.,  $\sum_{i=1}^n x_i(0) = X$ ),  $x$  asymptotically reaches the optimal solution of (4.1a–4.1b) under (4.2). However, the same method cannot be applied to solve (4.1) (the problem that considers lower bounds in the resource allocated to each entity) since some feasibility issues related with the constraints (4.1c) arise.

In the following section, we propose a novel method that extends the algorithm in Equation (4.2) to deal with the individual inequality constraints given in Equation (4.1c). Moreover, we show that this method satisfies some passivity properties. Therefore, its analysis can be done by using the result given in Theorem 3.2.3.

## 4.2 Center–Free Resource Allocation Algorithm

### 4.2.1 Resource Allocation Among a Subset of Nodes in a Graph

First, we consider the following subproblem: let  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$  be a graph comprised by a subset of active nodes  $\mathcal{V}_a$  and a subset of passive nodes  $\mathcal{V}_p$ , such that  $\mathcal{V}_a \cup \mathcal{V}_p = \mathcal{V}$ . A certain amount of resource  $X$  has to be split among those nodes to minimize the cost function  $\phi(x)$  subject to each passive node is allocated with its corresponding lower bound  $\underline{x}_i$ . Mathematically, we formulate this subproblem as follows.

$$\min_x \phi(x) \tag{4.3a}$$

$$\text{subject to } \sum_{i=1}^n x_i = X \tag{4.3b}$$

$$x_i = \underline{x}_i, \text{ for all } i \in \mathcal{V}_p. \tag{4.3c}$$

Feasibility of (4.3) is guaranteed by making the following assumption.

**Assumption 4.2.1.** *At least one node is active, i.e.,  $\mathcal{V}_a \neq \emptyset$ .*

According to KKT conditions, the active nodes have to equalize their marginal costs at the optimal solution. Therefore, a consensus among the active nodes is required to solve (4.3). Nonetheless, classic consensus algorithms, as the one given in Equation (4.2), cannot be used directly. For instance, if all the nodes of  $\mathcal{G}$  apply (4.2) and  $\mathcal{G}$  is connected, the marginal costs of both passive and active nodes are driven to be equal in steady state. This implies that the resource allocated to passive nodes can violate the constraint (4.3c). Besides, if the resource allocated to passive nodes is forced to satisfy (4.3c) by setting  $x_i^* = \underline{x}_i$ , for all  $i \in \mathcal{V}_p$ , there is no guarantee that the new solution satisfies (4.3b). Another alternative, is to apply (4.2) to only active nodes (in this case, the neighborhood of node  $i \in \mathcal{V}_a$  in Equation (4.2) has to be taken as  $\{j \in \mathcal{V}_a : (i, j) \in \mathcal{E}\}$ , and the initial condition must satisfy  $\sum_{i \in \mathcal{V}_a} x_i(0) = X - \sum_{i \in \mathcal{V}_p} \underline{x}_i$ ). However, the subgraph formed by the active nodes is not necessarily connected although  $\mathcal{G}$  is connected.

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

Hence, marginal cost of active nodes are not necessarily equalized at equilibrium, which implies that the obtained solution is sub-optimal. In conclusion, modification of (4.2) to address (4.3) is not trivial. In order to deal with this problem, we propose the following algorithm:

$$\hat{x}_i = \sum_{j \in \mathcal{N}_i} (y_j - y_i), \text{ for all } i \in \mathcal{V} \quad (4.4a)$$

$$\dot{\hat{x}}_i = (x_i - \underline{x}_i) + \sum_{j \in \mathcal{N}_i} (y_j - y_i), \text{ for all } i \in \mathcal{V}_p \quad (4.4b)$$

$$y_i = \begin{cases} \phi'_i(x_i) & \text{if } i \in \mathcal{V}_a \\ \phi'_i(x_i) + \hat{x}_i & \text{if } i \in \mathcal{V}_p. \end{cases} \quad (4.4c)$$

In the same way as in (4.2), the variables  $\{x_i, i \in \mathcal{V}\}$  in Equation (4.4) correspond to the resource allocated to both active and passive nodes. Notice that we have added auxiliary variables  $\{\hat{x}_i, i \in \mathcal{V}_p\}$  that allow the passive nodes to interact with their neighbors taking into account the constraint (4.3c). On the other hand, the term  $\sum_{j \in \mathcal{N}_i} (y_j - y_i)$ , in Equations (4.4a)–(4.4b), leads to a consensus among the elements of the vector  $y = [y_1, \dots, y_n]^\top$ , which are given in Equation (4.4c). For active nodes,  $y_i$  only depends on the marginal cost  $\phi'_i(x_i)$ , while for passive nodes,  $y_i$  depends on both the marginal cost and the state of the auxiliary variable  $\hat{x}_i$ . Therefore, if the  $i$ th node is passive, it has to compute both variables  $x_i$  and  $\hat{x}_i$ . Furthermore, it can be seen that, if all the nodes are active, i.e.,  $(\mathcal{V}_a = \mathcal{V})$ , then the proposed algorithm becomes the one stated in Equation (4.2).

Notice that the  $i$ th node only needs to know  $y_i$  and the values  $\{y_j : j \in \mathcal{N}_i\}$  to compute  $\sum_{j \in \mathcal{N}_i} (y_j - y_i)$  in (4.4a)–(4.4b). In other words,

$$L(\mathcal{G})y = \left[ \sum_{j \in \mathcal{N}_1} (y_j - y_1), \dots, \sum_{j \in \mathcal{N}_n} (y_j - y_n) \right]^\top$$

is a distributed map over the graph  $\mathcal{G}$  (cf., Definition 2.1.1). This implies that the dynamics given in Equation (4.4) can be computed by each node using only local information. In fact, the message that the  $i$ th node must send to its neighbors is solely composed by the variable  $y_i$ .

### 4.2.1.1 Feasibility

Let us prove that, under the multi-agent system proposed in Equation (4.4),  $x(t)$  satisfies the first constraint of the problem given by Equation (4.3), for all  $t \geq 0$ , provided that  $\sum_{i=1}^n x_i(0) = X$ .

**Lemma 4.2.1.** *The quantity  $\sum_{i=1}^n x_i(t)$  is invariant under Equation (4.4), i.e., if  $\sum_{i=1}^n x_i(0) = X$ , then  $\sum_{i=1}^n x_i(t) = X$ , for all  $t \geq 0$ .*

*Proof.* It is sufficient to prove that  $\dot{\Delta} = 0$ , where  $\Delta = \sum_{i=1}^n x_i$ . Notice that  $\dot{\Delta} = \sum_{i=1}^n \dot{x}_i = \mathbf{1}^\top \dot{x}$ , where  $\dot{x} = [\dot{x}_1, \dots, \dot{x}_n]^\top$ . Moreover, according to Equation (4.4),  $\mathbf{1}^\top \dot{x} = -\mathbf{1}^\top L(\mathcal{G})y$ . Since  $\mathcal{G}$  is undirected,  $\mathbf{1}^\top L(\mathcal{G}) = L(\mathcal{G})\mathbf{1} = 0$ . Therefore,  $\dot{\Delta} = 0$ .  $\square$

The above lemma does not guarantee that  $x(t)$  is always feasible because of the second constraint in Equation (4.3), i.e.,  $x_i = \underline{x}_i$ , for all  $i \in \mathcal{V}_p$ . However, it is possible to prove that, at equilibrium, this constraint is properly satisfied.

### 4.2.1.2 Equilibrium Point

The next proposition characterizes the equilibrium point of the multi-agent system given in Equation (4.4).

**Proposition 4.2.1.** *Let Assumption 4.2.1 holds. If  $\mathcal{G}$  is connected, the system in Equation (4.4) has an equilibrium point  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$  that satisfies the following conditions:  $\phi'_i(x_i^*) = \lambda$ , for all  $i \in \mathcal{V}_a$ , where  $\lambda \in \mathbb{R}$  is a constant;  $x_i^* = \underline{x}_i$ , for all  $i \in \mathcal{V}_p$ ; and  $\sum_{i=1}^n x_i^* = X$ . Moreover,  $\hat{x}_i^* = \lambda - \phi'_i(x_i^*)$ , for all  $i \in \mathcal{V}_p$ .*

*Proof.* Since (4.3) is feasible (due to the fact that Assumption 4.2.1 holds), there exists  $x^*$  that minimizes  $\phi(x)$  and satisfies (4.3b)–(4.3c), i.e.,  $\sum_{i=1}^n x_i^* = X$ , and  $x_i^* = \underline{x}_i$ , for all  $i \in \mathcal{V}_p$ . Notice that, applying the KKT first order necessary conditions to (4.3),  $x^*$  must also satisfy  $\phi'_i(x_i^*) = \lambda$ , for all  $i \in \mathcal{V}_a$ , and some  $\lambda \in \mathbb{R}$ . Let  $\hat{x}_i^* = \lambda - \phi'_i(x_i^*)$ , for all  $i \in \mathcal{V}_p$ . To finish the proof, we need to show that the point  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$  defined above is an equilibrium of (4.4).

It follows from Equation (4.4a) that  $y_i^* = \lambda$ , for all  $i \in \mathcal{V}$ , where  $y_i^*$  is the value of  $y_i$  at  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$ . Thus, replacing this value in (4.4a), we have that, at  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$ ,  $\hat{x}_i = 0$ , for all  $i \in \mathcal{V}$ . Besides, replacing the value of  $y_i^*$  in Equation (4.4b), and using the fact that  $x_i^* = \underline{x}_i$ , for all  $i \in \mathcal{V}_p$ , we obtain that  $\hat{x}_i = 0$ , for all  $i \in \mathcal{V}_p$ .  $\square$

## 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

**Remark 4.2.1.** *Proposition 4.2.1 states that the system (4.4) has an equilibrium point  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$  for which the active nodes equalize their marginal costs, while each passive node is allocated with an amount of resource equal to its corresponding lower bound. Moreover,  $x^*$  satisfies the constraints (4.3b)–(4.3c). In conclusion, this proposition guarantees that  $x^*$  minimizes the optimization subproblem given in Equation (4.3). Additionally, notice that the values  $\{\hat{x}_i^*, i \in \mathcal{V}_p\}$  are equal to the KKT multipliers associated with the constraint (4.3c).*

### 4.2.1.3 Convergence Analysis

According to Remark 4.2.1, to show that the algorithm proposed in Equation (4.4) converges to the optimal solution of the subproblem (4.3), it is sufficient to prove that the equilibrium point given in Proposition 4.2.1 is asymptotically stable (AS). Before doing so, let us notice that the algorithm (4.4) is a dynamical system that can be modeled by the feedback interconnection of (4.4a) and (4.4b)–(4.4c). Furthermore, since Equation (4.4a) is equivalent to the model of the multi-agent network (3.4) that applies the resource allocation algorithm (3.6), the convergence analysis can be performed by using the result on passivity stated in Proposition 3.2.2 of the previous chapter.

After these preliminaries, let us prove that the dynamics in Equation (4.4) converge to the optimal solution of (4.3), provided that each  $\phi_i(x_i)$  is strictly convex.

**Proposition 4.2.2.** *Assume that  $\phi_i(x_i)$ , is a strictly convex cost function, for all  $i \in \mathcal{V}$ . If  $\mathcal{G}$  is connected,  $\sum_{i=1}^n x_i(0) = X$ , and Assumption 4.2.1 holds, then  $x(t)$  converges to  $x^*$  under Equation (4.4), where  $x^*$  is the solution of the optimization problem stated in Equation (4.3), i.e.  $x^*$  is the same given in Proposition 4.2.1. Furthermore,  $\hat{x}_i$  converges to  $\hat{x}_i^*$ , for all  $i \in \mathcal{V}_p$ .*

*Proof.* Let us express our multi-agent system in error coordinates with respect to the equilibrium point  $(x^*, \{\hat{x}_i^*, i \in \mathcal{V}_p\})$ , as follows

$$\begin{aligned} \dot{e} &= -L(\mathcal{G})e_y \\ \dot{\hat{e}}_i &= e_i - (L(\mathcal{G})e_y)_i, \text{ for all } i \in \mathcal{V}_p \\ e_{y_i} &= \begin{cases} \phi'_i(x_i) - \phi'_i(x_i^*) & \text{if } i \in \mathcal{V}_a \\ \phi'_i(x_i) - \phi'_i(x_i^*) + \hat{e}_i & \text{if } i \in \mathcal{V}_p, \end{cases} \end{aligned} \quad (4.5)$$

where  $L(\mathcal{G})$  is the graph Laplacian of  $\mathcal{G}$ ;  $e_i = x_i - x_i^*$ , and  $e_{y_i} = y_i - y_i^*$ , for all  $i \in \mathcal{V}$ ;  $\hat{e}_i = \hat{x}_i - \hat{x}_i^*$ , for all  $i \in \mathcal{V}_p$ ;  $e = [e_1, \dots, e_n]^\top$ ;  $e_y = [e_{y_1}, \dots, e_{y_n}]^\top$ ; and  $(L(\mathcal{G})e_y)_i$  represents the  $i$ th element of the vector  $L(\mathcal{G})e_y$ .

## 4.2 Center-Free Resource Allocation Algorithm

Since Assumption 4.2.1 holds,  $\mathcal{V}_a \neq \emptyset$ . Let  $k$  be an active node, i.e.,  $k \in \mathcal{V}_a$ , and let  $e^k, e_y^k$  be the vectors obtained by removing the  $k$ th element from vectors  $e$  and  $e_y$ , respectively. We notice that, according to Lemma 4.2.1,  $e_k(t) = -\sum_{i \in \mathcal{V}, i \neq k} e_i(t)$ , for all  $t \geq 0$ . Therefore, Equation (4.5) can be expressed as

$$\begin{aligned} \dot{e}^k &= -L^k(\mathcal{G})e_y^k - \mathbf{1}_k^{k_r} e_{y_k} \\ e_k &= -\sum_{i \in \mathcal{V}, i \neq k} e_i \\ \dot{\hat{e}}_i &= e_i - \left( L^k(\mathcal{G})e_y^k + \mathbf{1}_k^{k_r} e_{y_k} \right)_i, \text{ for all } i \in \mathcal{V}_p \\ e_{y_i} &= \begin{cases} \phi'_i(x_i) - \phi'_i(x_i^*) & \text{if } i \in \mathcal{V}_a \\ \phi'_i(x_i) - \phi'_i(x_i^*) + \hat{e}_i & \text{if } i \in \mathcal{V}_p, \end{cases} \end{aligned} \quad (4.6)$$

where  $L^k(\mathcal{G})$ , and  $\mathbf{1}_k^{k_r}$  are defined in Lemma 2.1.1. In order to prove that the origin of the above system is AS, let us define the following Lyapunov function:

$$V = \frac{1}{2} e^{k\top} \left( L^k(\mathcal{G}) \right)^{-1} e^k + \frac{1}{2} \sum_{i \in \mathcal{V}_p} (e_i - \hat{e}_i)^2. \quad (4.7)$$

The function  $V$  is positive definite since  $\mathcal{G}$  is connected (the reason of this fact is that, according to Lemma 2.1.1,  $L^k(\mathcal{G})$  and its inverse are positive definite matrices if  $\mathcal{G}$  is connected). Notice that the first term of (4.7) is equivalent to the storage function (3.9) defined in the proof of Proposition 3.2.2. Therefore, using the result stated in that proposition, we obtain that the derivative of  $V$  along the trajectories of the system stated in Equation (4.6) is given by,

$$\dot{V} = -e^\top e_y - \sum_{i \in \mathcal{V}_p} e_i (e_i - \hat{e}_i)$$

Replacing  $e_y$  by the expression given in (4.6), we have that

$$\begin{aligned} \dot{V} &= -\sum_{i=1}^n (e_i (\phi'_i(x_i) - \phi'_i(x_i^*))) - \sum_{i \in \mathcal{V}_p} e_i \hat{e}_i + \sum_{i \in \mathcal{V}_p} e_i (\hat{e}_i - e_i) \\ &= -\sum_{i=1}^n ((x_i - x_i^*) (\phi'_i(x_i) - \phi'_i(x_i^*))) - \sum_{i \in \mathcal{V}_p} e_i^2, \end{aligned}$$

where  $\phi'_i$  is strictly increasing given the fact that  $\phi_i$  is strictly convex, for all  $i \in \mathcal{V}$ . Therefore,  $(x_i - x_i^*) (\phi'_i(x_i) - \phi'_i(x_i^*)) \geq 0$ , for all  $i \in \mathcal{V}$ , and thus  $\dot{V} \leq 0$ .

Since  $\dot{V}$  does not depend on  $\{\hat{e}_i, i \in \mathcal{V}_p\}$ , it is negative semidefinite. Let  $S = \{\{e_i, i \in \mathcal{V}\}, \{\hat{e}_i, i \in \mathcal{V}_p\} : \dot{V} = 0\}$ , i.e.,  $S = \{\{e_i, i \in \mathcal{V}\}, \{\hat{e}_i, i \in \mathcal{V}_p\} : e_i = 0, \text{ for all } i \in \mathcal{V}\}$ . Given the fact that  $\mathcal{G}$  is connected and  $\mathcal{V} \neq \mathcal{V}_p$  (by Assumption 4.2.1), then  $\dot{e} = \mathbf{0}$  iff  $e_y = \mathbf{0}$  (see Equation (4.5)). Therefore, the only solution that stays identically in  $S$  is the trivial solution, i.e.,  $e_i(t) = 0$ , for all  $i \in \mathcal{V}$ ,  $\hat{e}_i(t) = 0$ , for all  $i \in \mathcal{V}_p$ . Hence, we can conclude that the origin is AS by applying the invariance principle given in Theorem 2.2.1.  $\square$

## 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

In summary, we have shown that the algorithm described in Equation (4.4) asymptotically solves the subproblem in Equation (4.3), i.e., (4.4) guarantees that the resource allocated to each passive node is equal to its corresponding lower bound, while the remaining resource  $X - \sum_{i \in \mathcal{V}_p} \underline{x}_i$  is optimally allocated to active nodes.

### 4.2.2 Optimal Resource Allocation with Lower Bounds

Now, let us consider our original problem stated in Equation (4.1), i.e., the resource allocation problem that includes lower bound constraints. Let  $x^* = [x_1^*, \dots, x_n^*]^\top$  be the optimal solution of this problem. Notice that, if we know in advance which nodes will satisfy the constraint (4.1c) with strict equality after making the optimal resource allocation process, i.e.,  $\underline{\mathcal{J}} := \{i \in \mathcal{V} : x_i^* = \underline{x}_i\}$ , we can mark these nodes as passive and reformulate (4.1) as a subproblem of the form (4.3). Based on this idea, we propose a solution method for (4.1), which is divided in two stages: in the first one, the nodes that belong to  $\underline{\mathcal{J}}$  are identified and marked as passive; in the second one, the resulting subproblem of the form (4.3) is solved by using (4.4).

Protocol (4.4) can also be used in the first stage of the method as follows: in order to identify the nodes that will satisfy (4.1c) with strict equality at the optimal allocation, we start marking all nodes as active and apply the resource allocation process given by (4.4). The nodes that are allocated with an amount of resource below their lower bounds at equilibrium are marked as passive, and then (4.4) is newly applied (in this way, passive nodes are forced to meet (4.1c)). This iterative process is performed until all nodes satisfy their lower bound constraints. Notice that the last iteration of this procedure corresponds to solving a subproblem of the form (4.3) where the set of passive nodes is equal to the set  $\underline{\mathcal{J}}$ . Therefore, this last iteration is equivalent to the second stage of the proposed method.

