Distributed Transactive Control in Distribution Systems with Microgrids

Eder David Baron Prada

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, D. C., Colombia
Año 2018

Distributed Transactive Control in Distribution Systems with Microgrids

Eder David Baron Prada

Thesis presented as a partial requirement for the degree of:
Master in Industrial Automation

Advised by:
Ph.D., Eduardo Mojica Nava

Research Field:
Smart Grids
Research Group:
Programa de Análisis y Adquisición de Señales (PAAS)

Universidad Nacional de Colombia
Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, D. C., Colombia

Año 2018

Dedicated to Tili.

Dream big, plan well, work hard and things start to happen.

Acknowledgments

This journey is not possible without the support of people such as friends, family and a long list. If I don't include someone on this list, I apologize.

First of all, I want to thank my girlfriend Ana Torres, her invaluable collaboration and support was decisive for me. She always was when this road was difficult, when I needed support, she was for me once again.

Second, I want to thank my parents, for all their patience, for feeding me when I am hungry and for supporting me at all times. As a special mention, I would like to thank my little brother, who always gave all his support to finish this master's degree.

Third, I want to thank my friends: Saida Cortés, David Martinez, Alejandro Latorre and Jeisson Vergara who were always an incredible support for me and always were when I need advice. I would also like to thank the doctoral students Duvan Téllez and Fabián Salazar who always had words of support and encouragement for any difficulty.

Furthermore, I want to thank my advisor, Eduardo Mojica, who always supported me unconditionally, with enormous patience and from whom I can only be grateful for reading every word that was written, correcting it and giving feedback that was always useful to me.

Finally, I want to thank the National University of Colombia, for all its support, for the teaching assistant scholarship, which allowed me to dedicate myself to research in this master's degree full-time. I also thank the students that I had during this period of time, being a professor for two and a half years is something that fills my life very much. I would especially like to thank John Herrera, whom I had the good fortune to co-direct the final undergraduate work, and with this I was able to learn the difficult task of being someone's advisor and also how rewarding it is.

Title in English

Distributed Transactive Control in Distribution Systems with Microgrids

Título en español

Control distribuido basado en mercados en sistemas de distribución con microrredes

Abstract

Microgrids are considered as a cornerstone in the evolution to a smarter grid. However, this evolution brings some critical challenges for the control in a real-time implementation. We present two control algorithms to operate a power system with microgrids and other two to operate microgrids in order to reach the optimal social welfare. We consider three types of agents: photovoltaic generators, conventional generators and smart loads. These agents can be aggregated into a microgrid or interact directly in the power system depending on their power. To optimize the microgrids, we use two strategies. First one is based on projected consensus algorithm, where each agent iteratively optimizes its local utility function based on local information obtained from its neighbors and global information obtained through a distributed finite-time average algorithm. The second one is based on populations game theory; specifically we use a centralized replicator dynamics where a central agent iteratively optimizes the system status. To optimize the whole power system we use two strategies, first an asynchronous algorithm based on primal-dual optimization is proposed, where we consider that agents update the primal variables and a ”virtual agent” updates the dual variables. Our last algorithm is a distributed transactive control algorithm based on populations games to dynamically manage the distributed generators and smart loads in the system to reach the optimum social welfare. Agents are considered non-cooperative, and they are individually incentive-driven. The proposed algorithm preserve stability while guarantee optimality conditions considering several constraints in the system on the real-time operation. We show numerical results of the proposed control strategies.

Keywords:
Distributed control, distributed optimization, transactive control, projection algorithms, asynchronous algorithms, game theory, microgrids.

Resumen

Las microrredes están consideradas como la piedra angular de la evolución hacia una red más inteligente. Sin embargo, esta evolución trae consigo algunos retos importantes para el control en la implementación en tiempo real. Presentamos dos algoritmos de control para operar un sistema de energía con microrredes y otros dos para operar microrredes con el fin de alcanzar el bienestar social óptimo.

Consideramos dos tipos de agentes: generadores convencionales y cargas inteligentes. Estos agentes pueden ser agregados en una microred o interactuar directamente en el sistema de energía dependiendo de su potencia. Para optimizar las microrredes utilizamos dos estrategias, la primera se basa en un algoritmo de consenso proyectado, donde cada una de ellas optimiza iterativamente su función de utilidad local a partir de la información local obtenida de sus vecinos y la información global obtenida a través de un algoritmo distribuido de tiempo finito promedio. El segundo se basa en la teoría de juegos de poblaciones, específicamente usamos una dinámica de replicador centralizada donde un agente central optimiza iterativamente el estado del sistema. Para optimizar todo el sistema de potencia utilizamos dos estrategias, la primera es proponer un algoritmo asíncrono basado en la optimización prima-dual, donde consideramos que los agentes actualizan las variables primarias y un ”agente virtual” actualiza las variables duales. Nuestro último algoritmo es un algoritmo de control transaccional distribuido basado en juegos de poblaciones para gestionar dinámicamente los generadores distribuidos y las cargas inteligentes en el sistema para alcanzar el bienestar social óptimo. Se considera que los agentes no cooperan y se basan en incentivos individuales. El algoritmo propuesto preserva la estabilidad a la vez que garantiza condiciones óptimas considerando varias restricciones en el sistema sobre la operación en tiempo real. Se muestran los resultados numéricos de las estrategias de control propuestas.

Palabras clave:

Control distribuido, optimización distribuida, control transaccional, algoritmos de proyección, algoritmos asíncronos, teoría de juegos, microgrids.

Contents

List of Figures

List of Tables

1 Introduction

Power systems have been evolving towards the smart grid concept [1, 2]. A smart grid is a network that can supply the electricity demand integrating communication networks, distributed generators (renewables and non-renewables), and power storage [1]. Microgrids are defined as a cluster of small distributed generators, loads and battery energy storage systems that are connected to the point of common coupling or as a network [3]. The evolution towards the smart grid concept has been possible because of the technology progression associated with monitoring and control different power systems devices [4, 5]. As a result of large diffusion of a vast number of devices that can be monitored and control the power network, several challenges have emerged because of the necessity of coordinate them, in order to assure a robust operation [6].

However, power system evolution to smart grids has brought some problems such as economic dispatch [7], coordination between vast numbers of devices [6], among others. Economic dispatch is a problem where the main goal is minimizing the total generation cost, whereas the constraints such as power balance and generation capacity are maintained [8]. Economic dispatch initially was focused on the transmission level in power systems [9, 10]. Nevertheless, another recent focus has been how to optimize the operation in a smart grid, particularly microgrids.