Summarizing, our method relies on an iterative process that uses the continuous-time protocol (4.4) as a subroutine. The main idea of this methodology is to identify in each step the nodes that have an allocated resource out of their lower bounds. These nodes are marked as passive, so they are forced to satisfy their constraints in subsequent iterations, while active nodes seek to equalize their marginal costs using the remaining resource. In the worst case scenario, the classification between active and passive nodes requires  $|\mathcal{V}|$  iterations, where  $|\mathcal{V}|$  is the number of nodes in the network. This fact arises when only one active node becomes passive at each iteration.



---

## 4.2 Center-Free Resource Allocation Algorithm

---

The proposed method is formally described in Algorithm 1. Notice that this algorithm is fully decentralized since steps 4–6 can be computed by each agent using only local information. Step 4 corresponds to solve Equation (4.4), while Steps 5 and 6 describe the conditions for converting an active node into passive. Let us note that steps 4–6 have to be performed  $|\mathcal{V}|$  times since we are considering the worst case scenario. Therefore, each agent needs to know the total number of nodes in the network. This requirement can be computed in a distributed way by using the method proposed in ([103], p.90). We also notice the fact that the agents have to be synchronized (as usual in several distributed algorithms [27, 35, 103]) in order to apply the step 4 of Algorithm 1, i.e., all agents must start solving Equation (4.4) at the same time.

---

**Algorithm 1:** Resource allocation with lower bounds

---

**Input:** – Parameters of the problem in Equation (4.1).  
– An initial value  $x^{(0)}$ , such that  $\sum_{i=1}^n x_i^{(0)} = X$ .

**Output:** Optimal allocation  $\tilde{x}^*$

- 1 Mark all nodes as active, i.e.,  $\tilde{\mathcal{V}}_{a,0} \leftarrow \mathcal{V}$ ,  $\tilde{\mathcal{V}}_{p,0} \leftarrow \emptyset$ .
- 2  $\tilde{x}_{i,0} \leftarrow x_i^{(0)}$ , for all  $i \in \mathcal{V}$ .
- 3 **for**  $l \leftarrow 1$  **to**  $|\mathcal{V}|$  **do**
- 4      $\tilde{x}_{i,l} \leftarrow x_i(t_l)$ , for all  $i \in \mathcal{V}$ , where  $x_i(t_l)$  is the solution of Equation (4.4a) at time  $t_l$ , with initial conditions  $x(0) = [\tilde{x}_{1,l-1}, \dots, \tilde{x}_{n,l-1}]^\top$ ,  $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,l-1}$ ,  $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,l-1}$ , and  $\{\hat{x}_i(0) = 0, \forall i \in \mathcal{V}_p\}$ .
- 5      $\tilde{\mathcal{V}}_{p,l} \leftarrow \tilde{\mathcal{V}}_{p,l-1} \cup \{i \in \tilde{\mathcal{V}}_{a,l-1} : \tilde{x}_i < \underline{x}_i\}$ , and  $\tilde{\mathcal{V}}_{a,l} \leftarrow \tilde{\mathcal{V}}_{a,l-1} \setminus \{i \in \tilde{\mathcal{V}}_{a,l-1} : \tilde{x}_i < \underline{x}_i\}$ .
- 6      $\tilde{x}^* \leftarrow [\tilde{x}_{1,l}, \dots, \tilde{x}_{n,l}]^\top$ .
- 7 **return**  $\tilde{x}^*$

---

According to the reasoning described at the beginning of this subsection, we ideally require to know the steady state solution of Equation (4.4) at each iteration of Algorithm 1 (since we need to identify which nodes are allocated with an amount of resource below their lower bounds in steady state). This implies that the time  $t_l$  in step 4 of Algorithm 1 goes to infinity. Under this requirement, each iteration would demand infinite time and the algorithm would not be implementable. Hence, to relax the infinite time condition, we state the following assumption on the time  $t_l$ .

**Assumption 4.2.2.** Let  $x_{i,l}^*$  be the steady state of  $x_i(t)$  under Equation (4.4), with

## 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

initial conditions  $x(0) = \tilde{x}_{i,l-1}$ ,  $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,l-1}$ ,  $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,l-1}$ , and  $\{\hat{x}_i(0) = 0, \forall i \in \mathcal{V}_p\}^1$ . For each  $l = 1, \dots, |\mathcal{V}| - 1$ , the time  $t_l$  satisfies the following condition:  $x_i(t_l) < \underline{x}_i$  if and only if  $x_{i,l}^* < \underline{x}_i$ , for all  $i \in \mathcal{V}$ .

According to assumption 4.2.2, for the first  $|\mathcal{V}| - 1$  iterations, we only need a solution of (4.4) that is close enough to the steady state solution. We point out the fact that, if the conditions of Proposition 4.2.2 are met in the  $l$ th iteration of Algorithm 1, then  $x_i(t)$  asymptotically converges to  $x_{i,l}^*$ , for all  $i \in \mathcal{V}$ , under Equation (4.4). Therefore, Assumption 4.2.2 is satisfied for large values of  $t_1, \dots, t_{|\mathcal{V}|-1}$ .

Taking into account all the previous considerations, the next theorem states our main result regarding the optimality of the output of Algorithm 1.

**Theorem 4.2.1.** *Assume that  $\mathcal{G}$  is a connected graph. Moreover, assume that  $\phi_i$  is a strictly convex function for all  $i = 1, \dots, n$ . If  $t_1, \dots, t_{|\mathcal{V}|-1}$  satisfy Assumption 4.2.2, and the problem stated in Equation (4.1) is feasible, then the output of Algorithm 4.1 tends to the optimal solution of the problem given in Equation (4.1) as  $t_{|\mathcal{V}|} \rightarrow \infty$ .*

*Proof.* The  $i$ th component of the output of Algorithm 1 is equal to  $\tilde{x}_{i,|\mathcal{V}|} = x_i(t_{|\mathcal{V}|})$ , where  $x_i(t_{|\mathcal{V}|})$  is the solution of Equation (4.4a) at time  $t_{|\mathcal{V}|}$ , with initial conditions  $[\tilde{x}_{1,|\mathcal{V}|-1}, \dots, \tilde{x}_{n,|\mathcal{V}|-1}]^\top$ ,  $\mathcal{V}_a = \tilde{\mathcal{V}}_{a,|\mathcal{V}|}$ , and  $\mathcal{V}_p = \tilde{\mathcal{V}}_{p,|\mathcal{V}|}$ . Hence, it is sufficient to prove that  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  solves the problem in Equation (4.1). In order to do that, let us consider the following premises (the proof of each premise is written in square brackets).

- **P1:**  $\{\tilde{x}_{1,l}, \dots, \tilde{x}_{n,l}\}$  satisfies (4.1b), for all  $l = 1, \dots, |\mathcal{V}|$  [this follows from Lemma 4.2.1, and from the fact that  $\sum_{i=1}^n \tilde{x}_{i,0} = X$ ].
- **P2:**  $x_{i,l}^* = \underline{x}_i$ , for all  $i \in \tilde{\mathcal{V}}_{p,l-1}$ , and for all  $l = 1, \dots, |\mathcal{V}|$  [this follows directly from Proposition 4.2.2].
- **P3:**  $\tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{p,l-1} \cup \{i \in \tilde{\mathcal{V}}_{a,l-1} : x_{i,l}^* < \underline{x}_i\}$ , and  $\tilde{\mathcal{V}}_{a,l} = \tilde{\mathcal{V}}_{a,l-1} \setminus \{i \in \tilde{\mathcal{V}}_{a,l-1} : x_{i,l}^* < \underline{x}_i\}$ , for all  $l = 1, \dots, |\mathcal{V}|$  [this follows from step 5 of Algorithm 1, and from Assumption 4.2.2].
- **P4:** If for some  $l$ ,  $\tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{p,l-1}$ , then  $\tilde{\mathcal{V}}_{p,l+j} = \tilde{\mathcal{V}}_{p,l-1}$ , for all  $j = 0, \dots, |\mathcal{V}| - l$  [this can be seen from the fact that if the set of passive nodes does not change from one iteration to the next, the steady state of Equation (4.4a) is the same for both iterations].

---

<sup>1</sup>As well as in step 4 of Algorithm 1, we have initialized the auxiliary variables  $\hat{x}_i$  to zero by convention. If these variables are initialized to other value, convergence of (4.4) is not affected (cf. Proposition 4.2.2).

## 4.2 Center-Free Resource Allocation Algorithm

- **P5:**  $\tilde{\mathcal{V}}_{a,l} \cup \tilde{\mathcal{V}}_{p,l} = \mathcal{V}$ , for all  $l = 1, \dots, |\mathcal{V}|$  [from P3, we know that  $\tilde{\mathcal{V}}_{a,l} \cup \tilde{\mathcal{V}}_{p,l} = \tilde{\mathcal{V}}_{a,l-1} \cup \tilde{\mathcal{V}}_{p,l-1}$ , for all  $l = 1, \dots, |\mathcal{V}|$ . Moreover, given the fact that  $\tilde{\mathcal{V}}_{p,0} = \emptyset$ , and  $\tilde{\mathcal{V}}_{a,0} = \mathcal{V}$ , (see step 1 of Algorithm 1) we can conclude P5].
- **P6:** Since the problem in Equation (4.1) is feasible by assumption, then  $|\tilde{\mathcal{V}}_{p,l}| < |\mathcal{V}|$ , for all  $l = 1, \dots, |\mathcal{V}|$  [the fact that  $|\tilde{\mathcal{V}}_{p,l}| \leq |\mathcal{V}|$ , for all  $l = 1, \dots, \mathcal{V}$ , follows directly from P5. Let us prove that  $|\tilde{\mathcal{V}}_{p,l}| \neq |\mathcal{V}|$ , for all  $l = 1, \dots, \mathcal{V}$ , by contradiction: assume that there exists some  $l$ , such that  $|\tilde{\mathcal{V}}_{p,l-1}| < |\mathcal{V}|$  and  $|\tilde{\mathcal{V}}_{p,l}| = |\mathcal{V}|$ . Hence, from P2 and P3, we know that  $x_{i,l}^* \leq \underline{x}_i$ , for all  $i \in \mathcal{V}$ ; moreover,  $\{i \in \tilde{\mathcal{V}}_{a,l-1} : x_{i,l}^* < \underline{x}_i\} \neq \emptyset$ . Therefore,  $\sum_{i=1}^n x_{i,l}^* < \sum_{i=1}^n \underline{x}_i$ . According to P1, we know that  $\sum_{i=1}^n x_{i,l}^* = X$ ; thus,  $X < \sum_{i=1}^n \underline{x}_i$ , which contradicts the feasibility assumption].
- **P7:**  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  satisfies the constraints (4.1c) [in order to prove P7, we proceed by contradiction: assume that  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  does not satisfy the constraints (4.1c). Since P2 holds, this assumption implies that  $\{i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}-1|} : x_{i,|\mathcal{V}|}^* < \underline{x}_i\} \neq \emptyset$ . Therefore,  $\tilde{\mathcal{V}}_{p,|\mathcal{V}|} \neq \tilde{\mathcal{V}}_{p,|\mathcal{V}-1|}$  (see P3). Using P4, we can conclude that  $\tilde{\mathcal{V}}_{p,|\mathcal{V}|} \neq \tilde{\mathcal{V}}_{p,|\mathcal{V}-1|} \neq \dots \neq \tilde{\mathcal{V}}_{p,0} = \emptyset$ , i.e.,  $\{i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}-j|} : x_{i,|\mathcal{V}-j+1}^* < \underline{x}_i\} \neq \emptyset$ , for all  $j = 1, \dots, |\mathcal{V}|$ . Thus, according to P3,  $|\tilde{\mathcal{V}}_{p,|\mathcal{V}|}| > |\tilde{\mathcal{V}}_{p,|\mathcal{V}-1|}| > \dots > |\tilde{\mathcal{V}}_{p,1}| > 0$ . Hence,  $|\tilde{\mathcal{V}}_{p,|\mathcal{V}|}| \geq |\mathcal{V}|$ , which contradicts P6].
- **P8:**  $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l}^* \geq \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l+1}^*$  [we prove P8 as follows: using P1 and the result in Lemma 4.2.1, we know that  $\sum_{i \in \mathcal{V}} x_{i,l}^* = \sum_{i \in \mathcal{V}} x_{i,l+1}^* = X$ . Moreover, according to P5,  $\mathcal{V}$  can be expressed as  $\mathcal{V} = \tilde{\mathcal{V}}_{a,l} \cup \tilde{\mathcal{V}}_{p,l}$ , where  $\tilde{\mathcal{V}}_{p,l} = (\tilde{\mathcal{V}}_{p,l} \setminus \tilde{\mathcal{V}}_{p,l-1}) \cup \tilde{\mathcal{V}}_{p,l-1}$  (see P3). Thus, we have that  $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}} x_{i,l}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l-1}} x_{i,l}^* = \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l+1}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}} x_{i,l+1}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l-1}} x_{i,l+1}^*$ . Furthermore, since  $\tilde{\mathcal{V}}_{p,l-1} \subset \tilde{\mathcal{V}}_{p,l}$ , and given the fact that P2 holds, we have that  $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}} x_{i,l}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i = \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l+1}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i + \sum_{i \in \tilde{\mathcal{V}}_{p,l-1}} \underline{x}_i$ . Thus,  $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l}^* = \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l+1}^* + \sum_{i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}} (\underline{x}_i - x_{i,l}^*)$ , where  $\underline{x}_i - x_{i,l}^* > 0$ , for all  $i \in \tilde{\mathcal{V}}_{p,l}, i \notin \tilde{\mathcal{V}}_{p,l-1}$  (according to P3). Hence, we can conclude P8].
- **P9:** There exists  $k$ , such that  $k \in \tilde{\mathcal{V}}_{a,l}$ , for all  $l = 1, \dots, |\mathcal{V}|$  [in order to prove P9, we use the fact that, if  $k \in \tilde{\mathcal{V}}_{a,l}$ , then  $k \in \tilde{\mathcal{V}}_{a,l-j}$ , for all  $j = 1, \dots, l$  (this follows from P3). Moreover, according to P5 and P6,  $|\tilde{\mathcal{V}}_{a,|\mathcal{V}|}| \neq 0$ ; hence, there exists  $k$ , such that  $k \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|}$ . Therefore, P9]. P9 guarantees that Assumption 4.2.1 holds at each iteration.
- **P10:**  $\phi'_i(x_{i,l}^*) \geq \phi'_i(x_{i,l+1}^*)$ , for all  $i \in \tilde{\mathcal{V}}_{a,l}$  [we prove P10 by contradiction: assume that  $\phi'_i(x_{i,l}^*) < \phi'_i(x_{i,l+1}^*)$ , for some  $i \in \tilde{\mathcal{V}}_{a,l}$ . According to Proposition 4.2.2, and

## 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

since P1 and P9 hold,  $x_{i,l}^*$  has the characteristics given in Proposition 4.2.1, for all  $i \in \mathcal{V}$ , and for all  $l = 1, \dots, |\mathcal{V}|$ . Hence,  $\phi'_i(x_{i,l}^*)$  has the same value for all  $i \in \tilde{\mathcal{V}}_{a,l-1}$ , and  $\phi'_i(x_{i,l+1}^*)$  has the same value for all  $i \in \tilde{\mathcal{V}}_{a,l}$ . Moreover, since  $\tilde{\mathcal{V}}_{a,l} \subset \tilde{\mathcal{V}}_{a,l-1}$  (according to P3), we have that  $\phi'_i(x_{i,l}^*) < \phi'_i(x_{i,l+1}^*)$ , for all  $i \in \tilde{\mathcal{V}}_{a,l}$ . Thus,  $x_{i,l}^* < x_{i,l+1}^*$ , for all  $i \in \mathcal{V}_a, l$ , because  $\phi'_i$  is strictly increasing (this follows from the fact that  $\phi_i$  is strictly convex by assumption). Therefore,  $\sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l}^* < \sum_{i \in \tilde{\mathcal{V}}_{a,l}} x_{i,l+1}^*$ , which contradicts P8].

Now, let us prove that  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  solves the Problem in Equation (4.1). First, the solution  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  is feasible according to P1 and P7. On the other hand, from P9, it is known that  $\exists k : k \in \tilde{\mathcal{V}}_{a,l}$ , for all  $l = 1, \dots, |\mathcal{V}|$ . Let  $\phi'_k(x_{k,|\mathcal{V}|}^*) = \lambda$ , where  $\lambda \in \mathbb{R}$ . Moreover, let us define  $\mathcal{V}^0 = \{j \in \mathcal{V} : x_{j,|\mathcal{V}|}^* > \underline{x}_j\}$ , and  $\mathcal{V}^1 = \{j \in \mathcal{V} : x_{j,|\mathcal{V}|}^* \leq \underline{x}_j\}$ .

If  $i \in \mathcal{V}^0$ , then  $i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|-1}$  (given the fact that, if  $i \notin \tilde{\mathcal{V}}_{a,|\mathcal{V}|-1} \Rightarrow i \in \tilde{\mathcal{V}}_{p,|\mathcal{V}|-1} \Rightarrow x_{i,|\mathcal{V}|}^* = \underline{x}_i \Rightarrow i \notin \mathcal{V}^0$ ). Hence,  $\phi'_i(x_{i,|\mathcal{V}|}^*) = \phi'_k(x_{k,|\mathcal{V}|}^*) = \lambda$  (this follows from the fact that  $\phi'_j(x_{j,l}^*)$  has the same value for all  $j \in \tilde{\mathcal{V}}_{a,l-1}$ , which in turn follows directly from step 4 of Algorithm 1, and Proposition 4.2.2).

If  $i \in \mathcal{V}^1$ , then either  $i \in \tilde{\mathcal{V}}_{a,|\mathcal{V}|-1}$  or  $i \in \tilde{\mathcal{V}}_{p,|\mathcal{V}|-1}$ . In the first case,  $\phi'_i(x_{i,|\mathcal{V}|}^*) = \phi'_k(x_{k,|\mathcal{V}|}^*) = \lambda$  (following the reasoning used when  $i \in \mathcal{V}^0$ ). In the second case,  $\exists l : i \in (\tilde{\mathcal{V}}_{p,l} \setminus \tilde{\mathcal{V}}_{p,l-1})$ ; hence,  $\phi'_i(x_{i,l}^*) = \phi'_k(x_{k,l}^*)$  (this follows from the fact that, if  $i \in (\tilde{\mathcal{V}}_{p,l} \setminus \tilde{\mathcal{V}}_{p,l-1})$ , then  $i \in \tilde{\mathcal{V}}_{a,l-1}$ ). Furthermore, since  $i \in (\tilde{\mathcal{V}}_{p,l} \setminus \tilde{\mathcal{V}}_{p,l-1})$ ,  $x_{i,l}^* < \underline{x}_i$  (see P3), and given the fact that  $\phi_i$  is strictly increasing, we have that  $\phi'_i(x_{i,l}^*) < \phi'_i(\underline{x}_i)$ . Moreover, according to P10,  $\phi'_k(x_{k,l}^*) \geq \phi'_k(x_{k,|\mathcal{V}|}^*)$ . Hence,  $\phi'_i(\underline{x}_i) > \phi'_k(x_{k,|\mathcal{V}|}^*) = \lambda$ . In conclusion, if  $i \in \mathcal{V}^1$ , then  $\phi'_i(x_{i,|\mathcal{V}|}^*) \geq \lambda$ .

Thus, we can choose  $\mu_i \geq 0$ , for all  $i \in \mathcal{V}$ , such that  $\phi'_i(x_{i,|\mathcal{V}|}^*) - \mu_i = \lambda$ , where  $\mu_i = 0$  if  $i \in \mathcal{V}^0$ . Hence, let us note that,  $\left. \frac{\partial \phi}{\partial x_i} \right|_{x_i=x_{i,|\mathcal{V}|}^*} - \mu_i - \lambda = 0$ , for all  $i \in \mathcal{V}$ , where  $\left. \frac{\partial \phi}{\partial x_i} \right|_{x_i=x_{i,|\mathcal{V}|}^*} = \phi'_i(x_{i,|\mathcal{V}|}^*)$ . Therefore,  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*, \mu_1, \dots, \mu_n, -\lambda\}$  satisfies the KKT conditions for the problem given in Equation (4.1). Furthermore, since  $\phi(x)$  is a strictly convex function by assumption, then  $\{x_{1,|\mathcal{V}|}^*, \dots, x_{n,|\mathcal{V}|}^*\}$  is the optimal solution to that problem. □

### Early Stopping Criterion

Notice that, if the set of passive nodes does not change in the  $k$ th iteration of Algorithm 1 because all active nodes satisfy the lower bound constraints (see step 5), then the steady state solutions  $x_{i,k}^*$  and  $x_{i,k+1}^*$  are the same, for all  $i \in \mathcal{V}$ , which implies that the

set of passive nodes also does not change in the  $(k + 1)$ th iteration. Following the same reasoning, we can conclude that  $x_{i,k}^* = x_{i,k+1}^* = \dots = x_{i,|\mathcal{V}|}^*$ , for all  $i \in \mathcal{V}$ . Therefore, in this case,  $\{x_{1,k}^*, \dots, x_{n,k}^*\}$  is the solution of our resource allocation problem. Practically speaking, this implies that Algorithm 1 does not need to perform more iterations after the  $k$ th one. Thus, it is possible to implement a flag  $z_i^*$  (in a distributed way) that alerts the agents if all active nodes satisfy the lower bound constraints after step 4 of Algorithm 1. A way to do that is by applying a *min-consensus* protocol [27] with initial conditions  $z_i(0) = 0$  if the node  $i$  is active and does not satisfy its lower bound constraint, and  $z_i(0) = 1$  otherwise. Hence, notice that our flag  $z_i^*$  (i.e., the result of the *min-consensus* protocol) is equal to one, for all  $i \in \mathcal{V}$ , only if all the active nodes satisfy the lower bound constraints, which corresponds to the early stopping criterion described above.

### 4.3 Simulation Results and Comparison

In this section, we compare the performance of our algorithm with other continuous-time distributed techniques found in the literature. We have selected three techniques that are capable to address nonlinear problems and can handle lower bound constraints: *i*) a distributed interior point method [35], *ii*) the local replicator equation [22], *iii*) a distributed interior point method with exact barrier functions [36]. The first one is a traditional methodology that uses barrier functions; the second one is a novel technique based on population dynamics; and the third one is a recently proposed method that follows the same ideas as the first one, but replaces classic logarithmic barrier functions by exact penalty functions. Below, we briefly describe the aforementioned algorithms.

#### 4.3.1 Distributed Interior Point (DIP) Method

This algorithm is a variation of the one presented in Equation (4.2) that includes strictly convex barrier functions to prevent the solution to flow outside the feasible region (cf., Section 3.4.1). The barrier functions  $b_i(x_i)$  are added to the original cost function as follows,

$$\begin{aligned} \phi_b(x) &= \phi(x) + \epsilon \sum_{i=1}^n b_i(x_i) \\ b_i(x_i) &= -\ln(x_i - \underline{x}_i), \quad \text{for all } i \in \mathcal{V}, \end{aligned}$$

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

where,  $\phi_b(x)$  is the new cost function, and  $\epsilon > 0$  is a constant that minimizes the effect of the barrier function when the solution is far from the boundary of the feasible set. With this modification, the distributed algorithm is described by the following equation,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \left( \phi'_{b_j}(x_j) - \phi'_{b_i}(x_i) \right), \text{ for all } i \in \mathcal{V}, \quad (4.8)$$

where,  $\phi'_{b_i}(x_i) = \frac{d\phi_i}{dx_i} - \epsilon \frac{db_i}{dx_i}$ , i.e.,  $\phi'_{b_i}(x_i)$  is equal to the marginal cost plus a penalty term induced by the derivative of the corresponding barrier function.

##### 4.3.2 Distributed Replicator Dynamics (DRD)

This methodology is based on the classical replicator dynamics from evolutionary game theory. In the DRD, the growth rate of a population that plays a certain strategy only depends on its own fitness function and on the fitness of its neighbors. Mathematically, the DRD is given by

$$\begin{aligned} \dot{x}_i &= \sum_{j \in \mathcal{N}_i} (x_i - \underline{x}_i)(x_j - \underline{x}_j)(v_i(x_i) - v_j(x_j)), \\ v_i &= -\phi'_i(x_i), \text{ for all } i \in \mathcal{V}, \end{aligned} \quad (4.9)$$

where  $v_i$  is the fitness perceived by the individuals that play the  $i$ th strategy. In this case, the strategies correspond to the nodes of the network, and the fitness functions to the negative marginal costs (the minus appears because replicator dynamics are used to maximize utilities instead of minimize costs). On the other hand, it can be shown that, if the initial condition  $x(0)$  is feasible for the problem given in Equation (4.1), then  $x(t)$  remains feasible for all  $t \geq 0$ , under the DRD.

##### 4.3.3 Distributed Interior Point Method with Exact Barrier Functions (DIPe)

This technique follows the same reasoning of the DIP algorithm. The difference is that DIPe uses exact barrier functions [104] to guarantee satisfaction of the lower bound constraints. The exact barrier function for the  $i$ th node is given by:

$$b_i^\epsilon(x_i) = \frac{1}{\epsilon} [\underline{x}_i - x_i]_+,$$

### 4.3 Simulation Results and Comparison

where  $[\cdot]_+ = \max(\cdot, 0)$ ,  $0 < \varepsilon < \frac{1}{2 \max_{x \in \mathcal{F}} \|\nabla \phi(x)\|_\infty}$ , and  $\mathcal{F} = \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, x_i \geq \underline{x}_i\}$  is the feasible region of  $x$  for the problem (4.1). Using these exact barrier functions, the augmented cost function can be expressed as,

$$\phi_b^\varepsilon(x) = \phi(x) + \sum_{i=1}^n b_i^\varepsilon(x_i).$$

The DIPe algorithm is given in terms of the augmented cost function and its generalized gradient  $\partial \phi_b^\varepsilon(x) = [\partial_1 \phi_b^\varepsilon(x), \dots, \partial_n \phi_b^\varepsilon(x)]^\top$  as follows:

$$\dot{x}_i \in \sum_{j \in \mathcal{N}_i} (\partial_j \phi_b^\varepsilon(x) - \partial_i \phi_b^\varepsilon(x)), \text{ for all } i \in \mathcal{V}, \quad (4.10)$$

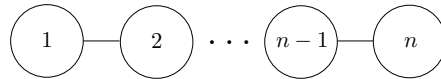
where

$$\partial_i \phi_b^\varepsilon(x) = \begin{cases} \{\phi'_i(x_i) - \frac{1}{\varepsilon}\} & \text{if } x_i < \underline{x}_i \\ [\phi'_i(x_i) - \frac{1}{\varepsilon}, \phi'_i(x_i)] & \text{if } x_i = \underline{x}_i \\ \{\phi'_i(x_i)\} & \text{if } x_i > \underline{x}_i \end{cases}$$

In [36], the authors show that the differential inclusion (4.10) converges to the optimal solution of the problem (4.1) provided that  $x(0)$  is feasible.

#### 4.3.4 Comparison

In order to compare the performance of our algorithm with the three methods described above, we use the following simulation scenario: a set of  $n$  nodes connected as in Figure 4.1 (we use this topology to verify the behavior of the different algorithms in the face of few communication channels since previous studies have shown that algorithms' performance decreases with the number of available communication links); a nonlinear cost function  $\phi(x) = \sum_{i=1}^n e^{a_i(x_i - b_i)} + e^{-a_i(x_i - b_i)}$ , where  $a_i$  and  $b_i$  are random numbers that belong to the intervals  $(1, 2)$  and  $(-\frac{1}{2}, \frac{1}{2})$ , respectively; a resource constraint  $X = 1$ ; and a set of lower bounds  $\{\underline{x}_i = 0 : i \in \mathcal{V}\}$ . For each  $n$ , we generate 50 problems with



**Figure 4.1:** Single path topology for  $n$  nodes.

the characteristics described above. The four distributed methods are implemented in Matlab employing the solver function ode23s. Moreover, we use the solution provided by a centralized technique as reference. The results on the average percentage decrease in

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

the cost function reached with each algorithm and the average computation time (time taken by each algorithm for solving a problem<sup>1</sup>) are summarized in Table 4.1. Results of DIPE for 100 and 200 nodes were not computed for practicality since the time required by this algorithm to solve a 100/200-nodes problem is very high.

We notice that the algorithm proposed in this chapter always reaches the maximum reduction, regardless of the number of nodes that comprise the network. The same happens with the DIPE algorithm. This is an important advantage of our method compared to other techniques. In contrast, the algorithm based on the DRD performs far from the optimal solution. This unsatisfactory behavior is due to the small number of links of the considered communication network. In [22], the authors prove the optimality of the DRD in problems involving well connected networks; however, they also argue that this technique can converge to suboptimal solutions in other cases. On the other hand, the DIP method provides solutions close to the optimum. Nonetheless, its performance decreases when the number of nodes increases. This tendency is due to the influence of barrier functions on the original problem. Notice that, the larger the number of nodes, the bigger the effect of the barrier functions in Equation (4.8).

Regarding the computation time, although convergence of the proposed method is slower than the one shown by DRD and DIP, it is faster than the convergence of the method based on exact barrier functions, i.e., DIPE. Therefore, among the methods that guarantee optimality of the solution, our technique shows the best convergence speed. Computation time taken by DIPE is affected by the use of penalty terms that generate strong changes in the value of the cost function near to the boundaries of the feasible set. The drastic variations of the generalized gradient of exact barrier functions produces oscillations of numerical solvers around the lower bounds (a visual inspection of the results given in Figure 3 of [36] confirms this claim). These oscillations are the main responsible for the low convergence speed shown by DIPE. On the other hand, DRD and DIP exhibit the fastest convergence. Hence, DRD and DIP are appealing to be implemented in applications that require fast computation and tolerate suboptimal solutions.

---

<sup>1</sup>Algorithms were implemented in a computer with an Intel Core i5 processor.



Table 4.1: Distributed Algorithms' Performance

Number of Nodes	Percentage Decrease, Computation Time			
	Proposed Approach	DIP	DRD	DIPe
$n = 5$	100%, 0.08 s.	100%, 0.04 s.	91%, 0.04 s.	100%, 7 s.
$n = 20$	100%, 0.7 s.	99%, 0.4 s.	69%, 0.2 s.	100%, 153 s.
$n = 50$	100%, 3.2 s.	98%, 1.6 s.	57%, 1.3 s.	100%, 841 s.
$n = 100$	100%, 17.8 s.	96%, 9.1 s.	50%, 6.5 s.	–
$n = 200$	100%, 181.2 s.	94%, 68.7 s.	46%, 41.6 s.	–

### 4.4 Application to Optimal Chiller Loading

In this section, we present a distributed solution to the optimal chiller loading problem in multiple chiller systems [2], by using the approach developed in this chapter. This problem arises in decoupled chilled–water plants, which are widely used in large air–conditioning systems. The goal is to distribute the cooling load among the chillers that comprise the plant for minimizing the total amount of power used by them. For a better understanding of the problem, below we present a brief description of the system.

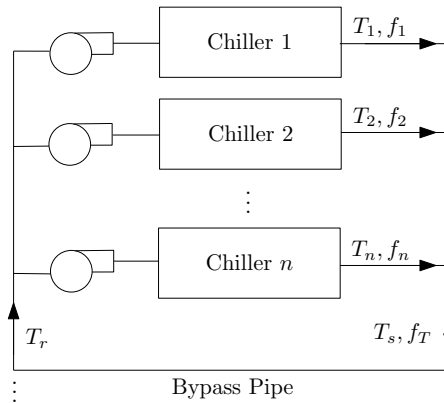


Figure 4.2: Decoupled chilled–water plant with  $n$  chillers (adapted from [2]).

A decoupled chilled–water plant comprised by  $n$  chillers is depicted in Figure 4.2. The purpose of this plant is to provide a water flow  $f_T$  at a certain temperature  $T_s$  to the rest of the air–conditioning system. In order to do that, the plant needs to meet a

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

cooling load  $C_L$  that is given by the following expression

$$C_L = mf_T(T_r - T_s), \quad (4.11)$$

where  $m > 0$  is the specific heat of the water, and  $T_r$  is the temperature of the water returning to the chillers. Since there are multiple chillers, the total cooling load  $C_L$  is split among them, i.e.,  $C_L = \sum_{i=1}^n Q_i$ , where  $Q_i$  is the cooling power provided by the  $i$ th chiller, which, in turn, is given by

$$Q_i = mf_i(T_r - T_i), \quad (4.12)$$

where  $f_i > 0$  and  $T_i$  are, respectively, the flow rate of chilled water and the water supply temperature of the  $i$ th chiller. As it is shown in Figure 4.2, we have that  $f_T = \sum_{i=1}^n f_i$ . In order to meet the corresponding cooling load, the  $i$ th chiller consumes a power  $P_i$  that can be calculated using the following expression [2]

$$P_i = (k_{0,i} + k_{1,i}mf_iT_r + k_{2,i}(mf_iT_r)^2) + (k_{3,i} - k_{1,i}mf_i - k_{4,i}mf_iT_r - 2k_{2,i}(mf_i)^2T_r)T_i + (k_{5,i} + k_{6,i}mf_i + k_{2,i}(mf_i)^2)T_i^2, \quad (4.13)$$

where  $k_{j,i}$ , for  $j = 0, \dots, 6$ , are constants related to the  $i$ th chiller. If we assume that the flow rate  $f_i$  of each chiller is constant, then  $P_i$  is a quadratic function of the temperature  $T_i$ . The optimal chiller loading problem involves the calculation of the chillers' water supply temperatures that meet the total cooling load given in Equation (4.11), and minimize the total amount of power consumed by the chillers, i.e.,  $\sum_{i=1}^n P_i$ . Moreover, given the fact that each chiller has a maximum cooling capacity, we have to consider the following additional constraints

$$mf_i(T_r - T_i) \leq \bar{Q}_i \quad \text{for } i = 1, \dots, n, \quad (4.14)$$

where  $\bar{Q}_i$  is the maximum capacity (rated value) of the  $i$ th chiller.

Summarizing, the optimal chiller loading problem can be expressed as follows

$$\begin{aligned} \min_{T_1, \dots, T_n} \quad & J = \sum_{i=1}^n P_i(T_i) \\ \text{s.t.} \quad & \sum_{i=1}^n mf_i(T_r - T_i) = C_L \\ & T_i \geq T_r - \frac{\bar{Q}_i}{mf_i}, \text{ for all } i = 1, \dots, n. \end{aligned} \quad (4.15)$$

Now, let us consider that we want to solve the aforementioned problem in a distributed way by using a multi-agent system, in which each chiller is managed by an

agent that decides the value of the water supply temperature. We assume that the  $i$ th agent knows (e.g., by measurements) the temperature of the water returning to the chillers, i.e.,  $T_r$ , and the flow rate of chilled water, i.e.,  $f_i$ . Moreover, agents can share their own information with their neighbors through a communication network with a topology given by the graph  $\mathcal{G}$ . If each  $P_i(T_i)$  is a convex function, then the problem can be solved by using the method proposed in Algorithm 1 (we take, in this case,  $x_i = f_i T_i$ ). The main advantage of this approach is to increase the resilience of the whole system in the face of possible failures, due to the fact that the plant operation does not rely on a single control center but on multiple individual controllers without the need for a centralized coordinator.

### Illustrative Example

We simulate a chilled–water plant comprised by 7 chillers<sup>1</sup>, the cooling capacity and the water flow rate of each chiller are, respectively,  $\bar{Q}_i = 1406.8$  kW, and  $f_i = 65$  kg.s<sup>-1</sup>, for  $i = 1, \dots, 7$ ; the specific heat of the water is  $m = 4.19$  kW.s.kg<sup>-1</sup>.C<sup>-1</sup>; the supply temperature of the system is  $T_s = 11$  C; and the coefficients  $k_{j,i}$  of Equation (4.13) are given in Table 4.2. We operate the system at two different cooling loads, the first one is 90% of the total capacity, i.e.,  $C_L = 0.9 \sum_{i=1}^n \bar{Q}_i$ , and the second one is 60% of the total capacity, i.e.,  $C_L = 0.6 \sum_{i=1}^n \bar{Q}_i$ . The  $P_i$ - $T_i$  curves are shown in Figure 4.3a for both cases, it can be noticed that all functions are convex. In order to apply the distributed solution presented in Algorithm 1, we use an agent per chiller (i.e., the  $i$ th agent controls the supply temperature  $T_i$  of the  $i$ th chiller) and the communication network shown in Figure 4.1. In all cases the initial conditions of the chillers' supply temperatures are  $T_i(0) = T_s$ , for  $i = 1, \dots, 7$ . The results for the first cooling load, i.e.,  $C_L = 8862.8$  kW, are depicted in Figure 4.3b, where it is shown that the cooling load is properly allocated among the chillers by adjusting the supply temperatures. More efficient chillers (i.e., chiller 3, chiller 6, and chiller 7 in Figure 4.3a) are more loaded than the less efficient ones (i.e., chiller 2 and chiller 5). This can be noticed from the fact that their supply temperatures, in steady state, reach the minimum value. Furthermore, the energy consumption is minimized and power saving reaches to 2.6%. The results for the second cooling load, i.e.,  $C_L = 5908.6$  kW, are shown in Figure 4.3c, where it can

---

<sup>1</sup>Simulation parameters are adapted from [2].

## 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

be noticed a similar performance to that obtained with the first cooling load. However, in this case, it is not necessary that the supply temperatures reach the minimum value to meet the required load. Newly, energy consumption is minimized and power saving reaches to 2.8%. As it is stated in Section 4.2, convergence and optimality of the method is guaranteed under the conditions given in Theorem 4.2.1. In both cases we use the early stopping criteria given in Section 4.2.

Although other techniques have been applied to solve the optimal chiller loading problem, e.g., the ones in [2], they require centralized information. In this regard, it is worth noting that the same objective is properly accomplished by using our approach, which is fully distributed.