To solve the economic dispatch problem, several approaches have been made including distributed energy resources. In 2008, Tsikalakis et al. [7] proposed a centralized control strategy in a microgrid where they maximize the power exchanges with the power system. The following years were evolving those approaches including distributed energy resources, [11, 12, 13, 14, 15]. Nevertheless, the approaches before mentioned are centralized methods, where a central controller receives all data and process it to make the optimal economic dispatch. Centralized control requires high computational processing, high bandwidth, and in case of a successful attack targeted to the controller, the system will be left without any control [16, 17]. The expensive communication infrastructure, and the possibility that the system may go out of control due to an attack make a centralized control a non-viable option to control a large-scale smart grid.

Distributed control is a promising strategy in order to deal with these problems. Several theories have been applied to make distributed control algorithms: Game theory [18, 19, 20], convex optimization methods [21, 22, 23] with interesting theoretical results. Several approaches have been used to economic dispatch problems, optimal power flow, and voltage control [24, 25, 26, 27, 28]. These algorithms are subject to several technical constraints. The first constraint is the coordination of devices introduced to the power system, such as smart loads, traditional and renewable generators with the objective maintaining the system stability instead of uncertainty and stochasticity of traditional loads and renewables generators on real-time operation [29]. The second constraint is to guarantee that all agents maximize their profit without destabilizing the power system.

In 2014 Dörfler et al. [30] have been proposed a distributed control strategy, in where each distributed generator can execute a control action without having all the microgrid information. Several approaches to make economic dispatch with a distributed control strategy have been proposed, among the strategies proposed has been [31, 32]. There are different approaches to make distributed economic dispatch based on game theory, [15, 33, 27], where it proves that the control algorithm based on game theory can be used to improve the results obtained with other algorithms.

Furthermore, several demand response strategies have been proposed in order to solve the economic dispatch problem with load and generation variations [34]. The inclusion of communication networks has given users the capability to interact with the distribution system through demand response mechanisms. Those mechanisms permit the users to vary its power consume considering the distribution system variables. In this context, transactive control is an effective strategy to assure coordination of a vast number of devices, including smart loads [6]. Transactive control uses a market mechanism that permits agents to interact between them through an economic signal in order to distribute the available resources [35, 36].

Transactive control, also called market-based control, is a strategy that uses a market mechanism to enabling actors to interact with each other through an economic signal to allocate the available resources [37, 38, 39]. Most of the literature of transactive control is focused on the transmission level of the electrical infrastructure [37, 39]. However, a scalable coordination approach to distribution systems operation could present several advantages over traditional control strategies.

To generate a control algorithm that can control the distribution system with all the above constraints, transactive control emerges as a strong strategy to coordinate the diverse devices in the distribution system. To use this control strategy, microgrids, generators and loads are modeled as agents. We assume that these agents act rationally. These three actors have different and definite goals. First, we have the generators, which looking to reduce their costs to generate the required energy for maximizing their profit. Second, we have consumers, who are looking for increase their utility while satisfied some minimum requirements. Microgrids are the last agent; microgrids are looking for maximizing its profit while providing the energy demanded by the loads.

Therefore, the problem to be addressed is the design of a control algorithm based on transactive control framework to control a distribution system, considering the constraints mentioned above, while it maintained an optimum operation point despite the disturbances caused by changes in the loads. To accomplish the above objective, several specific objectives were raised. Initially, we modeled the agents in the distribution system considering physical constraints and utility variables. Then, we develop four transactive control algorithms were the agents modeled before were included and finally, we test the algorithms taking into account an IEEE 30-bus power case where we prove the effectiveness of these algorithms to reach the optimum system operation while the power system constraints are maintained.

There are several contributions to these algorithms. First, we develop a fully distributed algorithm that does not need *a priori* knowledge in order to reach the optimum of the system. Second, we developed an algorithm based on game theory where our main contribution is that the algorithm can address optimization problems with linear constraints, and it also works in a distributed way. Finally, we propose an asynchronous algorithm to optimize the whole power system, this algorithm breaks a commonly used constraint that is communication synchronism.

This document is organized as follows. In chapter 2, we present the preliminaries concepts used in the development of this work, specifical concepts about graph theory, convex optimization and population games theory. In chapter 3 we present the problem statement where it is shown the model of each agent in the distribution system, i.e., traditional generators, photovoltaic generators, shiftable loads and adjustable loads. In chapter 4 we present four algorithms that aim to solve the problem statement in chapter 3. In chapter 5 we present two cases study. First one is implemented in a WSCC 9-bus system where we test the effectiveness of the algorithms presented in section 4.1 and section 4.2. In order to test the capability of algorithms presented in section 4.3 and section 4.4 to reach to the optimum operation we make a test in an IEEE 30 bus model based. Furthermore, we make an analysis of their properties and their main differences. Finally, we establish conclusions and future work.

2 Preliminaries

This chapter presents the preliminaries concepts of graph theory, convex optimization and game theory specifically population dynamics. Initially, we propose some notation used in this document.

If X is a set and x is an element of X, we write $x \in X$. A set that meets a property P is denoted as $X = \{x \mid x \text{ satisfies } P\}$. The union and intersection of the sets X_1 and X_2 is denoted as $X_1 \cup X_2$ and $X_1 \cap X_2$ respectively. We denote by $[a, b]$ the set of real numbers x that satisfies $a \leq x \leq b$. Besides a rounded, instead of square bracket means strict inequality definition as (a, b), this set satisfies $a < x < b$.

Furthermore, functions are defined using the notation $f : A \to B$, where A are the domain and B is the range of the function. When we use this notation, we assume that A and B are not empty sets.

We denote by $\mathbb{R}^n$ a vector of n-dimensions. For any $x \in \mathbb{R}^n$, we use x_i to indicate its i-th component. Vectors are designed as column vectors unless otherwise stated. $x^\top$ denotes the transpose of x, which is a n-dimensional row vector. We use the notation $\mathbb{R}_{x>0}$ and $\mathbb{R}_{x\geq 0}$ to denote that all elements of these sets are positive and nonnegative respectively. We denote a matrix as $A_{n\times m}$, where n and m are the matrix dimensions.

A function $f : [a, b] \to \mathbb{R}$ is said to satisfy the Lipschitz condition if there is a constant T such that $|f(h) - f(h')| \leq T|h - h'| \ \forall \ h, h' \in [a, b]$. Moreover, let f a function of n variables, The Hessian Matrix is a matrix of $n \times n$ where each element is the second-order partial derivatives of the function f. The Hessian matrix is denoted as $\mathbf{H}_{n\times n}(f)$.