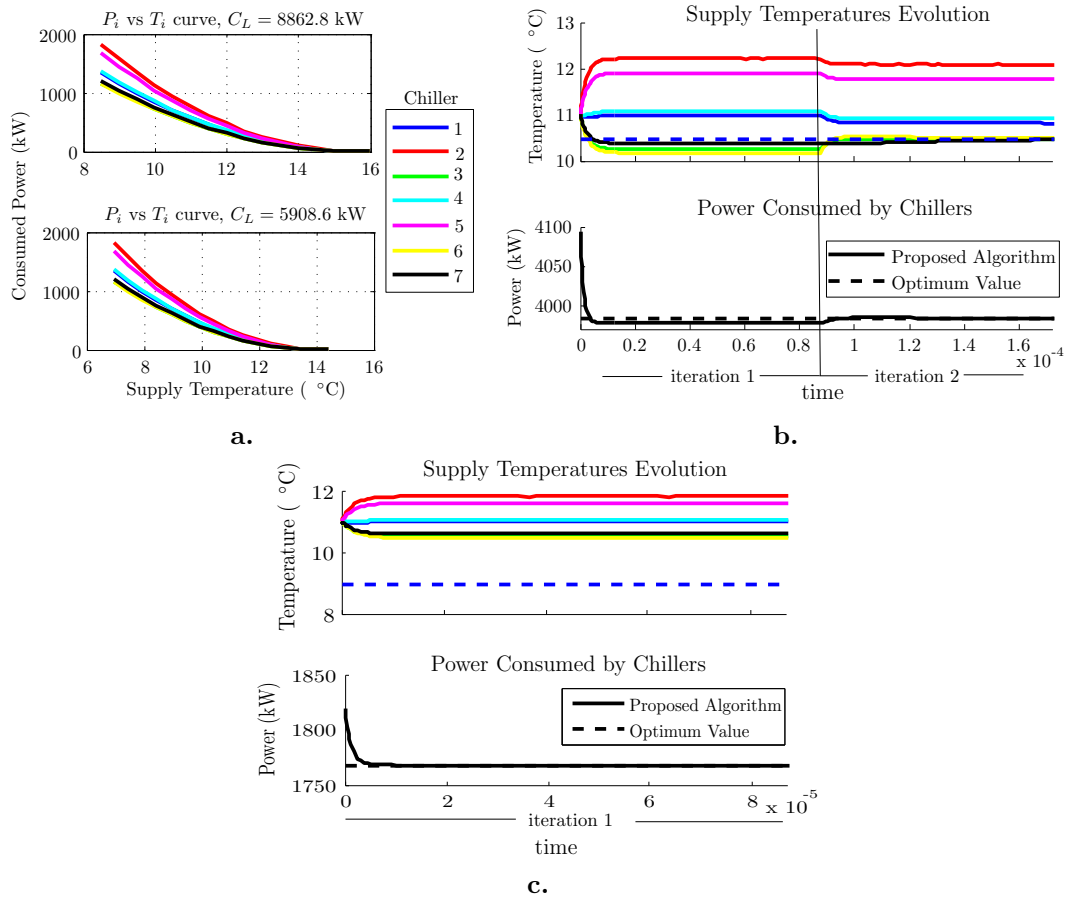
**Table 4.2:** Chillers' parameters

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
$k_{0,i}$	113.51	71.70	62.75	112.68	74.13	61.98	76.54
$k_{1,i}$	0.21	-0.45	0.49	0.18	-0.44	0.55	0.34
$k_{2,i}$	0.35	0.48	0.30	0.36	0.44	0.30	0.31
$k_{3,i}$	-8.19	-5.13	-4.53	-8.13	-5.30	-4.48	-5.527
$k_{4,i}$	0.43	-0.14	0.71	0.40	-0.13	0.76	0.54
$k_{5,i}$	0.14	0.09	0.08	0.14	0.09	0.08	0.10
$k_{6,i}$	0.01	-0.01	0.02	0.01	-0.01	0.02	0.01

### 4.5 Discussion

The method developed in this chapter solves the problem of resource allocation with lower bounds given in Equation (4.1). The main advantage of the proposed technique is its distributed nature; indeed, our approach does not need the implementation of a centralized coordinator. This characteristic is appealing, especially in applications where communications are strongly limited. Moreover, fully distributed methodologies increase the autonomy and resilience of the system in the face of possible failures.

In Section 4.3, we show by means of simulations that the performance of the proposed method does not decrease when the number of nodes (which are related to the decision variables of the optimization problem) is large, or the communication network that



**Figure 4.3:** a.  $P_i$ - $T_i$  curves for each chiller, b. Evolution of supply temperatures and total power consumed by the chillers,  $C_L = 8862.8$  kW, c. Evolution of supply temperatures and total power consumed by the chillers,  $C_L = 5908.6$  kW.

#### 4. OPTIMAL RESOURCE ALLOCATION WITH LOWER-BOUND CONSTRAINTS

---

allows the nodes to share information has few channels. In these cases, the behavior of our approach is better than that of other techniques found in the literature, such as the DIP method, or the DRD. Moreover, it is worth noting that our technique addresses the constraints as hard. This fact has two important consequences: *i*) in all cases, the solution satisfies the imposed constraints, and *ii*) the objective function (and therefore the optimum) is not modified (contrary to the DIP method that includes the constraints in the objective function decreasing the quality of the solution as shown in Section 4.3.4).

Other advantage of the method proposed in this chapter is that it does not require an initial feasible solution of the resource allocation problem (4.1). Similarly to the DIPe technique, our method only requires that the starting point satisfies the resource constraint (4.1b), i.e., we need that  $\sum_{i=1}^n x_i(0) = X$ . Notice that an initial solution  $x(0)$  that satisfies (4.1b) is not hard to obtain in a distributed manner. For instance, if we assume that only the  $k$ th node has the information of the available resource  $X$ , we can use  $(x_k(0) = X, \{x_i(0) = 0 : i \in \mathcal{V}, i \neq k\})$  as our starting point. Thus, an initialization phase is not required. In contrast, other distributed methods, such as DIP and DRD needs an initial feasible solution of the problem (4.1), i.e., a solution that satisfies (4.1b) and (4.1c). Finding this starting point is not a trivial problem for systems involving a large number of variables. Therefore, for these methods, it is necessary to employ distributed constraint satisfaction algorithm (as the one described in Section 3.4.2) as a first step.

On the other hand, we notice that to implement the early stopping criterion presented at the end of Section 4.2, it is required to perform an additional *min-consensus* step in each iteration. Despite this fact, if the number of nodes is large, this criterion saves computational time, because in most of the cases, all passive nodes are identified during the first iterations of Algorithm 1.

# 5

## Distributed Population Dynamics<sup>1</sup>

In the previous chapter, we briefly discussed the distributed replicator dynamics. We showed that, although this model does not reach the optimal solution of resource allocation problems that are subject to sparse graphs, its implementation is appealing to address applications that require fast computation and tolerate suboptimal solutions. The distributed replicator dynamics were proposed in [81] based on the classic replicator equation [105], which is a model that belongs to the field of population dynamics [3]. The purpose of this chapter is to extend the results in [81] for dealing with other models, different from the replicator equation. In general terms, we propose a procedure to obtain distributed population dynamics.

### 5.1 Motivation

Population dynamics [3, 106, 107] describe the dynamical process that a population experiences when there is a strategic interaction among the agents that comprise the population. The agents involved in this dynamical process evolve to an equilibrium according to a revision protocol, which establishes the individual decision rules that agents apply to choose the best strategies (i.e., those strategies earning higher payoffs). Population dynamics properties (e.g., passivity [108]) can be exploited to design solutions for a variety of engineering problems. For instance, designing of control and learning systems

---

<sup>1</sup>G. Obando and J. Barreiro-Gómez contributed equally to this chapter. Results are reported in [1].

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

[109, 110], optimization [111, 112], coordination [42], dynamic resource allocation [39], and so forth.

When using population dynamics for solving learning, control, and optimization problems, some elements of the problem are associated with “strategies” that agents in the population can adopt, and other elements are associated with “masses” of agents playing each strategy. This analogy has a direct implication in the information required to implement a solution based on a population dynamics algorithm, since the existing algorithms assume that the population is well-mixed [3, 106] (i.e., any pair of agents playing any pair of strategies can interact with each other). A consequence of the well-mixed population structure assumption is that the elements of the problem are allowed to interact each other without any constraint (i.e., following a full-information structure). Therefore, classic population dynamics are restricted to be implemented in problems characterized by a centralized information scheme. However, the number of problems that require distributed solutions has increased dramatically in the last few years. In this regard, some approaches have been proposed to model the interaction constraints in a population of players. For instance, the authors in [113, 114] deal with normal-form games and propose a graph-based interaction model, where each node in the graph represents an individual that repetitively plays a symmetric game with its neighbors. However, this approach is not suitable to be applied in the population game framework since, in this framework, it is preferable to avoid the individuation of players [3]. On the other hand, other approaches aim to apply learning algorithms that are capable to deal with information constraints [115, 116]. Similarly, the authors in [81] modify the well known replicator dynamics model to relax the full-information dependency. They propose a distributed replicator equation in which the evolution of each strategy is only governed by the “neighboring” strategies (according to a given topology).

This chapter extends the results in [81]. Our main contribution is the design of a general method that allows us to deduce several distributed population dynamics. The core of the proposed method is the use of the mean dynamics [3] in non-well-mixed populations. To illustrate our methodology, we present a distributed version of the fundamental population dynamics (those obtained by applying classic revision protocols), i.e., the distributed replicator dynamics, the distributed Smith dynamics, the distributed logit dynamics, and the distributed projection dynamics. It is worth noting that the deduction presented in this chapter can be used to generate other distributed dynamics



by using alternative revision protocols. Besides, we show that a well-mixed population obeys a structure given by a complete graph, whereas a non-well-mixed population has many different possible structures that are generally given by non-complete graphs. In this sense, the proposed approach is versatile to be implemented in a large variety of problems with different information structures. Moreover, we show that the distributed population dynamics exhibit similar stability and invariance properties as their classic counterpart.

## 5.2 Preliminaries

### 5.2.1 Graph Representation of Population Interactions

We use graph-theoretical tools to describe the constraints on the interactions among agents according to the population type, i.e., well-mixed or non-well-mixed. In this regard, let  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$  be a weighted graph, where  $\mathcal{V} = \{1, \dots, n\}$  is the set of nodes representing the strategies in the population;  $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{V}\}$  is the set of links representing possible interaction among agents playing different strategies; and  $\mathcal{A} = [a_{ij}]$  is an  $n \times n$  weighted adjacency matrix whose elements satisfy the following property:  $a_{ij} > 0$  if  $(i, j) \in \mathcal{E}$ ,  $a_{ij} = 0$  otherwise.

### 5.2.2 Population Games

Consider a population of mass  $m > 0$  comprised of a finite and large number of agents in a strategic interaction. Throughout this chapter, we assume without loss of generality, that the mass of the population is equal to one, i.e.,  $m = 1$ . As was stated before, the set of available strategies for the agents is given by  $\mathcal{V} = \{1, \dots, n\}$ . The scalar  $x_i \geq 0$  represents the fraction of the population mass that corresponds to the agents choosing the strategy  $i \in \mathcal{V}$ . The vector  $x \in \mathbb{R}_+^n$  is the state of the population with dimension  $n$  whose entries are nonnegative real numbers. Similarly,  $x \in \mathbb{R}_{++}^n$  denotes a vector of dimension  $n$  with strictly positive entries. The set of possible states of the population, which corresponds to all possible distributions of agents among the strategies, is given by the following simplex:

$$\Delta = \left\{ x \in \mathbb{R}_+^n : \sum_{i \in \mathcal{V}} x_i = 1 \right\}. \quad (5.1)$$

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

Agents playing the  $i$ th strategy obtain a reward given by a fitness function  $f_i(x)$ , i.e.,  $f_i : \Delta \rightarrow \mathbb{R}$  is a continuous map that specifies the payoff associated with the strategy  $i \in \mathcal{V}$ . Notice that a population game is completely characterized by the fitness vector  $F(x) = [f_1(x), \dots, f_n(x)]^\top$ . There are several types of games depending on  $F(x)$ . Below, we present the definition of two of them, which are found in a large number of applications [117].

**Definition 5.2.1.** *Let  $F : \mathbb{R}_+^n \mapsto \mathbb{R}^n$  be a population game with payoffs defined on the positive orthant. If there exists a continuously differentiable potential function  $V : \mathbb{R}_+^n \mapsto \mathbb{R}$  that satisfies  $\nabla V(x) = F(x)$ , for all  $x \in \mathbb{R}_+^n$ , then  $F$  is a full potential game.*

**Definition 5.2.2.** *The population game  $F : \Delta \mapsto \mathbb{R}^n$  is a stable game if:*

$$(y - x)^\top (F(y) - F(x)) \leq 0, \text{ for all } x, y \in \Delta.$$

The next theorem, adapted from [3], gives an equivalent characterization of a stable game in terms of the Jacobian matrix of  $F(x)$ , i.e.,  $DF(x)$ ; and the tangent space of the simplex  $\Delta$ .

**Theorem 5.2.1.** *Let the population game  $F : \Delta \mapsto \mathbb{R}^n$  be continuously differentiable.  $F$  is a stable game if and only if  $\xi^\top DF(x)\xi \leq 0$ , for all  $\xi \in T\Delta$ ,  $x \in \Delta$ , where  $T\Delta$  is the tangent space of the simplex  $\Delta$ , which is defined by  $T\Delta = \{z \in \mathbb{R}^n : \sum_{i \in \mathcal{V}} z_i = 0\}$ .*

A population game combined with a revision protocol lead to the emergence of population dynamics [3]. The function  $\rho : \mathbb{R}^n \times \Delta \rightarrow \mathbb{R}_+^{n \times n}$  is known as the revision protocol, and describes the timing and the result of the decisions of agents in the strategic interaction. The revision protocol takes the payoff vector  $F(x)$  and a determined population state  $x \in \Delta$ , returning a non-negative matrix, whose element of the  $i$ th row and  $j$ th column  $\rho_{ij}(F(x), x)$  represents the conditional switch rate from strategy  $i$  to strategy  $j$ , where  $i, j \in \mathcal{V}$ . Depending on the revision protocol used by the individuals, we can find several kinds of population dynamics (see Table 5.1), e.g., replicator dynamics, Smith dynamics, logit dynamics, projection dynamics, etc.

In order to simplify the notation for fitness functions, and revision protocols, we remove their corresponding arguments from now on, i.e.,  $f_i = f_i(x)$ , and  $\rho_{ij} = \rho_{ij}(F(x), x)$ .

**Table 5.1:** Some revision protocols and their corresponding population dynamics [3]

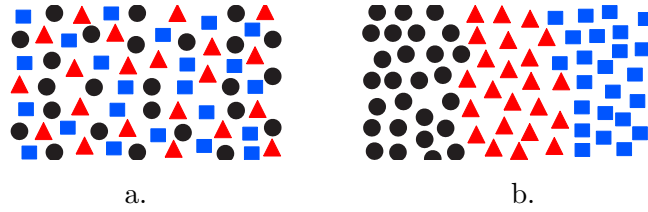
Revision protocol	Population dynamics
Pairwise proportional imitation: $\rho_{ij} = x_j[f_j - f_i]_+$	Replicator dynamics
Pairwise comparison: $\rho_{ij} = [f_j - f_i]_+$	Smith dynamics
Logit choice protocol: $\rho_{ij} = \frac{e^{\eta^{-1}f_j}}{\sum_{k \in \mathcal{V}} e^{\eta^{-1}f_k}}, \eta > 0$	Logit dynamics
Modified pairwise comparison: $\rho_{ij} = \frac{[f_j - f_i]_+}{x_i}$	Projection dynamics

$[\cdot]_+ := \max(\cdot, 0)$

### 5.3 General dynamics on graphs

The dynamics describing a population behavior depend on the population structure. In this regard, current literature assumes that the population under consideration is well-mixed, i.e., if we take any portion of the entire population, this contains all the strategies with the same probability. Figure 5.1a illustrates this fact by showing a population composed by a large and finite number of agents involved in a game. Each element in the figure represents an agent, and the shape of the element (“circle,” “square,” or “triangle”) denotes the strategy that the agent has adopted. In population games, all agents have the same probability to receive a revision opportunity. The agent receiving the revision opportunity randomly choose another agent from its neighbors and can change its own strategy by the neighbor’s strategy depending on the selected revision protocol. Since the population is well-mixed, the probability that the selected opponent is playing any of the available strategies is the same.

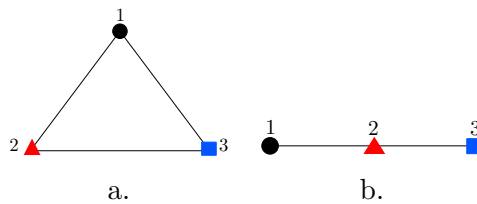
On the other hand, there could be a non-well-mixed population as the one shown in Figure 5.1b. For this population, all agents have the same probability to receive an opportunity to make a revision. However, the probability that the opponent is playing a particular strategy is not equal (e.g., if the strategy played by the agent receiving the revision opportunity is “square,” then there is the same probability to select an opponent playing strategy “triangle” or “square;” but the probability to select an opponent playing strategy “circle” is zero since the population structure does not allow it).



**Figure 5.1:** a. Well-mixed population. b. Non-well-mixed population.

Interactions among agents playing different strategies can be represented by a graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ . The set of nodes  $\mathcal{V}$  is associated with the available strategies and the set of links  $\mathcal{E}$  is related to the encounter probability between strategies, i.e., there exists a link between two strategies if their encounter probability is different from zero. Hence, the elements of the corresponding adjacency matrix  $\mathcal{A} = [a_{ij}]$  are as follows:  $a_{ij} = 1$  denotes that strategies  $i$  and  $j$  can encounter each other, while  $a_{ij} = 0$  denotes that the population structure makes impossible a matching between strategies  $i$  and  $j$ . According to this convention, the scenarios associated with well-mixed and non-well-mixed populations can be represented by two kinds of graphs. The well-mixed population case is always represented by a complete graph, whereas a non-well-mixed population is represented by a graph with a specific topology depending on the particular population structure (see Figure 5.2).

In this chapter, we assume that the encounter probability between strategies  $i$  and  $j$  is the same as the one of strategies  $j$  and  $i$ , i.e., the graph  $\mathcal{G}$  is undirected.



**Figure 5.2:** Graph representation of: a. The well-mixed population in Figure 5.1a. b. The non-well-mixed population in Figure 5.1b. Nodes 1,2, and 3 correspond, respectively, to strategies “circle,” “triangle,” and “square” of the proposed example.

### 5.3.1 Mean Dynamics

Taking into account the previously discussed considerations, this section formally describes the evolutionary process of a non-well-mixed population involved in a strategic game. Suppose that the population is composed by  $M$  agents, and each of them receives a revision opportunity that is given by an exponential distribution with rate  $R$ . Hence, during a time  $dt$ , the revision opportunity received by each agent is given by  $Rdt$ . Since we assume that the mass of the population is equal to one, the scalar  $x_i$  is equal to the portion of agents playing the  $i$ th strategy, and  $Mx_i$  is the total amount of agents playing strategy  $i \in \mathcal{V}$ . Consequently, the expected number of revision opportunities received by agents playing the  $i$ th strategy is approximately  $Mx_i Rdt$  during  $dt$  (notice that  $x_i$  may vary during  $dt$ ; however, this variation is negligible if  $dt$  is small). Agents playing  $i \in \mathcal{V}$  switch to strategy  $j \in \mathcal{V}$  with a probability that depends on the revision protocol, the probability distribution of receiving a revision opportunity, and the encounter probability between strategies  $i$  and  $j$  (given by the population structure, which is represented by the graph  $\mathcal{G}$ ), i.e.,  $a_{ij}\rho_{ij}/R$ . Finally, the expected number of agents switching from strategy  $i \in \mathcal{V}$  to strategy  $j \in \mathcal{V}$  during time  $dt$  is  $Mx_i a_{ij}\rho_{ij}dt$ .