2.1 Graph Theory

A graph is a representation of a set of elements with interconnections between them; a graph is denoted as $\mathcal{G}$. A graph is composed by a set of *vertexes* denoted as $\mathcal{V}$ and a set of interconnections between vertexes that are called *links or edges* and they are denoted as $\mathcal{E}$. w is a function that assigns a real value to each edge such that $w : \mathcal{E} \to \mathbb{R}$, if a graph has weights for all links equals to one, this graph is called *unweighted* graph. Let x and y be any agent of the graph $\mathcal{G}$. Then, a graph can be defined as [40, Section 3.4]

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, w), \tag{2-1}$$

where $\mathcal{V} = \{1, 2, 3, \ldots, V\}$, and V represents the number of vertexes in the graph. Furthermore, $\mathcal{E}$ is a set of ordered pairs of vertexes, such that $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, i.e., $(x, y) \in \mathcal{E}$ if there is a link between $x \in \mathcal{V}$ and $y \in \mathcal{V}$. A link between (x, y) implies that a vertex x can send information to y. Besides, we let $\mathcal{N}_x$ denote the neighbors of an agent $x \in \mathcal{V}$ such that $\mathcal{N}_x = \{y \mid (x, y) \in \mathcal{E}\}$ [41].

An undirected graph is a graph where if there exists a link between (x, y), then, there exist other links between (y, x) for all links. A path between x to y in a graph is a sequence of edges that connect a sequence of vertexes that starts in vertex x and finishes in vertex y. A graph $\mathcal{G}$ is connected, which means that there is at least one path between two different nodes [42, Section 2.2]. A cycle in a graph $\mathcal{G}$ is a path that starts and ends in one vertex. Three different vertexes at least compose a cycle, a graph without cycles is called a tree. Additionally, if $\mathcal{G}$ is connected, it implies the existence of at least one tree [40, subsection 3.2]. A spanning tree is a subgraph denoted as $\mathcal{G}'$ such that includes all vertexes of $\mathcal{G}$ with the minimum number of links.

An example of before statements is in **Figure 2-1** where there is an unweighted, undirected graph composed by a set of vertexes and edges defined as

$$\mathcal{V} = \{a, b, c, d, e, f, g\} \tag{2-2}$$
$$\mathcal{E} = \{(a, b), (b, c), (a, d), (b, d), (b, e), (c, e), (d, f), (e, f), (e, g), (f, g)\} \tag{2-3}$$

The black links are a path between vertex a to vertex g, this path involves a subset of (2-2) that is $\mathcal{V}_P = \{a, b, e, g\}$. Moreover, a subset of (2-3) is involved, this subset is $\mathcal{E}_P = \{(a, b), (b, e), (e, g)\}$. Finally, the blue links are a tree that involves all vertexes in the graph without forming a cycle.

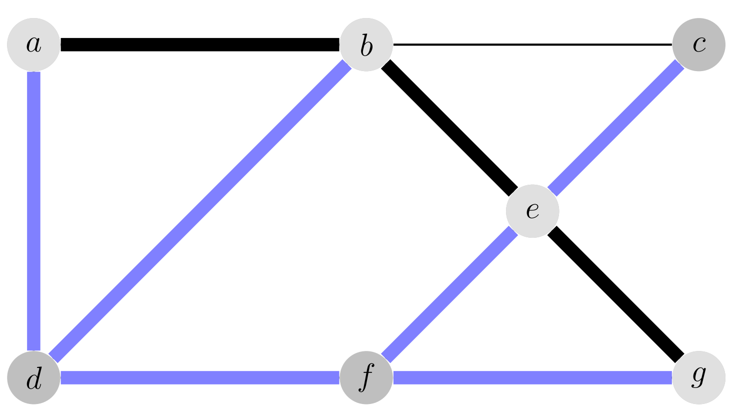

Figure 2-1: Graph representation

Adjacency Matrix

For an undirected, weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, with $\mathcal{V} = \{1, 2, \ldots, V\}$. The adjacency matrix of $\mathcal{G}$ is a non-negative matrix denoted as $A(\mathcal{G}) \in \mathbb{R}^{V \times V}$. Positions $A(x, y)$ and $A(y, x)$ are equal to the weight of the edge (x, y), and all other positions of A are equal to zero. Then, for an undirected graph, the adjacency matrix is a symmetric matrix. Let a_{xy} be the weight of an edge, an adjacency matrix is shown in (2-4).

$$A(\mathcal{G}) = \begin{bmatrix} 0 & a_{12} & a_{13} & \ldots & a_{1V} \\ a_{21} & 0 & a_{23} & \ldots & a_{2V} \\ a_{31} & a_{32} & 0 & \ldots & a_{3V} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{V1} & a_{V2} & a_{V3} & \ldots & 0 \end{bmatrix} \tag{2-4}$$

Degree Matrix

The *degree* of a vertex x is the cardinality of the neighborhood set of x, it is denoted as $r_x = |\mathcal{N}_x|$. Then, by definition a degree matrix is a diagonal matrix, denoted as $\Delta(\mathcal{G})$, a degree matrix is shown in (2-5).

$$\Delta(\mathcal{G}) = \begin{bmatrix} r_1 & 0 & 0 & \ldots & 0 \\ 0 & r_2 & 0 & \ldots & 0 \\ 0 & 0 & r_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & r_V \end{bmatrix} \tag{2-5}$$

2.2 Convex Optimization

This section presents the convex optimization and analysis concepts used during this work, and it follows the work in [43, 44, 45, 46]. This chapter presents some definitions about convex sets and convex functions initially. Then, we define the projection operation that is extensively used in this work. Finally, Lagrange multipliers and duality are explained.

2.2.1 Convex Sets

Let x and y be two points in $\mathbb{R}^n$, where n is an arbitrary space dimension. A segment line between x and y can be defined if $x \neq y$ as [44]

$$z = \alpha x + (1 - \alpha) y, \qquad \alpha \in [0, 1] \tag{2-6}$$

Let $\mathcal{C}$ a set, let x and y be any two points in $\mathcal{C}$. $\mathcal{C}$ is a convex set if any line segment between x and y always belongs to $\mathcal{C}$.

Definition 1 *Taken from [43]. A set $\mathcal{C} \subset \mathbb{R}^n$ is convex if*

$$\alpha x + (1-\alpha)y \in \mathcal{C}, \qquad \forall\, x,\ y\ \in\ \mathcal{C}, \forall\, \alpha\ \in [0,1] \tag{2-7}$$

As a consequence, it is possible to deduce that an arbitrary point x inside a convex set $\mathcal{C}$ can see all points in the convex set, furthermore, these arbitrary points can be connected with a line that always belongs to $\mathcal{C}$ as is shown in **2-2**.

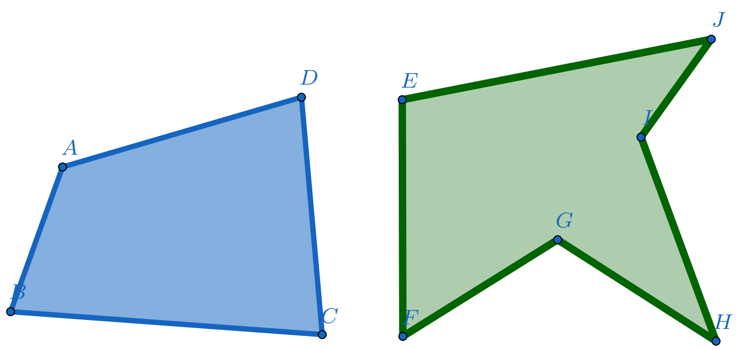

Figure 2-2: Convex and no convex sets representation. On the left, in blue there is a convex set. On the right, in green there is a no convex set, note that a segment line (H, I) does not belong anywhere to the set, then the set on the right is called a no convex set.

Some operations are possible to make with convex sets that preserve convexity of the sets [43]. Let $\mathcal{I}_C$ be a set of convex sets as $\mathcal{I}_C = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_I\}$

Proposition 1 *The intersection of convex sets, i.e., $\cap_i \mathcal{C}_i$, $\{i \in \mathcal{I}_C\}$ is convex.*

Proposition 2 *A sum between two convex sets $\mathcal{C}_1 + \mathcal{C}_2$ is convex.*

Proposition 3 *Let χ be a scalar that belongs to $\mathbb{R}$, the set $\chi\mathcal{C}_1$ is convex*

Proposition 4 *The closure and the interior of a convex set are convex.*

2.2.2 Convex Functions

A function $f : \mathcal{C} \to \mathbb{R}$ is convex, where $\mathcal{C}$ is a convex subset of $\mathbb{R}^n$. In other words, if the domain of the function f is a convex set. i,e a convex function respects

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \qquad \forall\, x, y\ \in \mathcal{C},\ \ \forall\, \alpha \in [0,1]. \tag{2-8}$$

A convex function has characteristics highly used in optimization analysis, these characteristics are:

- A twice continuously differentiable function is convex on a convex set if and only if its Hessian matrix is positive semi-definite on the interior of the convex set.
- A local minimum is also a global minimum, and then a convex function only has one minimum.

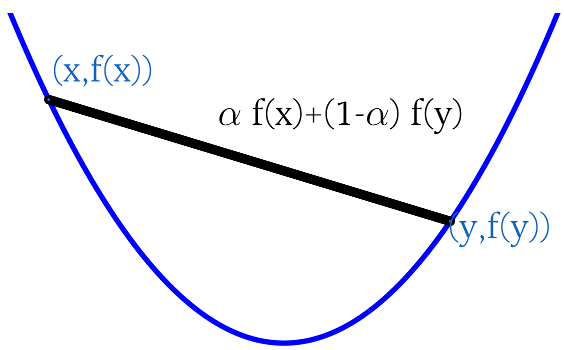

Figure 2-3: Representation of a convex function.

2.2.3 Global and local minima

In this subsection, it is given some basic concepts about global and local minima related to convex optimization. First of all, consider a function $f : \mathbb{R}^n \to (-\infty, \infty)$. We consider the problem of minimizing $f(x)$ over $\mathbf{X}$, where $\mathbf{X}$ is a subset of $\mathbb{R}^n$, as is shown

Problem 1

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathbf{X} \end{aligned} \tag{2-9}$$

As a result of (2-9), we obtain a vector $x^* \in \mathbf{X}$, this vector is called *minimizing point* or *global minimum over* $\mathbf{X}$, this vector is

$$x^* = \underset{x \in \mathbf{X}}{\mathbf{argmin}}\, f(x). \tag{2-10}$$

It is proved through the Weistrass theorem [43] that any continuous function has a minimum over a compact set, then we can assume the existence of (2-10).

2.2.4 Projection Operation

A projection is a linear transformation from a vector space to a subset of it. Among its characteristics is the idempotency, which means that a projection gives the same result if the linear transformation is applied once or more times [47]. The projection is denoted as $\mathbb{P}_X[\cdot]$, and it is defined as

$$\mathbb{P}_{\mathbf{X}}[Y] = \underset{a \in \mathbf{X}}{\mathbf{argmin}}\, \|a - Y\|, \tag{2-11}$$

where X is the set where a is feasible [43, 46]. Consider a compact set $\mathbf{X} \subset \mathbb{R}^n$ and a point $Y \notin \mathbf{X}$, the projection operation moves point Y to the nearest point that belongs to $\mathbf{X}$ as is shown in Figure **2-4**. If the point $Y \in \mathbf{X}$, the projection operation does not move the point.

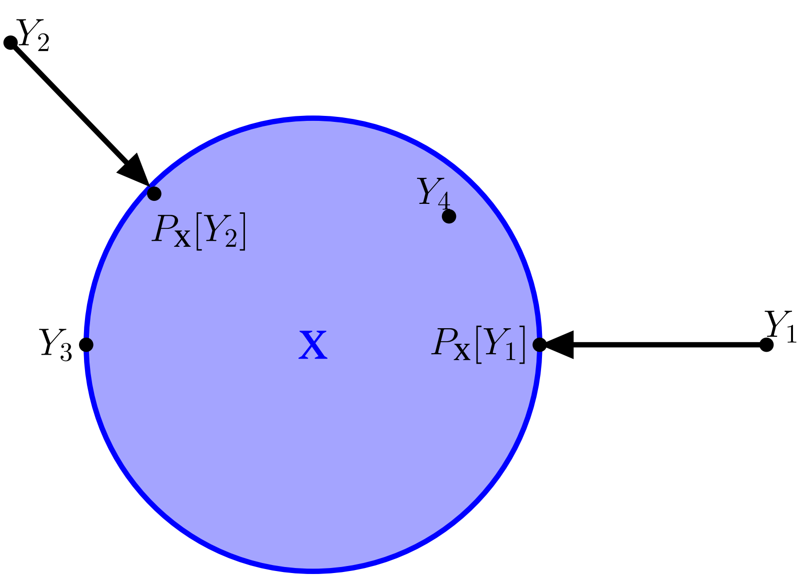

Figure 2-4: Projection operation over the points Y_1,Y_2,Y_3 and Y_4

2.2.5 Lagrange Multipliers

Consider a more specific case of (2-9), as it is shown in (2-12), in several text it is refered as primal problem [43, 44].