Now, if we consider all possible strategies in the population, the expected number of agents switching to strategy  $i \in \mathcal{V}$  is given by

$$M \sum_{j \in \mathcal{V}} x_j a_{ji} \rho_{ji} dt,$$

and the expected number of agents playing strategy  $i \in \mathcal{V}$  changing to other strategies is given by

$$Mx_i \sum_{j \in \mathcal{V}} a_{ij} \rho_{ij} dt.$$

Therefore, the variation of the proportion of agents playing the  $i$ th strategy is deduced by a mass balance as follows,

$$\dot{x}_i = \sum_{j \in \mathcal{V}} x_j a_{ji} \rho_{ji} - x_i \sum_{j \in \mathcal{V}} a_{ij} \rho_{ij}.$$

This equation corresponds to the *Distributed Mean Dynamics*, or mean dynamics for non-well-mixed populations. Since  $\mathcal{G}$  is undirected, notice that the distributed mean dynamics can be rewritten as follows,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} x_j \rho_{ji} - x_i \sum_{j \in \mathcal{N}_i} \rho_{ij}. \quad (5.2)$$

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

For complete graphs, i.e., for well-mixed-populations, we have that  $\mathcal{N}_i = \mathcal{V}$ , obtaining the classic *Mean Dynamics* [3].

### 5.3.2 Distributed Population Dynamics

Distributed mean dynamics allow the inference of population dynamics involving non-well-mixed populations comprised of agents that are programmed with a specific revision protocol. This section shows the deduction of different distributed population dynamics using (5.2). The deduced dynamics are named after the classic population dynamics that are generated with the corresponding revision protocol.

#### 5.3.2.1 Distributed Replicator Dynamics (DRD)

The distributed replicator dynamics are obtained from the distributed mean dynamics using the *pairwise proportional imitation* protocol (Table 5.1), as follows,

$$\begin{aligned}\dot{x}_i &= \sum_{j \in \mathcal{N}_i} x_j x_i [f_i - f_j]_+ - x_i \sum_{j \in \mathcal{N}_i} x_j [f_j - f_i]_+ \\ &= \sum_{j \in \mathcal{N}_i} x_j x_i (f_i - f_j).\end{aligned}$$

Finally, the distributed replicator dynamics are given by,

$$\dot{x}_i = x_i \left( f_i \sum_{j \in \mathcal{N}_i} x_j - \sum_{j \in \mathcal{N}_i} x_j f_j \right). \quad (5.3)$$

#### 5.3.2.2 Distributed Smith Dynamics (DSD)

In this case, we use the *pairwise comparison* protocol (Table 5.1). Substituting this revision protocol in (5.2), we get

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} x_j [f_i - f_j]_+ - x_i \sum_{j \in \mathcal{N}_i} [f_j - f_i]_+. \quad (5.4)$$

Notice that (5.4) can be written as,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left( (1 - \phi_{ij}) x_i + (1 + \phi_{ij}) x_j \right) [f_i - f_j],$$

where  $\phi_{ij} = \text{sgn}(f_i - f_j)$ .

### 5.3.2.3 Distributed Logit Dynamics (DLD)

The deduction of the distributed logit dynamics is based on the *logit choice* protocol (Table 5.1). However, notice that this protocol requires full information since the sum at the denominator is taken over all the strategies. In order to satisfy the information constraint given by the graph  $\mathcal{G}$ , we modify the protocol as follows,

$$\rho_{ij}(F, x) = e^{\eta^{-1}f_j}, \eta > 0.$$

Distributed logit dynamics are obtained by replacing the above protocol in the distributed mean dynamics, i.e.,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} x_j e^{\eta^{-1}f_i} - x_i \sum_{j \in \mathcal{N}_i} e^{\eta^{-1}f_j}. \quad (5.5)$$

### 5.3.2.4 Distributed Projection Dynamics (DPD)

The *projection dynamics* use the *modified pairwise comparison* protocol, i.e.,

$$\begin{aligned} \dot{x}_i &= \sum_{j \in \mathcal{N}_i} x_j \frac{[f_i - f_j]_+}{x_j} - x_i \sum_{j \in \mathcal{N}_i} \frac{[f_j - f_i]_+}{x_i}, \\ &= \sum_{j \in \mathcal{N}_i} (f_i - f_j). \end{aligned}$$

Thus, the distributed projection dynamics are given by

$$\dot{x}_i = |\mathcal{N}_i|f_i - \sum_{j \in \mathcal{N}_i} f_j, \quad (5.6)$$

where  $|\mathcal{N}_i|$  denotes the cardinality of the set  $\mathcal{N}_i$ , i.e., the number of neighbors of the  $i$ th node.

## 5.4 Invariant Set and Stability Analysis

### 5.4.1 Invariant Set Analysis

As was mentioned in Section 5.2, the population mass does not vary over time. Hence, all possible states generated during the evolution of the population should belong to the simplex  $\Delta$  given in (5.1). This section shows that the simplex  $\Delta$  is an invariant set under the distributed population dynamics deduced in the previous section.

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

**Theorem 5.4.1.** *The simplex  $\Delta$  is an invariant set under: the distributed replicator dynamics (5.3), the distributed Smith dynamics (5.4), and the distributed logit dynamics (5.5).*

*Proof.* According to Equation (5.1),  $\Delta$  has to conditions: *i)*  $\sum_{i \in \mathcal{V}} x_i = 1$  (mass conservation); *ii)*  $x_i \geq 0$ , for all  $i \in \mathcal{V}$  (non-negativeness).

First, we prove that DRD, DSD, and DLD satisfy condition *i)*. Notice that this is equivalent to show that  $\sum_{i \in \mathcal{V}} \dot{x}_i = 0$  under the distributed mean dynamics (5.2). These dynamics can be written by using the adjacency matrix of the graph  $\mathcal{G}$  as follows

$$\dot{x}_i = \sum_{j \in \mathcal{V}} a_{ij} \rho_{ji} x_j - \sum_{j \in \mathcal{V}} a_{ij} \rho_{ij} x_i.$$

Hence,

$$\sum_{i \in \mathcal{V}} \dot{x}_i = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} \rho_{ji} x_j - \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} \rho_{ij} x_i.$$

Since  $\mathcal{G}$  is undirected (i.e.,  $a_{ij} = a_{ji}$ ), we have

$$\begin{aligned} \sum_{i \in \mathcal{V}} \dot{x}_i &= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ji} \rho_{ji} x_j - \sum_{j \in \mathcal{V}} \sum_{i \in \mathcal{V}} a_{ji} \rho_{ji} x_j \\ &= 0. \end{aligned}$$

Second, we prove that each dynamic satisfies condition *ii)*:

- DRD: Non-negativeness of each  $x_i$  is satisfied given the fact that  $\dot{x}_i = 0$  if  $x_i = 0$  under distributed replicator dynamics. Thus, if  $x_i(0) \geq 0$ , then  $x_i(t) \geq 0$  for all  $t \geq 0$ .
- DSD: According to (5.4), notice that when  $x_i = 0$  for any  $i \in \mathcal{V}$ , then  $\dot{x}_i \geq 0$ . Hence, the non-negativeness of  $x_i$  is satisfied under distributed Smith dynamics.
- DLD: Notice that  $\dot{x}_i \geq 0$  when  $x_i = 0$  under distributed logit dynamics (5.5). Therefore, if  $x(0) \in \Delta$ , then  $x_i(t) \geq 0$  for all  $t \geq 0$ .

□

**Proposition 5.4.1.** *The set  $\Delta' = \{x \in \mathbb{R}^n : \sum_{i \in \mathcal{V}} x_i = 1\}$  is invariant under the distributed projection dynamics (5.6).*



*Proof.* The distributed projection dynamics can be written by using the adjacency matrix of the graph  $\mathcal{G}$  as follows

$$\dot{x}_i = \sum_{j \in \mathcal{V}} a_{ij} f_i - \sum_{j \in \mathcal{V}} a_{ij} f_j.$$

Therefore,

$$\sum_{i \in \mathcal{V}} \dot{x}_i = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} f_i - \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} f_j.$$

Since  $a_{ij} = a_{ji}$  because  $\mathcal{G}$  is undirected, we obtain

$$\begin{aligned} \sum_{i \in \mathcal{V}} \dot{x}_i &= \sum_{j \in \mathcal{V}} \sum_{i \in \mathcal{V}} a_{ji} f_i - \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} f_j \\ &= 0. \end{aligned} \quad \square$$

**Remark 5.4.1.** *It should be noticed that the distributed projection dynamics satisfy one of the conditions of the original simplex  $\Delta$ , i.e.,  $\sum_{i \in \mathcal{V}} x_i = 1$  (mass conservation). However, the non-negativeness of  $x_i$  is not guaranteed. This fact also occurs in the classic projection dynamics.*  $\diamond$

**Remark 5.4.2.** *Notice that Theorem 5.4.1 and Proposition 5.4.1 do not impose any conditions on the interaction graph  $\mathcal{G}$ . Thus, the studied distributed population dynamics exhibit simplex invariance under any population structure.*  $\diamond$

### 5.4.2 Stability Analysis

Classic population dynamics usually converge to Nash equilibria since Nash equilibria correspond to the expected outcome of games played by rational individuals (i.e., individuals that are trying to maximize their profit). Given a population game  $F$ , the set of Nash equilibria is defined as follows [3],

$$NE(F) = \{x^* \in \Delta : x_i^* > 0 \Rightarrow f_i(x^*) \geq f_j(x^*), \forall i, j \in \mathcal{V}\}.$$

Thus, in a Nash equilibrium, all players perceives the same profit.

This section provides sufficient conditions guaranteeing that a Nash equilibrium  $x^*$  of the population game  $F$  is asymptotically stable under the distributed population dynamics derived in Section 5.3.2. These conditions, which are related to the connectivity of the interaction graph and the characteristics of the Nash equilibrium, are summarized in the following assumptions.

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

**Assumption 5.4.1.** *The graph  $\mathcal{G}$  that describes the population structure is connected.*

**Assumption 5.4.2.** *The Nash equilibrium  $x^*$  belongs to the interior of the simplex  $\Delta$ , i.e.,  $x^* \in \text{int}\Delta$ , where  $\text{int}\Delta = \{x \in \mathbb{R}_{++}^n : \sum_{i \in \mathcal{V}} x_i = 1\}$ .*

Below, we provide our results on convergence of the distributed population dynamics to a Nash equilibrium.

**Theorem 5.4.2.** *Let  $F$  be a full potential game with strictly concave potential function  $V(x)$ , and let  $x^* \in NE(F)$ . If Assumptions 5.4.1 and 5.4.2 hold, then  $x^*$  is asymptotically stable under the distributed replicator dynamics (5.3) and the distributed Smith dynamics (5.4).*

*Proof.* Since  $x^* \in NE(F)$  and  $x^* \in \text{int}\Delta$ , we conclude that  $f_i(x^*) = f_j(x^*)$ , for all  $i, j \in \mathcal{V}$ . Moreover, notice that  $x^* = \arg \max_{x \in \Delta} V(x)$  (applying the Karush–Kuhn–Tucker conditions). Additionally, since  $V(x)$  is strictly concave, we can take  $E_V(x) = V(x^*) - V(x)$  as a Lyapunov function candidate. The derivative of  $E_V(x)$  along the trajectories of DRD (5.3) and DSD (5.4) is given by

$$\begin{aligned} \dot{E}_V(x) &= -(\nabla V(x))^\top \dot{x} \\ &= -F^\top \dot{x} \\ &= -F^\top L(x) F, \end{aligned}$$

where  $L(x) = [l_{ij}^{(x)}]$  is a matrix whose entries  $l_{ij}^{(x)}$  are defined as follows,

$$\begin{aligned} \text{for DRD: } l_{ij}^{(x)} &= \begin{cases} -a_{ij}x_i x_j, & \text{if } i \neq j \\ \sum_{k \in \mathcal{V}, k \neq i} a_{ik}x_i x_k, & \text{if } i = j, \end{cases} \\ \text{for DSD: } l_{ij}^{(x)} &= \begin{cases} -\frac{a_{ij}}{2} \left( (1 - \phi_{ij})x_i + (1 + \phi_{ij})x_j \right), & \text{if } i \neq j \\ \sum_{k \in \mathcal{V}, k \neq i} \frac{a_{ik}}{2} \left( (1 - \phi_{ik})x_i + (1 + \phi_{ik})x_k \right), & \text{if } i = j. \end{cases} \end{aligned}$$

Notice that  $L(x)$  is the Laplacian of the undirected graph given by the tuple  $\mathcal{G}^{(x)} = (\mathcal{V}, \mathcal{E}, \mathcal{A}^{(x)})$ , where  $\mathcal{A}^{(x)} = [a_{ij}^{(x)}]$  is the adjacency matrix whose entries are defined as follows:  $a_{ij}^{(x)} = a_{ij}x_i x_j$ , for DRD; and  $a_{ij}^{(x)} = \frac{a_{ij}}{2} \left( (1 - \phi_{ij})x_i + (1 + \phi_{ij})x_j \right)$ , for DSD. These entries are nonnegative since  $x \in \Delta$ . Thus,  $L(x) \geq 0$  and  $\dot{E}_V(x) \leq 0$ . Therefore,  $x^*$  is stable under DRD and DSD.

Considering that  $x^* \in \text{int}\Delta$  is stable, a set  $\mathcal{B}$  around  $x^*$  can be defined such that if  $x(0) \in \mathcal{B}$ , then  $x(t) \in \text{int}\Delta$ , for all  $t \geq 0$  (it is possible to show that  $\mathcal{B} = \text{int}\Delta$  for DRD).

Thus, if  $x(0) \in \mathcal{B}$ , the null space of  $L^{(x)}$  is equal to  $\text{span}\{\mathbf{1}\}$  (we use  $\mathbf{1}$  to denote a vector of dimension  $n$  whose entries are all 1) since  $\mathcal{G}^{(x)}$  is connected (we conclude that  $\mathcal{G}^{(x)}$  is connected since: *i*)  $\mathcal{G}^{(x)}$  and  $\mathcal{G}$  have the same topology in  $\mathcal{B}$ , i.e., if  $x \in \mathcal{B}$ ,  $a_{ij}^{(x)} = 0$  only if  $a_{ij} = 0$ ; *ii*)  $\mathcal{G}$  is connected by assumption). In this case,  $\dot{E}_V(x) = 0$  if and only if  $f_i = f_j$ , for all  $i, j \in V$ , i.e.,  $\dot{E}_V(x) = 0$  only in  $x^*$ . Therefore,  $x^*$  is asymptotically stable.  $\square$

**Remark 5.4.3.** *Proposition 5.4.2 requires that, in steady state, all strategies are played by the individuals involved in the game. Indeed, when any proportion of individuals is extinct at equilibrium (i.e.,  $x_i^* = 0$  for some  $i \in \mathcal{V}$ ), then convergence of the distributed replicator equation and the distributed Smith dynamics to the Nash equilibrium is not guaranteed.*  $\diamond$

**Theorem 5.4.3.** *Let  $F$  be a continuously differentiable stable game, let  $x^* \in NE(F)$ , and let  $\dot{x}$  be the distributed projection dynamics (5.6). If Assumptions 5.4.1 and 5.4.2 hold, then  $x^*$  is asymptotically stable.*

*Proof.* Consider the pairwise comparison protocol  $\rho_{ij} = [f_j - f_i]_+$ , and define

$$\rho_{ij} = \varphi(f_j - f_i),$$

where  $\varphi(\cdot) = [\cdot]_+$ . Then, consider the Lyapunov function candidate:

$$V(x) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} \int_0^{f_j - f_i} \varphi(s) ds.$$

Since  $\varphi : \mathbb{R} \mapsto \mathbb{R}_+$  is increasing on  $[0, +\infty)$  and  $\mathcal{G}$  is connected, the function  $V(x) > 0$ , for all  $x \neq x^*$ . Additionally,  $V(x^*) = 0$  since  $f_j(x^*) = f_i(x^*)$ , for all  $i, j \in \mathcal{V}$ . Moreover, notice that

$$\begin{aligned} \frac{\partial V(x)}{\partial x_l} &= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} \left( \frac{\partial f_j}{\partial x_l} - \frac{\partial f_i}{\partial x_l} \right) \varphi(f_j - f_i) \\ &= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ij} \varphi(f_j - f_i) \frac{\partial f_j}{\partial x_l} - \sum_{j \in \mathcal{V}} \sum_{i \in \mathcal{V}} a_{ji} \varphi(f_i - f_j) \frac{\partial f_j}{\partial x_l}. \end{aligned}$$

Taking into account that  $a_{ij} = a_{ji}$ , we obtain

$$\begin{aligned} \frac{\partial V(x)}{\partial x_l} &= \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} a_{ji} (f_j - f_i) \frac{\partial f_j}{\partial x_l} \\ &= \sum_{j \in \mathcal{V}} \frac{\partial f_j}{\partial x_l} \sum_{i \in \mathcal{N}_j} (f_j - f_i). \end{aligned}$$

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

According to (5.6), notice that  $\sum_{i \in \mathcal{N}_j} (f_j - f_i) = \dot{x}_j$ , where  $\dot{x}_j$  is the  $j$ th element of the distributed projection dynamics  $\dot{x}$ . Hence,

$$\frac{\partial V(x)}{\partial x_l} = \sum_{j \in \mathcal{V}} \dot{x}_j \frac{\partial f_j}{\partial x_l}. \quad (5.7)$$

Therefore, the time derivative of the Lyapunov function is

$$\begin{aligned} \dot{V}(x) &= (\nabla V(x))^\top \dot{x} \\ &= \dot{x}^\top DF(x)\dot{x}, \end{aligned}$$

where  $\dot{x}^\top DF(x)\dot{x} \leq 0$  since  $F$  is stable. □

**Remark 5.4.4.** *As was stated in the proof of Theorem 5.4.2, the fact that a Nash equilibrium  $x^*$  belongs to  $\text{int}\Delta$  implies that all the fitness functions reach the same value, i.e.,  $f_i(x^*) = f_j(x^*)$ , for all  $i, j \in \mathcal{V}$ . Therefore, the results given in Theorems 5.4.2 and 5.4.3 are related to the contributions reported in the literature on consensus in multi-agent networks (e.g., see [26, 28, 33]). An essential difference is that Theorems 5.4.2 and 5.4.3 show a direct relationship between game-theoretic properties and Lyapunov stability of a population game under distributed dynamics. ◇*

The connectivity condition of the graph  $\mathcal{G}$  in Theorems 3 and 4 is sufficient for Nash equilibrium stability. Regarding this fact, it is interesting to study if this condition is also necessary. The following proposition gives us insights on this issue.

**Proposition 5.4.2.** *Assume that the population game  $F$  has a unique Nash equilibrium, which is in the interior of the simplex  $\Delta$ , i.e.,  $x^* \in \text{int}\Delta$ . Let  $\dot{x}$  be the distributed mean dynamics (5.2). If  $x(t)$  asymptotically converges to  $x^*$ , for all  $x(0) \in \text{int}\Delta$ , then the graph  $\mathcal{G}$  is connected.*

*Proof.* We prove the contrapositive. Assume that  $\mathcal{G}$  is non-connected. We can express  $\mathcal{G}$  as the union of  $r \geq 2$  connected components (maximal connected sub-graphs) denoted by  $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{E}^p)$ , where  $p = 1, \dots, r$ , i.e.,  $\mathcal{G} = \bigcup_{p=1}^r \mathcal{G}^p$ . We use the arguments in the proof of Theorem 5.4.1 to conclude that, under the distributed mean dynamics,  $\sum_{i \in \mathcal{V}^p} x_i(t) = \sum_{i \in \mathcal{V}^p} x_i(0)$ , for all  $p = 1, \dots, r$ , and for all  $t \geq 0$ . Take two connected components  $\mathcal{G}^1$  and  $\mathcal{G}^2$  of the graph  $\mathcal{G}$ . Furthermore, take the following initial condition:

$$x_i(0) = \begin{cases} x_i^* + \frac{\varepsilon}{|\mathcal{V}^1|} & \text{if } i \in \mathcal{V}^1 \\ x_i^* - \frac{\varepsilon}{|\mathcal{V}^2|} & \text{if } i \in \mathcal{V}^2 \\ x_i^* & \text{otherwise} \end{cases}$$

where  $i \in \mathcal{V}$ , and  $\varepsilon > 0$ . Notice that, for small values of  $\varepsilon$ ,  $x(0) \in \text{int}\Delta$  since  $x^* \in \text{int}\Delta$ . Under this initial condition, it is not possible that  $x(t)$  converges to the unique Nash equilibrium  $x^*$  since  $\sum_{i \in \mathcal{V}^1} x_i(t) = \sum_{i \in \mathcal{V}^1} x_i(0) > \sum_{i \in \mathcal{V}^1} x_i^*$ , for all  $t \geq 0$ .  $\square$

Therefore, if we need to guarantee convergence to the Nash equilibrium from any initial condition inside the simplex  $\Delta$ , we require that the graph  $\mathcal{G}$  is connected. However, this condition might not be necessary if the initial conditions  $x(0)$  are constrained. For instance, suppose that the graph  $\mathcal{G}$  in the population game is non-connected. Moreover,  $\mathcal{G}$  is composed of  $r$  connected components (maximal connected sub-graphs) denoted by  $\mathcal{G}^p = (\mathcal{V}^p, \mathcal{E}^p)$ , where  $p = 1, \dots, r$ , i.e.,  $\mathcal{G} = \bigcup_{p=1}^r \mathcal{G}^p$ . Then, it can be shown (following the same reasoning as in proof of Proposition 5.4.2) that the equilibrium point  $x^* \in \text{int}\Delta$  is asymptotically stable if  $\sum_{i \in \mathcal{V}^p} x_i^* = \sum_{i \in \mathcal{V}^p} x_i(0)$  for all  $p = 1, \dots, r$ . Consequently, in this case, the connectivity condition of  $\mathcal{G}$  is not necessary.

## 5.5 Illustrative Examples

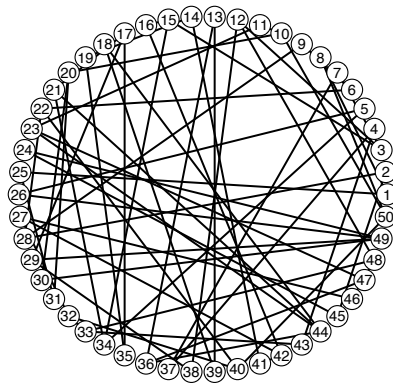
Classic population dynamics have been applied to solve a large number of engineering problems. For instance, access control in communication networks [42], combinatorial optimization [118], bandwidth allocation [41], hierarchical frequency control in microgrids [44], dispatch of electric generators [119], building temperature control [39], constrained extremum seeking [56], control of drinking water networks [120], and so forth. The same applications can be addressed by using distributed population dynamics. This approach has some benefits related to the information privacy, resilience to central failures, and parallelization of the computations. In order to show the versatility of the distributed population dynamics, three application examples are presented corresponding to the following key areas: distributed optimization, classic games, and distributed control design.

### 5.5.1 Solving a Distributed Optimization Problem

First, we propose the following distributed optimization problem,

$$\begin{aligned} \max \quad & V(x) := -x^\top x + b^\top x \\ \text{s.t.} \quad & \sum_{i=1}^{50} x_i = 1 \\ & x_i > 0 \text{ for all } i = 1, \dots, 50, \end{aligned} \tag{5.8}$$

where  $x \in \mathbb{R}^{50}$  is the vector of decision variables, and  $b \in \mathbb{R}^{50}$  is a vector of constants, whose entries are given by  $b_i = \frac{2i}{1275}$ , i.e.,  $b = \frac{1}{1275}[2 \ 4 \ \dots \ 100]^\top$ . Each decision variable is managed by a node in a network. Furthermore, we impose an information constraint given by the graph shown in Figure 5.3. This graph is obtained by following the Erdős–Rényi model (which is the simplest model of several kind of social and biological networks [121]) with edge generation probability equal to 0.01. Besides, we add a path connecting all nodes to guarantee that the generated graph is connected. The information constraint implies that the  $i$ th node only has information about the state of its neighbors.



**Figure 5.3:** Non-complete graph for the distributed optimization example.

To solve the problem in (5.8), we define a full potential game  $F = \left[ \frac{\partial V}{\partial x_1}, \dots, \frac{\partial V}{\partial x_{50}} \right]^\top$  (i.e., the fitness functions correspond to the marginal utilities) and apply the distributed population dynamics derived in Section 5.3.2. Notice that all nodes satisfy the information constraint (this fact is not possible by using the classic population dynamics). Results are shown in Figure 5.4 considering an initial condition  $x_i(0) = \frac{1}{50}$ , for all

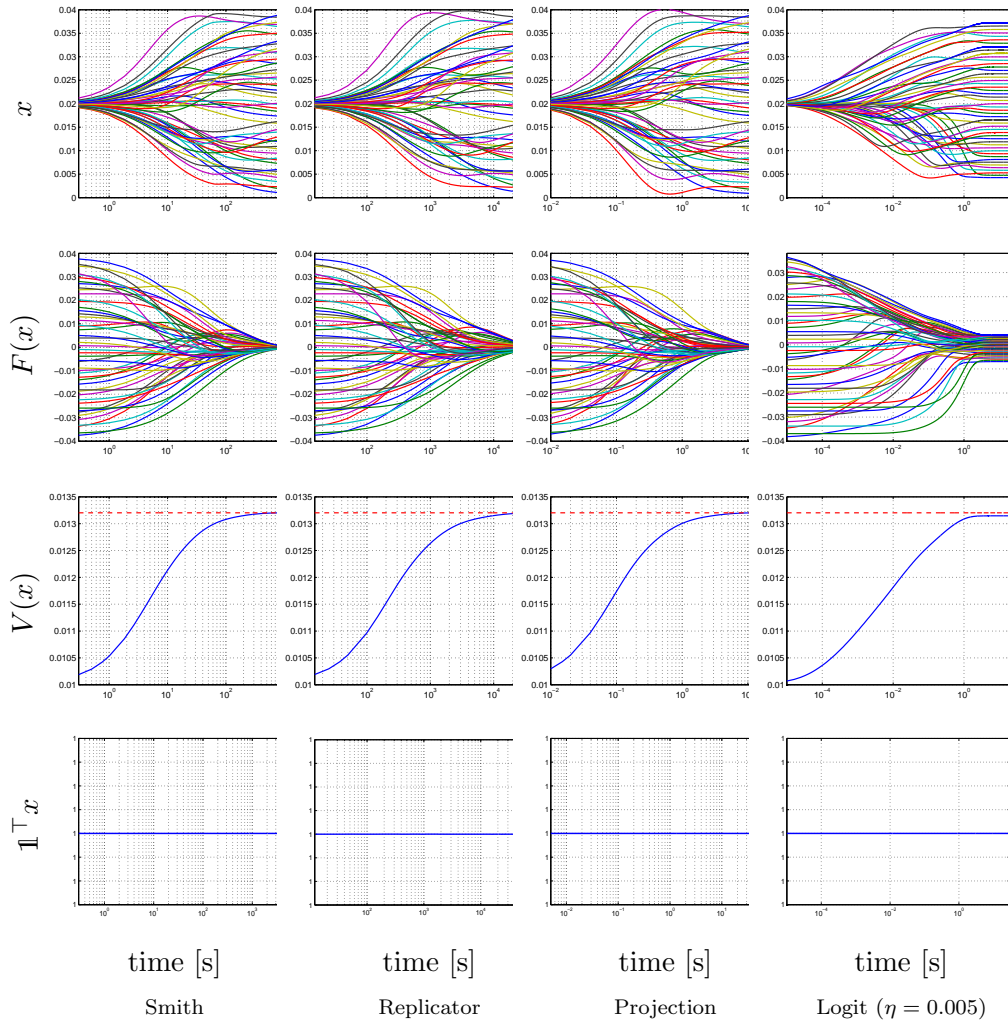
$i = 1, \dots, 50$ . First and fourth row of Figure 5.4 show that  $x(t)$  satisfies the problem constraints for all time, i.e.,  $x_i(t)$  remains nonnegative, for all  $i = 1, \dots, 50$ ; and  $\sum_{i=1}^{50} x_i(t) = 1$ . Furthermore, third row of Figure 5.4 shows that all distributed dynamics increase the objective function  $V(x)$ . However, only DRD, DSD, and DPD reach the optimum value (which are depicted in dashed red line). According to the second row of Figure 5.4, DRD, DSD, and DPD equalize the fitness functions' values in steady state, i.e., these dynamics converge to a Nash equilibrium. This behavior is consistent with the results stated in Theorems 5.4.2 and 5.4.3 since  $V(x)$  corresponds to a strictly concave potential function, i.e.,  $F$  is a full potential and stable game. Moreover, convergence time varies from one dynamic to another. DLD shows the fastest time response while the convergence of DRD is the slowest.

The academical example proposed in Equation (5.8) can be adapted to model practical optimization problems. For instance, let us consider the economic dispatch of distributed generators reported in [81]. This problem is stated as follows: consider a set of  $n$  generators that are connected to the electric distribution network. These generators have to supply a certain load denoted by  $L$ . Therefore,  $\sum_{i=1}^n p_i = L$ , where  $p_i \geq 0$  is the power supplied by the  $i$ th generator. The cost of production of  $p_i$  is given by a quadratic function  $c_i(p_i) = \alpha_i + \beta_i p_i + \gamma_i p_i^2$ , where  $\alpha_i, \beta_i, \gamma > 0$  are parameters associated with the  $i$ -th generator. The goal of the economic dispatch problem is to minimize the total cost of energy production, which is given by  $J(p_1, \dots, p_n) := \sum_{i=1}^n c_i(p_i)$ . Mathematically, this problem is formulated as follows:

$$\begin{aligned} \min \quad & J(p_1, \dots, p_n) := \sum_{i=1}^n \alpha_i + \beta_i p_i + \gamma_i p_i^2 \\ \text{s.t.} \quad & \sum_{i=1}^n p_i = L \\ & p_i > 0 \text{ for all } i = 1, \dots, n. \end{aligned} \tag{5.9}$$

Notice that the formulation of problems (5.8) and (5.9) are similar. Indeed, if we let the generators be the strategies of a game, and the power supplied by the  $i$ th generator be the amount of population playing the strategy  $i$ . Then, it is possible to model the economic dispatch of distributed generators as a population game, where the fitness functions are given by the marginal cost of energy production, i.e.,  $f_i = \frac{\partial J}{\partial p_i}$ , for all  $i = 1, \dots, n$ . In this regard, we can address the economic dispatch of distributed generators by using

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>



**Figure 5.4:** Evolution under different distributed population dynamics: states (1st row), fitness functions (2nd row), objective function (3rd row), equality constraint (4th row).

the same procedure described in the solution of the optimization problem (5.8). This approach is appealing, specially in scenarios where the number of distributed generators is large.



### 5.5.2 Classic Population Games: Bad Rock–Paper–Scissors with a Twin

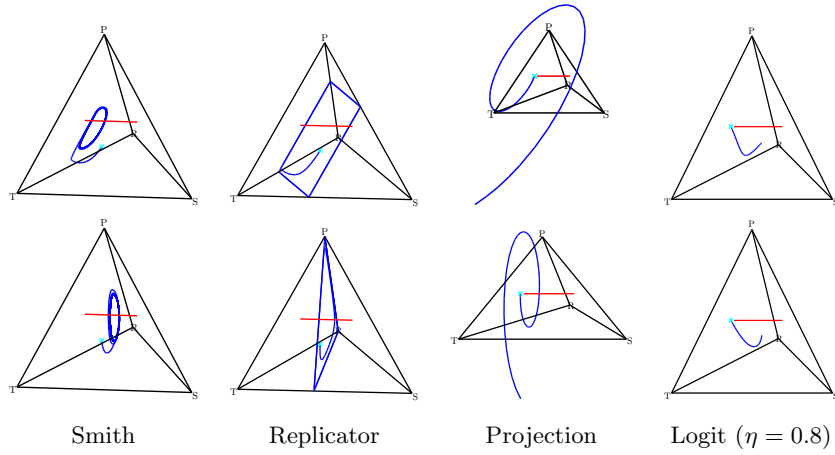
To evaluate the behavior of a non–well mixed population involved in a strategic interaction, we use a classic game called “bad rock–paper–scissor with a twin” (adapted from [3]). Moreover, we compare the performance of the distributed dynamics proposed in Section 5.3.2 with their classic counterpart (i.e., assuming a well–mixed population).

The bad rock–paper–scissor with a twin preserves the same rules as the rock–paper–scissors game. The difference is that the losing strategies have double penalty, i.e., this is not a zero–sum game. Moreover, another strategy called “twin” is added, which earns the same payoff as scissors. Summarizing, this game can be represented by the following payoff matrix

$$A = \begin{pmatrix} 0 & -2 & 1 & 1 \\ 1 & 0 & -2 & -2 \\ -2 & 1 & 0 & 0 \\ -2 & 1 & 0 & 0 \end{pmatrix},$$

where the first, second, third, and fourth column/row corresponds to rock, paper, scissors, and twin, respectively. In this case, the fitness functions are given by  $F(x) = Ax$ .

Evolution of the population state is shown in Figure 5.5 under distributed (first row) and classic (second row) dynamics. For the distributed case, we use a path graph. It can be noticed a similar behavior between the trajectories obtained by using the classic population dynamics and its distributed counterpart. DSD, DRD, and DPD exhibit oscillations around the set of Nash equilibria (depicted in red). Indeed, a limit cycle emerges in each case. The main difference is that the limit cycle lies in the plane orthogonal to the set of Nash equilibria for the classic dynamics while it is not orthogonal for the distributed case. Additionally, the angle formed by the plane containing the limit cycle and the set of Nash equilibria is the same for DSD, DRD, and DPD. On the other hand, both DLD and classic logit dynamics reach the same equilibrium point. However, this rest point does not belong to the set of Nash Equilibria. We notice the fact that this characteristic has been reported before in the literature related to classic logit dynamics (e.g., see [3]).



**Figure 5.5:** Distributed (up) and classic (bottom) population dynamics applied to bad rock(R)–paper(P)–scissors(S) with a twin(T).

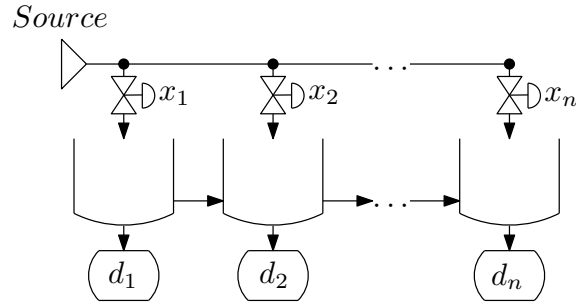
### 5.5.3 Distributed Control of Dynamical Systems

Finally, we design a distributed controller for the optimal transportation of drinking water<sup>2</sup>(relationship between distributed control and games has been pointed out in several papers, e.g., [122, 123]). The system is composed by  $n$  coupled tanks as shown in Figure 5.6. The arrows in the graphical representation show how flow directions are. Each tank has an outflow given by an unknown demand considered as a disturbance that is denoted by  $d_i$ , and an inflow  $u_i$  from a limited water source, i.e., the control outputs are subject to a constraint given by  $\sum_{i=1}^n u_i \leq K$ , where  $K$  is the total available resource. The mentioned inflows are controlled by  $x_i$  that determines a percentage of the total resource  $K$ , then  $u_i = Kx_i$ , and  $\sum_{i=1}^n x_i = 1$ . We assume that there are local controllers at the valves to guarantee the desired inflow according to the signal  $x_i$ . The dynamics for this system are as follows,

$$\begin{aligned} \frac{dh_1}{dt} &= u_1 - \sqrt{\rho gh_1} - d_1 \\ \frac{dh_i}{dt} &= u_i + \sqrt{\rho gh_{i-1}} - \sqrt{\rho gh_i} - d_i, \quad i = 2, \dots, n-1 \\ \frac{dh_n}{dt} &= u_n + \sqrt{\rho gh_{n-1}} - d_n, \end{aligned}$$

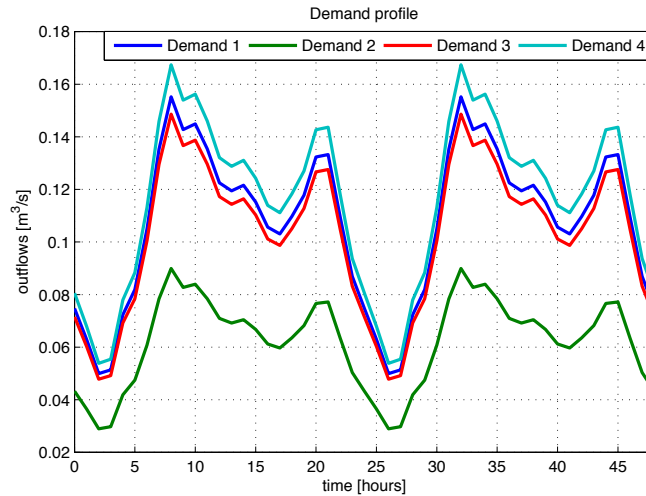
---

<sup>2</sup>Parameters of this academical problem have been scaled from real transportation of drinking water problems.



**Figure 5.6:** Simple drinking water system with a unique resource and unknown demands.

where  $h_i$  is the water level of the  $i$ th tank,  $\rho$  is the density of the fluid, and  $g$  is the gravity. The proposed example considers the case of 4 tanks, and the control objective is to maintain the water level of each tank at a safety value of reference, which is given by the company in charge of the management of the network. This safety value is obtained according to the demand that each tank supplies. For this particular example the safety reference is established at 0.5 m, and the unknown demand profile at each node during two days is shown in Figure 5.7 (adapted from [124]).



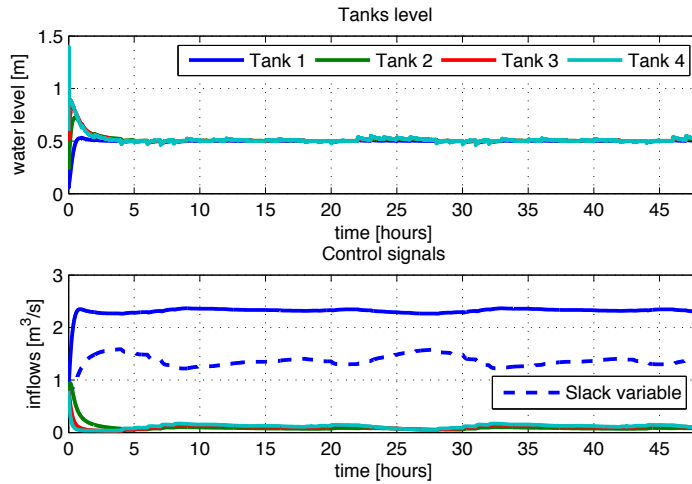
**Figure 5.7:** Demand profile ( $d_1, d_2, d_3$  and  $d_4$ ) during 2 days for the 4 tanks case.