Problem 2 *Consider an smooth objective function $f : \mathbb{R}^n \to \mathbb{R}$, constraints functions $g : \mathbb{R}^I \to \mathbb{R}$ and $h : \mathbb{R}^J \to \mathbb{R}$*

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) && \text{(2-12a)} \\ \text{subject to} \quad & g_i(x) \leq 0 \quad i = 1, 2, \ldots, I, && \text{(2-12b)} \\ & h_j(x) = 0 \quad j = 1, 2, \ldots, J && \text{(2-12c)} \end{aligned}$$

where x is the optimization variable that belongs to vectorial space $\mathbb{R}^n$. It is assumed for this problem that it exists at least a x that satisfies $h_j(x) = 0$ and $g_i(x) \leq 0 \; \forall \; i \in I, j \in J$. In other words, we assume that the domain $D = \bigcap_{i=0}^{I} \textbf{dom}\, g_i(x) \cap \bigcap_{j=0}^{J} \textbf{dom}\, h_j(x)$ is not empty. Inside D it is assumed that exists a point optimal point denoted as x^*.

The main idea of Lagrange duality is to include (2-12b) and (2-12c) into the objective function in (2-12). Constraints (2-12b) and (2-12c) are included through a weighted sum augmenting the objective function as is shown

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \sum_{i=0}^{I} \lambda_i g_i(x) + \sum_{j=0}^{J} \mu_j h_j(x), \tag{2-13}$$

where $\lambda \in R^I_{>0}$ is the Lagrange multiplier associated to inequalities constraints and $\mu \in R^J_{>0}$ is the Lagrange multiplier associated to equalities constraints. It is possible to note in (2-13), if the constraints are not fulfilled, the value of the (2-13) is higher and then the optimal point of $\mathcal{L}$ is not reached.

2.2.6 Lagrange Duality

Consider the minimization of x in (2-13), it is well defined in [44] that the Lagrange dual function is

$$q(\lambda, \mu) = \inf_{x \in X} \mathcal{L}(x, \lambda, \mu) = \inf_{x \in X} \left(f(x) + \sum_{i=0}^{I} \lambda_i g_i(x) + \sum_{j=0}^{J} \mu_j h_j(x) \right) \tag{2-14}$$

where $h : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}$, h is the minimum value of (2-13). When Lagrange duality is applied x is no longer a variable. Since the dual function is the punctual minimum of a family of related functions (λ, μ), it is concave, even when the problem (2-12) is not convex [44].

2.2.6.1 The Lagrange dual Problem

Consider $\lambda \in \mathbb{R}_{>0}$ and $\mu \in \mathbb{R}$. Then, the dual function always produces lower bounds on the optimal value p^*, and this is denoted as

$$q(\lambda, \mu) \leq x^* \tag{2-15}$$

However, it is possible to realize that p^* depends on (λ, μ), in consequence, it is possible to search a value of (λ, μ) such as we can obtain the best lower bound. This problem can be seen as

$$\underset{\lambda,\, \mu}{\text{maximize}} \quad q(\lambda, \mu) \tag{2-16a}$$

$$\text{subject to} \quad \lambda \geq 0 \tag{2-16b}$$

We denote the optimum values obtained (λ^*, μ^*) as optimal Lagrange multipliers. This problem is convex independently of the primal problem nature. The duality gap is the difference between the solution of the primal problem p^* and the dual problem d^*, and this difference is always a non-negative value. However, we use them assumption and proposition proposed in [43, Chapter 6].

Assumption 1 *(Taking from [43, Chapter 6])*
Convexity and linear constraints: *The optimal value p^* of problem (2-12) is finite, and the following hold:*

- *The set X is the intersection of a polyhedral set and a convex set C.*
- *The cost function $f : \mathbb{R}^N \to \mathbb{R}$ over C.*
- *There exists a feasible solution of the problem that belongs to the relative interior of C.*

Then, considering Assumption 1, it is possible to affirm that there is not duality gap, and there exist at least a multiplier that satisfies problem (2-12) [43, Proposition 6.4.2].

2.3 Population Games Basics

Population games describe how a mass of players evolve in time while they choose a strategy. These agents follow a strategy selected from among a finite set of pure strategies. Due to the size of the population, the analysis focuses through a payoff function associated with the chosen strategy.

2.3.1 Populations

Let $\mathcal{P} = \{1, 2, 3, \ldots, p\}$ be a population of agents. Agents in population $\mathcal{P}$ form a mass of population $m > 0$. The set of strategies available to agents in population $\mathcal{P}$ is denoted as $S = \{1, 2, 3, \ldots, n\}$. During gameplay, each agent in population $\mathcal{P}$ selects a strategy from the set of available strategies. The set of population states for population $\mathcal{P}$ is

$$X = \left\{ x \in \mathbb{R}^n_{\geq 0} : \sum_{i \in S} x_i = m \right\}.$$

The scalar $x_i \in \mathbb{R}_{\geq 0}$ represents the mass of players in population $\mathcal{P}$ choosing strategy $i \in S$. Elements of X_v, the set of vertexes of X, are called pure population states because at these states all agents choose the same strategy.

2.3.2 Payoffs or Fitness Functions

Usually, populations and strategies that can take these populations are considered as fixed. However, in order to identify a game, we use its payoff function. A fitness function $F : X \rightarrow \mathbb{R}$ is a map that assigns each social state an utility, one for each strategy in each population.

2.3.3 Replicator Dynamics

In population games, replicator dynamics has been used in several engineering applications considering some benefits in implementation such as real-time adaptation and robustness to dynamic environmental uncertainties [33, 48, 49, 29]. Replicator dynamics has been developed over several years and as a result of this work centralized and distributed forms to implement replicator dynamics has been developed [18, 49, 29, 50]. Initially, we present the centralized replicator dynamics and then the distributed case.