To control this plant using population dynamics, we propose the following analogy: the population is related to the available water, the strategies correspond to the tanks. Therefore, the population state  $x$  represents the distribution of the water (in percentage) among the tanks. Additionally, the fitness functions are selected to be the error at each

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

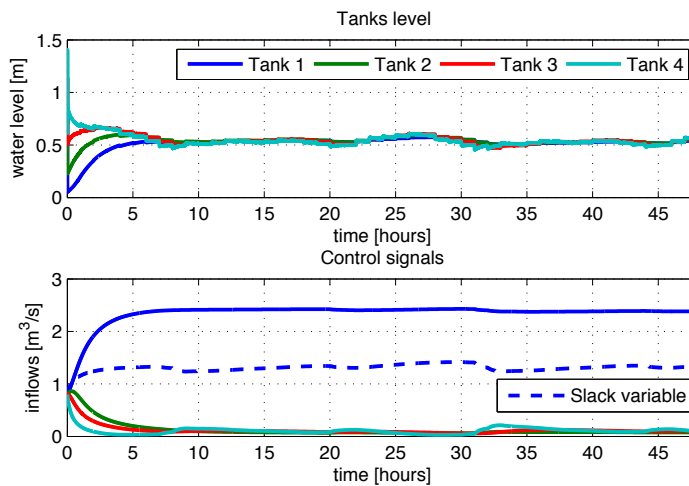
tank, i.e.,  $f_i = 0.5 - h_i$ , for all  $i = 1, \dots, n$ . Notice that this fitness is appropriate since: *i*) more proportion of inflow is assigned to those tanks with larger error, and *ii*) it is known that  $h_i$  increases as the inflow controlled by  $x_i$  increases, i.e., the fitness function  $f_i$  is decreasing with respect to  $x_i$  satisfying the condition for a stable game [3]. Furthermore, consider the case in which the quantity of water required to meet the safety levels is less than the available resource. In this situation, if all the available resource is allocated in the tanks, the safety levels would be exceeded. Therefore, it is necessary to introduce an additional strategy whose corresponding population  $x_{n+1}$  is used as a slack variable. The fitness function for this slack variable is  $f_{n+1} = 0$ . Hence, if the level in the  $i$ th tank is higher than the safety level (i.e.,  $f_i < 0$ ), it is more profitable to allocate resource to the slack variable than to the  $i$ th tank.

Figure 5.8 shows the control performance considering full information in the classic Smith dynamics, i.e., that at each point of the network, the information related to all the system is available to make decisions. Figure 5.9 shows the performance of a distributed



**Figure 5.8:** System states evolution for a controller with full information.

controller designed based on DSD. The information graph considered for this example is a path graph, i.e., that the  $i$ th tank only has information about the  $(i - 1)$ th and the  $(i + 1)$ th tanks. First, notice that in this particular problem with four control actions and one slack variable, the classical Smith dynamics with full information require 10 communication links. In contrast, the distributed population dynamics approach for a problem with four control actions, one slack variable, and a path configuration just



**Figure 5.9:** System states evolution for a distributed population dynamics based controller.

require four communication links.

Regarding control performance, results show that the control objective is achieved with both the Smith dynamics controller with full information and with the DSD controller by using partial information despite the hard and unknown outflow disturbances at each tank. Furthermore, it can be noticed that the settling time of the distributed controller is not far from the one obtained with full information. Additionally, once the system achieves the set-point, both controllers have similar behavior showing the well performance of the distributed population dynamics approach.

## 5.6 Discussion

This chapter proposes a methodology to generate distributed population dynamics from revision protocols considering different population structures. The core of the proposed method lies in a generalization of the mean dynamics for non-well-mixed populations. This novel concept provides us a tool to deal with information constraints related to the strategic interactions of the individuals that conform the population. Allowed agents' interactions within the population are described by means of an undirected graph. Following this convention, we have shown that well-mixed populations are characterized by complete graphs, while non-well-mixed populations are represented by non-complete graphs. The main implication of this feature is the possibility to derive population dynamics that only use local information to evolve, i.e., distributed population dynamics.

## 5. DISTRIBUTED POPULATION DYNAMICS<sup>1</sup>

---

The appealing features of distributed population dynamics have been exploited in some application examples that illustrate their usefulness. We have shown that distributed population dynamics can be used in the design of constrained optimization algorithms, and the synthesis of control systems for problems requiring non-centralized schemes. Notice that normal form games can also model similar engineering applications. Therefore, distributed learning strategies might be used to solve these problems. However, the authors in [125] suggest that this approach fails if the application includes a coupled constraint that involves all the decision variables. Under this scenario, distributed population dynamics become a proper alternative since their trajectories evolve to the optimal solution satisfying the referred coupled constraint as shown in Theorems 2–4.

Simulations have illustrated the equivalence between the steady state behavior of the proposed distributed dynamics and classic population dynamics. Moreover, regarding the transient performance, simulations have also shown that there exists a relationship between the convergence rate and the algebraic connectivity of the underlying communication graph. The larger the algebraic connectivity is, the faster the transient response of the corresponding dynamics. Indeed, the fastest response is obtained by using a complete-graph, i.e., by using classic population dynamics. Similarly to other distributed schemes, the dependence of the performance on the graph connectivity evidences a tradeoff between the number of communication links and how fast the optimal solution is reached.

## 6

# Conclusions and Future Directions

### 6.1 Distributed Resource Allocation Among Dynamic Networks

We have extended the applicability of consensus-based resource allocation algorithms from static problems to dynamic scenarios. We have proved that, if the systems that comprise the network are passive (either dynamical or memoryless), then a multi-agent controller that uses classic consensus protocols can drive the network to a desired global behavior, while a resource constraint is satisfied for any time. This result has been used to synthesize a distributed technique for controlling the temperature in the rooms of a building that uses a central heating system. We have proved by means of graph theoretical tools and invariance analysis, that the proposed controller is able to deal with several power constraints without the need of full information. Moreover, we have used concepts from passivity theory to show that, under connected graphs, our methodology optimally allocates the available heating power among the rooms, considering different scenarios, e.g., when the temperature references cannot be reached due to critical load conditions. In this situation we have used the variance of the steady state temperature errors as our performance index. Some simulations that show the advantages of the proposed method compared to other widely used techniques have been presented. Furthermore, we have analyzed static resource allocation problems from the perspective of passive memoryless systems. We have shown that convex problems have associated a passive memoryless

## 6. CONCLUSIONS AND FUTURE DIRECTIONS

---

function, which is given by the gradient of the corresponding cost function.

An interesting future direction is the extension of the results given in this dissertation to deal with directed and time-varying graphs. This extension is important since directed and time-varying graphs emerge in a number of real applications. For instance, in networks of mobile agents [31], where communications between agents vary depending on agents' relative positions. This proposed future direction is challenging since the analysis of distributed resource allocation algorithms highly depends on the graph structure.

### 6.2 Optimal Resource Allocation with Lower-Bound Constraints

We have developed a distributed method that solves a class of resource allocation problems with lower bound constraints. The proposed approach is based on a multi-agent system, where coordination among agents is done by using a consensus protocol. We have proved that convergence and optimality of the method is guaranteed under some mild assumptions, specifically, we require that the cost function is strictly convex and the graph related to the communication network that enables the agents to share information is connected. The main advantage of our technique is that it does not need a centralized coordinator, which makes the method appropriate to be applied in large-scale distributed systems, where the inclusion of centralized agents is undesirable or infeasible. As future work, we propose to use a switched approach in order to eliminate the iterations in Algorithm 1. Moreover, we plan to include upper bound constraints in our original formulation.

### 6.3 Distributed Population Dynamics

We have generalized a methodology to generate distributed population dynamics from the distributed mean dynamics and different revision protocols. To illustrate the method, we have derived four distributed population dynamics, i.e., the distributed Smith dynamics, the distributed replicator dynamics, the distributed projection dynamics and the distributed logit dynamics. We have proved that the distributed population dynamics presented in this work exhibit mass conservation and convergence to Nash equilibria in the same way as classic population dynamics do. Finally, we have presented



some applications of the proposed dynamics in the design of distributed optimization algorithms, solution of classic games, and the synthesis of controllers for multi-variable system. Simulation results have shown that distributed population dynamics exhibit an appropriate performance and provide optimal solutions despite the lack of full information. Thus, distributed population dynamics are promising to address problems requiring non-centralized information structures. This fact is relevant since distributed problems are becoming more common in many fields, especially when large-scale and complex systems are involved.

A future research direction is the inclusion of dynamics in the fitness functions (in a similar way as in Chapter 3). We notice that this modification would give more flexibility to distributed population dynamics for dealing with a wider range of problems.

## 6. CONCLUSIONS AND FUTURE DIRECTIONS

---

# References

- [1] J BARREIRO-GÓMEZ, G OBANDO, AND N QUIJANO. **Distributed Population Dynamics: Optimization and Control Applications.** *Submitted to the IEEE Transactions on Systems, Man, and Cybernetics: Systems.* iii, 81
- [2] Y.C. CHANG AND W.H. CHEN. **Optimal chilled water temperature calculation of multiple chiller systems using Hopfield neural network for saving energy.** *Energy*, **34**(4):448–456, 2009. v, 75, 76, 77, 78
- [3] WILLIAM H SANDHOLM. *Population games and evolutionary dynamics.* MIT press, 2010. vii, 81, 82, 84, 85, 88, 91, 99, 102
- [4] C. SCHNEEWEISS. *Distributed decision making.* Springer, 2003. 1
- [5] DIMITRI P BERTSEKAS AND JOHN N TSITSIKLIS. *Parallel and distributed computation: Numerical methods.* Athena Scientific, 1997. 1
- [6] DIMITRI P BERTSEKAS. *Network optimization: continuous and discrete models.* Athena Scientific, 1998. 1
- [7] A. BEMPORAD, M. HEEMELS, AND M. JOHANSSON. *Networked control systems.* Springer, 2010. 1, 2, 51
- [8] JORGE CORTES AND SONIA MARTINEZ. **Distributed line search via dynamic convex combinations.** In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2346–2351. IEEE, 2013. 1
- [9] MINGHUI ZHU AND SONIA MARTÍNEZ. **An approximate dual subgradient algorithm for multi-agent non-convex optimization.** *Automatic Control, IEEE Transactions on*, **58**(6):1534–1539, 2013. 1
- [10] JOHN C DUCHI, ALEKH AGARWAL, AND MARTIN J WAINWRIGHT. **Dual averaging for distributed optimization: convergence analysis and network scaling.** *Automatic control, IEEE Transactions on*, **57**(3):592–606, 2012. 1
- [11] MICHAEL PATRIKSSON. **A survey on the continuous nonlinear resource allocation problem.** *European Journal of Operational Research*, **185**(1):1–46, 2008. 1, 51, 57, 60

## REFERENCES

---

- [12] JON M CONRAD. *Resource economics*. Cambridge University Press, 2010. 1, 5, 59
- [13] ANDRÉS PANTOJA AND NICANOR QUIJANO. **A population dynamics approach for the dispatch of distributed generators**. *IEEE Transactions on Industrial Electronics*, **58**(10):4559–4567, 2011. 1
- [14] RIKY SUBRATA, ALBERT Y ZOMAYA, AND BJORN LANDFELDT. **Game-theoretic approach for load balancing in computational grids**. *IEEE Transactions on Parallel and Distributed Systems*, **19**(1):66–76, 2008. 1
- [15] DE-NIAN YANG AND MING-SYAN CHEN. **Efficient resource allocation for wireless multicast**. *IEEE Transactions on Mobile Computing*, **7**(4):387–400, 2008. 1, 5, 59
- [16] EDUARDO RAMIREZ-LLANOS AND SONIA MARTINEZ. **Distributed and robust resource allocation algorithms for multi-agent systems via discrete-time iterations**. *To appear in proceedings of the American Control Conference*, 2015. 1, 6, 59
- [17] DANIEL P PALOMAR AND MUNG CHIANG. **A tutorial on decomposition methods for network utility maximization**. *IEEE Journal on Selected Areas in Communications*, **24**(8):1439–1451, 2006. 1, 6
- [18] SUVRIT SRA, SEBASTIAN NOWOZIN, AND STEPHEN J WRIGHT. *Optimization for machine learning*. MIT Press, 2011. 2
- [19] FREDRIK YGGE AND HANS AKKERMANS. **Decentralized Markets versus Central Control: A Comparative Study**. *Journal of Artificial Intelligence Research*, **11**:301–333, 1999. 2, 4, 41
- [20] BO AN AND VICTOR LESSER. **Characterizing contract-based multiagent resource allocation in networks**. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **40**(3):575–586, 2010. 2
- [21] JASON R MARDEN. **State based potential games**. *Automatica*, 2012. 2
- [22] ANDRES PANTOJA AND NICANOR QUIJANO. **Distributed optimization using population dynamics with a local replicator equation**. In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3790–3795. IEEE, 2012. 2, 3, 5, 6, 22, 71, 74
- [23] WALID SAAD, ZHU HAN, AND H VINCENT POOR. **Coalitional game theory for cooperative micro-grid distribution networks**. In *Proceedings of the IEEE International Conference on Communications Workshops*, pages 1–5. IEEE, 2011. 2
- [24] YC HO, L SERVI, AND R SURI. **A class of center-free resource allocation algorithms**. *Large Scale Systems*, **1**(1):51–62, 1980. 2, 3, 4, 51, 57

- 
- [25] MINGHUI ZHU AND SONIA MARTINEZ. **On distributed convex optimization under inequality and equality constraints.** *IEEE Transactions on Automatic Control*, **57**(1):151–164, 2012. 2
- [26] R. OLFATI-SABER, J.A. FAX, AND R.M. MURRAY. **Consensus and cooperation in networked multi-agent systems.** *Proceedings of the IEEE*, **95**(1):215–233, 2007. 2, 94
- [27] J. CORTÉS. **Distributed algorithms for reaching consensus on general functions.** *Automatica*, **44**(3):726–737, 2008. 2, 15, 23, 67, 71
- [28] LUC MOREAU. **Stability of multiagent systems with time-dependent communication links.** *Automatic Control, IEEE Transactions on*, **50**(2):169–182, 2005. 2, 3, 23, 94
- [29] YONGCAN CAO, WENWU YU, WEI REN, AND GUANRONG CHEN. **An overview of recent progress in the study of distributed multi-agent coordination.** *Industrial Informatics, IEEE Transactions on*, **9**(1):427–438, 2013. 2
- [30] ALEX OLSHEVSKY AND JOHN N TSITSIKLIS. **Convergence speed in distributed consensus and averaging.** *SIAM Journal on Control and Optimization*, **48**(1):33–55, 2009. 2
- [31] A JADABABAIE, J LIN, AND AS MORSE. **Coordination of groups of mobile autonomous agents using nearest neighbour rules.** *IEEE Transactions on Automatic Control*, **48**(6):988–1001, 2003. 3, 106
- [32] R. OLFATI-SABER AND R. MURRAY. **Consensus problems in networks of agents with switching topology and time-delays.** *IEEE Transactions on Automatic Control*, **49**(9):1520–1533, 2004. 3, 5, 23, 40
- [33] WEI REN AND RW BEARD. **Consensus seeking in multiagent systems under dynamically changing interaction topologies.** *IEEE Transactions on Automatic Control*, **50**(5):655–661, 2005. 3, 94
- [34] SOLMAZ S KIA, JORGE CORTÉS, AND SONIA MARTINEZ. **Dynamic average consensus under limited control authority and privacy requirements.** *To appear in International Journal of Robust and Nonlinear Control*. 3
- [35] LIN XIAO AND STEPHEN BOYD. **Optimal scaling of a gradient method for distributed resource allocation.** *Journal of optimization theory and applications*, **129**(3):469–488, 2006. 3, 4, 6, 23, 51, 57, 67, 71
- [36] ASHISH CHERUKURI AND JORGE CORTÉS. **Distributed generator coordination for initialization and anytime optimization in economic dispatch.** *To appear in IEEE Transactions on Control of Network Systems*. 3, 6, 23, 51, 57, 59, 71, 73, 74

## REFERENCES

---

- [37] BJÖRN JOHANSSON AND MIKAEL JOHANSSON. **Distributed non-smooth resource allocation over a network.** In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 1678–1683. IEEE, 2009. 3, 6
- [38] GIULIO BINETTI, ALI DAVOUDI, DAVID NASO, BIAGIO TURCHIANO, AND FRANK L LEWIS. **A distributed auction-based algorithm for the nonconvex economic dispatch problem.** *IEEE Transactions on Industrial Informatics*, **10**(2):1124–1132, 2014. 3
- [39] G. OBANDO, A. PANTOJA, AND N. QUIJANO. **Building Temperature Control based on Population Dynamics.** *IEEE Transactions on Control Systems Technology*, **22**(1):404–412, 2014. 3, 22, 82, 95
- [40] EDUARDO RAMÍREZ-LLANOS AND NICANOR QUIJANO. **A population dynamics approach for the water distribution problem.** *International Journal of Control*, **83**(9):1947–1964, 2010. 3, 22
- [41] JORGE POVEDA AND NICANOR QUIJANO. **Dynamic bandwidth allocation in wireless networks using a Shahshahani gradient based extremum seeking control.** In *proceedings of the 2012 6th International Conference on Network Games, Control and Optimization (NetGCooP)*, pages 44–50. IEEE, 2012. 3, 22, 95
- [42] H. TEMBINE, E. ALTMAN, R. EL-AZOUZI, AND Y. HAYEL. **Evolutionary games in wireless networks.** *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **40**(3):634–646, 2010. 3, 22, 82, 95
- [43] HAMIDOU TEMBINE, EITAN ALTMAN, RACHID EL-AZOUZI, AND YEZEKAEEL HAYEL. **Bio-inspired delayed evolutionary game dynamics with networking applications.** *Telecommunication Systems*, **47**(1-2):137–152, 2011. 3, 22
- [44] EDUARDO MOJICA-NAVA, CARLOS ANDRÉS MACANA, AND NICANOR QUIJANO. **Dynamic population games for optimal dispatch on hierarchical microgrid control.** *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, **44**(3):306–317, 2014. 3, 22, 95
- [45] J BARREIRO-GÓMEZ, G OBANDO, G RIANO-BRICENO, N QUIJANO, AND C OCAMPO-MARTINEZ. **Decentralized Control for Urban Drainage Systems Via Population Dynamics: Bogotá Case Study.** *To appear in proceedings of the 14th European Control Conference*, 2015. 3, 22
- [46] ZHONGKUI LI, ZHISHENG DUAN, GUANRONG CHEN, AND LIN HUANG. **Consensus of multiagent systems and synchronization of complex networks: a unified viewpoint.** *IEEE Transactions on Circuits and Systems I: Regular Papers*, **57**(1):213–224, 2010. 4, 5

- 
- [47] WEI REN AND YONGCAN CAO. *Distributed coordination of multi-agent networks: emergent problems, models, and issues*. Springer Science & Business Media, 2010. 4, 31
- [48] JIANXIANG XI, ZHE XU, GUANGBIN LIU, AND YISHENG ZHONG. **Stable-protocol output consensus for high-order linear swarm systems with time-varying delays**. *IET Control Theory & Applications*, **7**(7):975–984, 2013. 4, 5
- [49] NIKHIL CHOPRA. **Output synchronization on strongly connected graphs**. *IEEE Transactions on Automatic Control*, **57**(11):2896–2901, 2012. 4, 5
- [50] JIANXIANG XI, ZONGYING SHI, AND YISHENG ZHONG. **Output consensus analysis and design for high-order linear swarm systems: partial stability method**. *Automatica*, **48**(9):2335–2343, 2012. 4
- [51] JI XIANG, YANJUN LI, AND WEI WEI. **Synchronised output regulation of nonlinear multi-agent systems**. *International Journal of Control*, **88**(1):54–64, 2015. 4
- [52] CUI-QIN MA AND JI-FENG ZHANG. **Necessary and sufficient conditions for consensusability of linear multi-agent systems**. *IEEE Transactions on Automatic Control*, **55**(5):1263–1268, 2010. 4
- [53] ZHONGKUI LI AND ZHISHENG DUAN. *Cooperative Control of Multi-agent Systems: A Consensus Region Approach*. CRC Press, 2014. 4
- [54] GERMAN OBANDO, ANDRES PANTOJA, AND NICANOR QUIJANO. **Building temperature control based on population dynamics**. *IEEE Transactions on Control Systems Technology*, **22**(1):404–412, 2014. 4, 6, 59
- [55] JULIAN BARREIRO-GÓMEZ, N QUIJANO, AND CARLOS OCAMPO-MARTÍNEZ. **Constrained distributed optimization based on population dynamics**. In *IEEE 53rd Annual Conference on Decision and Control (CDC)*, pages 4260–4265. IEEE, 2014. 4
- [56] JORGE I POVEDA AND NICANOR QUIJANO. **Shahshahani gradient-like extremum seeking**. *Automatica*, **58**:51–59, 2015. 4, 95
- [57] GERMÁN OBANDO, JORGE I POVEDA, AND NICANOR QUIJANO. **Replicator Dynamics under Perturbations and Time-Delays**. *Submitted to Mathematics of Control, Signals, and Systems*. 4
- [58] YUJI WAKASA, MIZUE ARAKAWA, KANYA TANAKA, AND TAKUYA AKASHI. **Decentralized model predictive control via dual decomposition**. In *47th Conference on Decision and Control*, pages 381–386. IEEE, 2008. 4
- [59] MATO BAOTIĆ AND VEDRANA SPUDIĆ. **Fast Coordinated Model Predictive Control of Large-Scale Distributed Systems with Single Coupling Constraint**. In *13th European Control Conference*, pages 2783–2788. IEEE, 2013. 4

## REFERENCES

---

- [60] ANTOINE LEFORT, ROMAIN BOURDAIS, GUILLAUME ANSANAY-ALEX, AND HERVÉ GUÉGUEN. **Hierarchical control method applied to energy management of a residential house.** *Energy and Buildings*, pages 53–61, 2013. 4, 34
- [61] S.H. CLEARWATER, R. COSTANZA, M. DIXON, AND B. SCHROEDER. **Saving energy using market-based control.** *Market-Based Control: A Paradigm for Distributed Resource Allocation*, pages 253–273, 1996. 4, 34, 41
- [62] FREDRIK YGGE AND HANS AKKERMANS. **Power load management as a computational market.** In *Proceedings of the 2nd International Conference on Multi-Agent Systems*, pages 393–400, 1996. 4, 55
- [63] ETTORE F BOMPARD AND BEI HAN. **Market-based control in emerging distribution system operation.** *IEEE Transactions on Power Delivery*, **28**(4):2373–2382, 2013. 4
- [64] JOHN TSITSIKLIS, DIMITRI BERTSEKAS, AND MICHAEL ATHANS. **Distributed asynchronous deterministic and stochastic gradient optimization algorithms.** *IEEE Transactions on Automatic Control*, **31**(9):803–812, 1986. 4
- [65] J.L. WYATT, L.O. CHUA, J. GANNETT, I.C. GOKNAR, AND D. GREEN. **Foundations of Nonlinear Network Theory, Part I: Passivity.** *Memorandum UCB/ERL M78/76, Electronics Research Laboratory, University of California at Berkeley, 1978.* 5
- [66] JAN C WILLEMS. **Dissipative dynamical systems part I: General theory.** *Archive for rational mechanics and analysis*, **45**(5):321–351, 1972. 5, 16
- [67] NIKHIL CHOPRA AND MARK W SPONG. **Passivity-based control of multi-agent systems.** In *Advances in robot control*, pages 107–134. Springer, 2006. 5, 21, 31, 32, 33
- [68] MURAT ARCAK. **Passivity as a design tool for group coordination.** *IEEE Transactions on Automatic Control*, **52**(8):1380–1390, 2007. 5, 25, 31
- [69] HE BAI, MURAT ARCAK, AND JOHN WEN. *Cooperative control design: a systematic, passivity-based approach.* Springer Science & Business Media, 2011. 5, 31
- [70] AD DOMINGUEZ-GARCIA AND CN HADJICOSTIS. **Distributed algorithms for control of demand response and distributed energy resources.** In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 27–32, 2011. 5, 53, 55
- [71] ANDRÉS PANTOJA, NICANOR QUIJANO, AND KEVIN M PASSINO. **Dispatch of distributed generators under local-information constraints.** In *Proceedings of the American Control Conference*, pages 2682–2687. IEEE, 2014. 6, 59
- [72] S. TAN, S. YANG, AND J. XU. **Consensus based approach for economic dispatch problem in a smart grid.** *IEEE Transactions on Power Systems*, **28**(4):4416–4426, 2013. 6, 59



- 
- [73] DAMON MOSK-AOYAMA, TIM ROUGHGARDEN, AND DEVAVRAT SHAH. **Fully distributed algorithms for convex optimization problems.** *SIAM Journal on Optimization*, **20**(6):3260–3279, 2010. 6
- [74] PAUL A JENSEN AND JONATHAN F BARD. *Operations research models and methods*. John Wiley & Sons Incorporated, 2003. 6, 59
- [75] ALEJANDRO D DOMINGUEZ-GARCIA, STANTON T CADY, AND CHRISTOFOROS N HADJICOSTIS. **Decentralized optimal dispatch of distributed energy resources.** In *Proceedings of the 51st IEEE Conference on Decision and Control*, pages 3688–3693. IEEE, 2012. 6
- [76] A. NEDIC, A. OZDAGLAR, AND P.A. PARRILO. **Constrained consensus and optimization in multi-agent networks.** *IEEE Transactions on Automatic Control*, **55**(4):922–938, 2010. 6
- [77] PENG YI, YIGUANG HONG, AND FENG LIU. **Distributed gradient algorithm for constrained optimization with application to load sharing in power systems.** *Systems & Control Letters*, **83**:45–52, 2015. 6
- [78] S. SHARMA AND D. TENEKETZIS. **An externalities-based decentralized optimal power allocation algorithm for wireless networks.** *IEEE/ACM Transactions on Networking (TON)*, **17**(6):1819–1831, 2009. 7
- [79] A. KAKHBOD AND D. TENEKETZIS. **Power allocation and spectrum sharing in multi-user, multi-channel systems with strategic users.** *IEEE Transactions on Automatic Control*, **57**(9):2338–2342, 2012. 7
- [80] W.S. LEE, Y.T. CHEN, AND Y. KAO. **Optimal chiller loading by differential evolution algorithm for reducing energy consumption.** *Energy and Buildings*, **43**(2):599–604, 2011. 7
- [81] A. PANTOJA AND N. QUIJANO. **Distributed optimization using population dynamics with a local replicator equation.** In *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, pages 3790–3795, 2012. 10, 81, 82, 97
- [82] C. GODSIL AND G. ROYLE. *Algebraic graph theory, volume 207 of Graduate Texts in Mathematics*. Springer-Verlag, 2001. 14
- [83] HASSAN K KHALIL. *Nonlinear systems*, **3**. Prentice Hall, 2002. 16, 17, 33
- [84] JOHN TSITSIKLIS. **Problems in Decentralized Decision making and Computation.** Technical report, DTIC Document, 1984. 20
- [85] JIANXIANG XI, ZHE XU, GUANGBIN LIU, AND YISHENG ZHONG. **Stable-protocol output consensus for high-order linear swarm systems with time-varying delays.** *IET Control Theory & Applications*, **7**(7):975–984, 2013. 31

## REFERENCES

---

- [86] JIANXIANG XI, ZONGYING SHI, AND YISHENG ZHONG. **Output consensus analysis and design for high-order linear swarm systems: partial stability method.** *Automatica*, **48**(9):2335–2343, 2012. 31
- [87] WASSIM M HADDAD AND VIJAYSEKHAR CHELLABOINA. *Nonlinear dynamical systems and control: a Lyapunov-based approach.* Princeton University Press, 2008. 32
- [88] FERREIDOOON P SIOSHANSI. *Energy Efficiency: Towards the End of Demand Growth.* Academic Press, access Online via Elsevier, 2013. 34
- [89] MAHMOUD ALAHMAD, EVANS SORDIASHIE, LANA EL CHAAR, NASSER ALJUHAISHI, AND KUWAIT HAMID SHARIF. **Information Technology and the Smart Grid-A Pathway to Conserve Energy in Buildings.** In *Proceedings of the 9th International Conference on Innovations in Information Technology*, pages 60–65. IEEE, 2013. 34
- [90] S.Y. NOF. *Springer Handbook of Automation.* Springer Publishing Company, Incorporated, 2009. 34, 49
- [91] YUDONG MA, ANTHONY KELMAN, ALLAN DALY, AND FRANCESCO BORRELLI. **Predictive control for energy efficient buildings with thermal storage: Modeling, stimulation, and experiments.** *IEEE Control Systems Magazine*, **32**(1):44–64, 2012. 34
- [92] SIDDHARTH GOYAL, HERBERT A INGLEY, AND PRABIR BAROOAH. **Occupancy-based zone-climate control for energy-efficient buildings: Complexity vs. performance.** *Applied Energy*, **106**:209–221, 2013. 34
- [93] PENG ZHAO, SIDDHARTH SURYANARAYANAN, AND MARCELO GODOY SIMÕES. **An energy management system for building structures using a multi-agent decision-making control methodology.** *IEEE Transactions on Industry Applications*, **49**(1):322–330, 2013. 34
- [94] N. LONG, E. BONNEMA, K. FIELD, AND P. TORCELLINI. **Evaluation of ANSI/ASHRAE/USGBC/IES Standard 189.1-2009.** Technical Report NREL/TP-550-47906. 34
- [95] G. OBANDO, A. PANTOJA, AND N. QUIJANO. **Building Temperature Control based on Population Dynamics.** To appear in *IEEE Transactions on Control Systems Technology*. DOI: 10.1109/TCST.2013.2252175. 34, 41
- [96] MEHDI MAASOUMY, ALESSANDRO PINTO, AND ALBERTO SANGIOVANNI-VINCENTELLI. **Model-based hierarchical optimal control design for HVAC systems.** In *Proceedings of the Dynamic System Control Conference.* ASME, 2011. 35

- 
- [97] GERMAN OBANDO, NICANOR QUIJANO, AND ANDRES PANTOJA. **Population dynamics applied to building energy efficiency.** In *Proceedings of the 50th IEEE Conference on Decision and Control*, pages 4777–4782, 2011. 35
- [98] G. OBANDO, N. QUIJANO, AND N. RAKOTO-RAVALONTSALAMA. **Distributed Building Temperature Control with Power Constraints.** In *Proceedings of the 13th European Control Conference*, pages 2857–2862, 2014. 38
- [99] J. NOCEDAL AND S.J. WRIGHT. *Numerical optimization.* Springer verlag, 1999. 43, 52
- [100] PETRU-DANIEL MOROȘAN, ROMAIN BOURDAIS, DIDIER DUMUR, AND JEAN BUISSON. **A distributed MPC strategy based on Benders’ decomposition applied to multi-source multi-zone temperature regulation.** *Journal of Process Control*, **21**(5):729–737, 2011. 50
- [101] STEPHEN BOYD AND LIEVEN VANDENBERGHE. *Convex optimization.* Cambridge university press, 2004. 52
- [102] GERMÁN OBANDO, NICANOR QUIJANO, AND NALY RAKOTO-RAVALONTSALAMA. **Distributed Resource Allocation Over Stochastic Networks: An Application in Smart Grids.** *To appear in proceedings of the 2nd IEEE Colombian Conference on Automatic Control (CCAC).* 54
- [103] F. GARIN AND L. SCHENATO. **A survey on distributed estimation and control applications using linear consensus algorithms.** In *Networked Control Systems*, pages 75–107. Springer, 2010. 67
- [104] DIMITRI P BERTSEKAS. **Necessary and sufficient conditions for a penalty method to be exact.** *Mathematical programming*, **9**(1):87–99, 1975. 72
- [105] PETER. D. TAYLOR AND LEO. B. JONKER. **Evolutionary Stable Strategies and Game Dynamics.** *Mathematical Biosciences*, **40**:145–156, 1978. 81
- [106] J. W. WEIBULL. *Evolutionary Game Theory.* The MIT Press, London, England, 1997. 81, 82
- [107] J. HOFBAUER AND K. SIGMUND. *Evolutionary games and population dynamics.* Cambridge University Press, Cambridge, 1998. 81
- [108] M. J. FOX AND J. S. SHAMMA. **Population games, stable games, and passivity.** In *Proceedings of the IEEE 51st Conference on Decision and Control (CDC)*, pages 7445–7450, 2012. 81
- [109] J. MARDEN AND J. S. SHAMMA. **Game Theory and Distributed Control.** In H.P. YOUNG AND S. ZAMIR, editors, *Handbook of Game Theory Vol. 4*, pages 861–900. North Holland, 2014. 82

## REFERENCES

---

- [110] S. MANNOR AND J. S. SHAMMA. **Multi-agent learning for engineers.** *Artificial Intelligence, special issue on Foundations of Multi-Agent Learning*, **171**(7):417–422, May 2007. 82
- [111] N. LI AND J. MARDEN. **Designing games for distributed optimization with a time varying communication graph.** In *Proceedings of the IEEE 51st Conference on Decision and Control (CDC)*, pages 7764 – 7769, 2012. 82
- [112] J. BARREIRO-GOMEZ, N. QUIJANO, AND C. OCAMPO-MARTINEZ. **Constrained distributed optimization based on population dynamics.** In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 4260–4265, 2014. 82
- [113] SIEGFRIED BERNINGHAUS AND HANS HALLER. **Local Interaction on Random Graphs.** *Games*, **1**(3):262–285, 2010. 82
- [114] CARLOS ALÓS-FERRER AND SIMON WEIDENHOLZER. **Imitation, local interactions, and efficiency.** *Economics Letters*, **93**:163–168, 2006. 82
- [115] B. GHARESIFARD AND J. CORTES. **Distributed convergence to Nash equilibria by adversarial networks with directed topologies.** In *Proceedings of the American Control Conference*, pages 5881–5886, 2012. 82
- [116] O. BOUSSATON AND J. COHEN. **On the distributed learning of Nash equilibria with minimal information.** In *Proceedings of the 6th International Conference on Network Games, Control, and Optimization*, pages 30–37, 2012. 82
- [117] J. HOFBAUER AND W. H. SANDHOLM. **Stable games and their dynamics.** *Journal of Economic Theory*, **144**(4):1665–1693, 2009. 84
- [118] IMMANUEL BOMZE, MARCELLO PELILLO, AND VOLKER STIX. **Approximating the maximum weight clique using replicator dynamics.** *IEEE Transactions on Neural Networks*, **11**(6):1228–1241, 2000. 95
- [119] A. PANTOJA AND N. QUIJANO. **A Population Dynamics Approach for the Dispatch of Distributed Generators.** *IEEE Transactions on Industrial Electronics*, **58**(10):4559–4567, 2011. 95
- [120] J. BARREIRO-GOMEZ, N. QUIJANO, AND C. OCAMPO-MARTINEZ. **Distributed control of drinking water networks using population dynamics: Barcelona case study.** In *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pages 3216–3221, 2014. 95
- [121] S. BORNHOLDT AND H. G. SCHUSTER. *Handbook of graphs and networks*, **2**. Wiley Online Library, 2003. 96

- [122] J. R. MARDEN, G. ARSLAN, AND J. S. SHAMMA. **Cooperative Control and Potential Games.** *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, **39**:1393–1407, December 2009. 100
- [123] JING WANG AND NICOLA ELIA. **A control perspective for centralized and distributed convex optimization.** In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 3800–3805, 2011. 100
- [124] C. OCAMPO-MARTINEZ, D. BARCELLI, V. PUIG, AND A. BEMPORAD. **Hierarchical and decentralised model predictive control of drinking water networks: application to Barcelona case study.** *IET Control Theory and Applications*, **6**(1):62, January 2012. 101
- [125] J. R. MARDEN. **State based potential games.** *Automatica*, **48**(12):3075–3088, 2012. 104

# Thèse de Doctorat

Germán Darío OBANDO BRAVO

## Distributed Methods for Resource Allocation: A Passivity-Based Approach

Métodos Distribuidos para Asignación de Recursos: Un Enfoque basado en Pasividad

### Abstract

Since the complexity and scale of systems have been growing in the last years, distributed approaches for control and decision making are becoming more prevalent. This dissertation focuses on an important problem involving distributed control and decision making, the dynamic resource allocation in a network. To address this problem, we explore a consensus-based algorithm that does not require any centralized computation, and that is capable to deal with applications modeled either by dynamical systems or by memoryless functions. The main contribution of our research is to prove, by means of graph theoretical tools and passivity analysis, that the proposed controller asymptotically reaches an optimal solution without the need of full information.

In order to illustrate the relevance of our main result, we address several engineering applications including: distributed control for energy saving in smart buildings, management of the customers of an aggregating entity in a smart grid environment, and development of an exact distributed optimization method that deals with resource allocation problems subject to lower-bound constraints.

Finally, we explore resource allocation techniques based on classic population dynamics models. In order to make them distributed, we introduce the concept of non-well-mixed population dynamics. We show that these dynamics are capable to deal with constrained information structures that are characterized by non-complete graphs. Although the proposed non-well-mixed population dynamics use partial information, they preserve similar properties of their classic counterpart, which uses full information. Specifically, we prove mass conservation and convergence to Nash equilibrium.

### Key Words

Resource allocation, output consensus, distributed algorithms, graph theory, passivity, multi-agent systems, population dynamics.

### Resumen

Dado que la complejidad y la escala de los sistemas se han ido incrementando en los últimos años, las técnicas centralizadas de control y toma de decisiones están siendo reemplazadas por métodos distribuidos. Esta tesis se centra en un importante problema que involucra control y toma de decisiones distribuidas: la asignación dinámica de recursos en redes. Para abordar este problema, exploramos un algoritmo basado en consenso que no requiere computación centralizada, y que puede ser usado en aplicaciones modeladas ya sea por sistemas dinámicos o funciones sin memoria. La principal contribución de esta tesis es probar, por medio de teoría de grafos y pasividad, que el algoritmo propuesto alcanza asintóticamente una solución óptima sin la necesidad de usar información completa.

Para ilustrar la relevancia del resultado principal de esta disertación, abordamos varias aplicaciones en ingeniería, incluyendo: el control distribuido en edificios inteligentes orientado a la eficiencia energética, la gestión de los clientes de un agregador en una red inteligente en la que se aplican estrategias de respuesta de la demanda, y el desarrollo de un método de optimización exacto que permite incluir restricciones de límite inferior.

Finalmente, se exploran otras técnicas de asignación de recursos inspiradas en modelos de dinámicas poblacionales. Se introduce el concepto de poblaciones no-bien-mezcladas, y se muestra que las dinámicas asociadas a este tipo de poblaciones cuentan con una estructura de información local, caracterizada por grafos que no son completos. A pesar de que las dinámicas propuestas usan información parcial, ellas preservan características similares a las dinámicas poblacionales clásicas que usan información completa.

### Palabras Clave

Asignación de recursos, consenso, algoritmos distribuidos, teoría de grafos, pasividad, sistemas multi-agente, dinámicas poblacionales.