2.3.3.1 Centralized Replicator Dynamics

The replicator dynamics solves the problems as economic dispatch, transactive control among others, as a resource allocation problem. Replicator dynamics considers $\mathcal{N} = \{1, 2, 3, \ldots, N\}$ finite number of agents, who adopt a i-th strategy from a finite set of pure strategies.
Let $x_i \geq 0$ be the population proportion in the i-th habitat, and the stage population vector is given by $x(t) = [x_1(t), x_2(t), x_3(t), \ldots, x_N(t)]^\top$, with $\top$ as the vector transpose, where each element represents the population at each time t. The total population is given by $P = \sum_{i \in \mathcal{N}} \rho_i(t)$, where $\rho_i(t)$ corresponds to the number of individuals with pure strategy i at a time instant. The population proportion playing the i-th strategy is $x_i = \rho_i(t)P$.
Accordingly, to achieve an appropriate performance in the steady state, the total resources demanded should be the sum of all set points of resources allocated. The payoff function (F_i) is

associated with the chosen strategy. Therefore, a continuous time implementation is also used. In its general form, the replicator dynamics can be represented as

$$\dot{x}_i = \beta(F_i - \bar{F}(\mathbf{x}))x_i \tag{2-17}$$

where x_i is the state associated with the agent i, β is a parameter that permits modify the convergence speed. The average payoff in the population $\bar{F}(\mathbf{x})$ is given by

$$\bar{F}(\mathbf{x}) = \frac{1}{N}\sum_{i=0}^{N} x_i F_i \tag{2-18}$$

To guarantee the power balance, the results in [18, 51] are used, where the choice of (2-18) guarantees the invariance of the constraint set Δ defined as

$$\Delta = \left\{ x_i \in \mathbb{R}^N_{\geq 0} : \sum_{i=0}^{N} x_i = R_t \right\} \tag{2-19}$$

This result ensures that if $x_i(0) \in \Delta$, $\forall i \in \mathcal{N}$, all $x_i(t) \in \Delta$ with $t \geq 0$. In other words, ever is dispatched the resource demanded by the system, i.e., the replicator dynamic algorithm guarantee a perfect balance between resources demanded and generated.

2.3.3.2 Distributed Replicator Dynamics

The dynamic population model describes how a pure population strategy changes through time. The replicator dynamics consider a N finite number of agents in the system, who adopt a i-th strategy from a finite set of pure strategies. Accordingly, to achieve an appropriate performance in the steady state, the demanded power load should be the sum of all power set points [33].
In its general form, the distributed replicator dynamics equation can be represented as

$$\dot{x}_i = x_i \left(F_i(x_i) \sum_{j \in \mathcal{N}_i} x_j - \sum_{j \in \mathcal{N}_i} x_j F_j(x_j) \right), \tag{2-20}$$

where F_i is the fitness function associated with each strategy, and $\mathcal{N}_i$ is the neighborhood of agent i sharing information through a communication network. In [29], we can find the simplex invariance for the distributed replicator dynamics, which means that if $x(0) \in \Delta$, the dynamic variable $x_i(t)$ evolves inside Δ.
The fundamental concept in the design of the population games is the appropriate selection of the fitness function such that the distributed replicator dynamic equation (2-20) can be used as a constrained distributed optimization algorithm [18] as is shown in Section 4.4.

2.3.4 Discrete Time Population Dynamics

In order to implement the centralized and distributed replicator dynamics we use a discrete time simulation, due to the implementation is made in digital computers and using a digital communication system. Furthermore, we use this discrete time implementation in order to make comparisons with the others proposed algorithms.

Then, in this context replicator dynamics are represented as a discrete time system. First, consider $k \in \mathbb{R}_{\geq 0}$ and $\alpha \in \mathbb{R}_{>0}$, where k denotes discrete time or iterations and α is a stepsize, in this work we assume a fix stepsize. Then, (2-17) and (2-20) in a discrete time implementation are presented in (2-21) and (2-21).

$$x_i(k+1) = x_i(k) + \alpha x_i(k) \left(f_i(k) - \frac{1}{n} \sum_{i=0}^{N} f_i(k) \right) \tag{2-21}$$

$$x_i(k+1) = x_i(k) + \alpha x_i(k) \left(f_i(k) \sum_{j \in \mathcal{N}_i} x_j(k) - \sum_{j \in \mathcal{N}_i} x_j(k) f_j(k) \right) \tag{2-22}$$

In the following section we present the problem statement where we present the agent models, estimation about line losses and the social welfare problem.

3 Problem Statement

This work is focused explicitly on distribution systems because the distributed generation is highly used near to loads [52]. We study an economic dispatch problem where loads and generators interact between them, called social welfare problem [53, 37, 39], which is described in Section 3.3. The models used in this work are described in Section 3.1, we use two generators models, two consumers models and microgrid model, which is, in fact, a combination of generators and loads. Traditionally, the losses in distribution lines are high, this was taken into the system modeling, and they are presented in the Section 3.2.

3.1 Agents in the Distribution System

We represent the distribution network as a graph $\mathcal{G}$, where each agent is represented by a node. We consider five types of agents, two generators, two consumers and a microgrid. Each one of them has different characteristics that are modeled through an objective function and several constraints depending on their characteristics. The objective function represents a cost function for generators and microgrids or utility function for loads. These agents interact between them through a communication network that is represented by edges in the graph $\mathcal{G}$.

3.1.1 Generators

Generators are considered as distributed energy resources. We assume that there are two types of generators, traditional and photovoltaic generators (renewable generators). These generators are dispatchable. A dispatchable generator can change the power that it is generating by taking into account the system requirements (i.e., the generation-demand balance), and respond dynamically to changes in the power demanded by consumers. Photovoltaic generators are dispatchable due to the implementation of maximum power point tracking (MPPT) algorithms [54, 55, 56].

3.1.1.1 Traditional Generators

A traditional generator is denoted as n. The set of generators is denoted as $\mathcal{N} = \{1, 2, 3, \ldots, N\}$, where N is the number of traditional generators in the network. Moreover, the cost of generation is assumed quadratic [37, 39], i.e.,

$$C(P_{g_n}) = \rho_{g_n} P_{g_n} + \frac{\beta_{g_n}}{2} P_{g_n}^2 \tag{3-1}$$

where $\rho_g \in \mathbb{R}^N$ and $\beta_g \in \mathbb{R}^N$ are cost coefficients, $P_g(k) \in \mathbb{R}^N$ is a vector that contains the power delivered by each generator at time instant k, $P_g(k) = [P_{g_1}(k), P_{g_2}(k), P_{g_3}(k), \ldots, P_{g_N}(k)]^\top$. Moreover, each generator has local constraints given by

$$\overline{P}_{g_n} \geq P_{g_n} \geq \underline{P}_{g_n}, \tag{3-2}$$

where $\underline{P}_{g_n}$ and $\overline{P}_{g_n}$ are the minimum and maximum power delivered by the n-th generator, respectively. We assume that $\underline{P}_{g_n}$ and $\overline{P}_{g_n}$ do not have changes through the time.
A traditional generator is denoted as n. The set of generators is denoted as $\mathcal{N} = \{1, 2, 3, \ldots, N\}$, where N is the number of traditional generators in the network. Moreover, the cost of generation is assumed quadratic [37, 39], i.e.,

$$C(P_{g_n}) = \rho_{g_n} P_{g_n} + \frac{\beta_{g_n}}{2} P_{g_n}^2 \tag{3-3}$$

where $\rho_g \in \mathbb{R}^N$ and $\beta_g \in \mathbb{R}^N$ are cost coefficients, $P_g(k) \in \mathbb{R}^N$ is a vector that contains the power delivered by each generator at time instant k, $P_g(k) = [P_{g_1}(k), P_{g_2}(k), P_{g_3}(k), \ldots, P_{g_N}(k)]^\top$. Moreover, each generator has local constraints given by

$$\overline{P}_{g_n} \geq P_{g_n} \geq \underline{P}_{g_n}, \tag{3-4}$$

where $\underline{P}_{g_n}$ and $\overline{P}_{g_n}$ are the minimum and maximum power delivered by the n-th generator, respectively. We assume that $\underline{P}_{g_n}$ and $\overline{P}_{g_n}$ do not have changes through the time.

3.1.1.2 Photovoltaic Generators

In this approach, we consider that within the microgrid agents, there may be photovoltaic generators. These generators have inherent stochasticity in the way as electricity is generated due to depending on the solar radiation at each instant. Let r be any photovoltaic generator in the system such that $r \in \mathcal{R} = \{1, 2, 3, \ldots, R\}$. The cost function of these generators is a quadratic function [34] represented by

$$C(P_{P_r}) = \rho_{P_r} P_{p_r} + \frac{\beta_{P_r}}{2} P_{p_r}^2, \tag{3-5}$$

where $\rho_P \in \mathbb{R}^R$ and $\beta_P \in \mathbb{R}^R$ are cost coefficients, $P_p(k) \in \mathbb{R}^R$ is the vector that contains the power generated by each photovoltaic generator at time instant k as follows

$$P_p(k) = [P_{p_1}(k), P_{p_2}(k), P_{p_3}(k), \ldots, P_{p_R}(k)]^\top, \tag{3-6}$$

P_{p_r} represents the power of the agent r. Maximum power available of any photovoltaic generator ($\overline{P_{p_r}}$) is proportional to the photovoltaic system area (A_r), efficiency of the photovoltaic panel (e_r) and solar radiation data (SR_r) [57], as follows

$$\overline{P_{p_r}} = A_r e_r SR_r. \tag{3-7}$$

SR_r is a property that makes power available by photovoltaic generator r to change due to this parameter varies through the time. The agent r has local constraints given by

$$\overline{P}_{p_r} \geq P_{p_r} \geq \underline{P}_{p_r}, \tag{3-8}$$

where $\overline{P}_{p_r}$ and $\underline{P}_{p_r}$ are the maximum and minimum power of the photovoltaic generator, respectively. Then, considering that SR_r is a variable parameter, it is possible to show that $\overline{P}_{p_r}$ and $\underline{P}_{p_r}$ are variables that change over time.

3.1.2 Consumers

Consumers are divided into three types of consumers. These are adjustable, shiftable and base loads. Initially, the base load is that which is necessary and cannot be negotiated with the network. However, the base load can be changed through the time depending on users preferences, and then the base load does not have any utility function. The shiftable load is a load that can vary over time depending on its objective function. However, this type of load can only support two states, *on* or *off* load. Finally, an adjustable load is a load that can vary over time depending on its objective function, and this load can support several states in order to maximize their profit.

3.1.2.1 Shiftable Load

Shiftable loads are those that can shift their consumption to another time zone, in order to prevent the shifting load from being activated at times when there is more congestion in the network, and therefore a higher energy price, examples of this kind of load are the washing machine and electric vehicles. The set of consumers with shiftable loads is denoted as $s \in \mathcal{S} = \{1, 2, 3, \ldots, S\}$. We assume that these agents can obtain several variables of the distribution system through demand response devices. As well as all agents in the distribution system, shiftable loads have an objective function that measures the utility of turn on or turn off the load, and it is represented as

$$U(P_{L_s}) = \rho_L P_{L_s} + \frac{\beta_L}{2} P_{L_s}^2, \tag{3-9}$$

where $\rho_L \in \mathbb{R}^S$ and $\beta_L \in \mathbb{R}^S$ are utility coefficients, $P_L(k) \in \mathbb{R}^S$ is the vector that contains the power consumed by each shiftable load at time instant k as follows

$$P_L(k) = [P_{L_1}(k), P_{L_2}(k), P_{L_3}(k), \ldots, P_{L_{S(k)}}]^\top, \tag{3-10}$$

P_{L_s} represents the power of the agent s. Demand response devices are capable of making decisions about turn on or off the load. If the objective function is equal or greater than a threshold, then the load is turned on. If the objective function is lower than a threshold, then the load is turned off as is shown

$$P_{L_s} = \begin{cases} \overline{P}_{L_s} & \text{if } U(P_{L_s}) \geq T_s; \\ 0, & \text{if } U(P_{L_s}) < T_s; \end{cases} \tag{3-11}$$

where T_s is the threshold of the user s, this threshold is based on users preferences, and it can switch between users.

3.1.2.2 Adjustable Load

Adjustable loads are those that can adjust its consume taking into account several information given by the network. This kind of load has several states, and it chooses one of them depending on its utility through its objective function, an example of this kind of load is the air conditioning. The set of consumers is denoted as $a \in \mathcal{A} = \{1, 2, 3, \ldots, A\}$. Consumers are assumed to be agents who can obtain some system variables through demand response devices. These devices are capable of making decisions about the amount of power required by each consumer in order to maximize its utility [58]. The consumers' utility function is defined as [37]

$$U(P_{d_a}) = \rho_{d_a} P_{d_a} + \frac{\beta_{d_a}}{2} P_{d_a}^2, \tag{3-12}$$

where $\rho_d \in \mathbb{R}^A$ and $\beta_d \in \mathbb{R}^A$ are utility coefficients, $P_d(k) \in \mathbb{R}^A$ is the vector that contains the load demanded by each smart consumer at time instant k, $P_d(k) = [P_{d_1}(k), P_{d_2}(k), \ldots, P_{d_A}(k)]^\top$. The agent a-th has local constraints given by

$$\overline{P}_{d_a} \geq P_{d_a} \geq \underline{P}_{d_a}, \tag{3-13}$$

where $\underline{P}_{d_a}$ and $\overline{P}_{d_a}$ are the minimum and maximum power demanded by the load a-th respectively.

3.1.3 Microgrids

Currently distribution systems require to any agent that wants to be dispatched in an economic dispatch to have a minimum power available. However, this amount of power is usually only meet by industrial consumers and generators.Microgrids have appeared in order to include small agents as an active part of the distribution systems.

Microgrids have certain characteristics that make them the cornerstone of the transition to smart grids. First of all, they include small adjustable loads, shiftable loads, traditional generators and renewables generators [58, 59]. Moreover, the agents inside a microgrid can communicate between them as is shown in Figure **3-1**.

A microgrid can be grid-tied or off-grid, as consequence a microgrid can operate isolated from the main grid. The main restriction in a microgrid is that this is delimited to a specific geographical area [60]. Let q be any microgrid in the power system such that $q \in \mathcal{Q} = \{1, 2, 3, \ldots, Q\}$. Then the microgrid utility function can be denoted as

$$U_q\left(P_M\right) = \sum_{a_q=1}^{A^q} U\left(P_{d_{a_q}}\right) + \sum_{s_q=1}^{S^q} U\left(P_{L_{s_q}}\right) - \sum_{n_q=1}^{N^q} C\left(P_{g_{n_q}}\right) - \sum_{r_q=1}^{R^q} C\left(P_{p_{r_q}}\right), \tag{3-14}$$

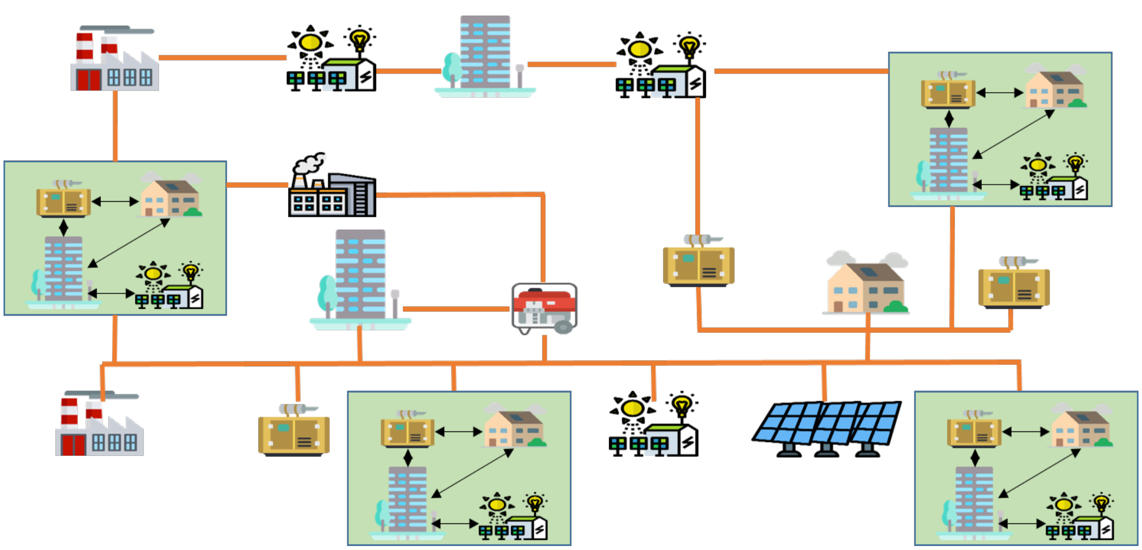

Figure 3-1: General diagram of a power system with high penetration of microgrids. The elements enclosed in green are microgrids. The orange connections represent the power lines among each agent.

where P_M is a stacked vector such that $P_M = \left[P_d^q, P_L^q, P_g^q, P_p^q\right]^\top$. Any agent in the microgrid q is denoted as $x_q \in \mathcal{V}^q$. Besides, $P_{d_{a_q}}$ is the power demanded by a consumer a_q belonging to the microgrid q such that $a_q \in \mathcal{A}^q = \{1, 2, 3, \ldots, A^q\}$. Moreover, $P_{L_{s_q}}$ is the power demanded by a consumer s_q belonging to the microgrid q such that $s_q \in \mathcal{S}^q = \{1, 2, 3, \dots, S^q\}$. Furthermore, $P_{g_{n_q}}$ is the power generated by a generator n_q belonging to the microgrid q such that $n_q \in \mathcal{N}^q = \{1, 2, 3, \dots, N^q\}$. Finally, $P_{p_{r_q}}$ is the power generated by a photovoltaic generator r_q belonging to the microgrid q such that $r_q \in \mathcal{R}^q = \{1, 2, 3, \dots, R^q\}$.

Remark 1 *SR_r in each microgrid q is considered equal for all agents $r_q \in \mathcal{R}^q$ as in [61].*

Microgrids are subject to different constraints in order to meet their main objective, i.e., optimize the state of agents inside itself. Furthermore, a microgrid can be trading with the distribution system in order to sell (buy) the available (missing) power in order to fulfill its power requirements. These constraints are shown in (3-15)

$$\sum_{a_q=1}^{A^q} P_{d_{a_q}} + \sum_{s_q=1}^{S^q} P_{L_{s_q}} - \sum_{n_q=1}^{N^q} P_{g_{n_q}} - \sum_{r_q=1}^{R^q} P_{p_{r_q}} = P_{E_q} \tag{3-15a}$$

$$\overline{P}_{E_q} \geq P_{E_q} \geq \underline{P}_{E_q} \tag{3-15b}$$

where, if $P_{E_q} \in \mathbb{R}_{\geq 0}$, P_{E_q} is the power requested by the power system after to sell the power available by the microgrid q. Otherwise, P_{E_q} is the power requested to the power system after to buy the power missing by the microgrid q in order to fulfill constraint (3-15a). Furthermore, $\overline{P}_{E_q}$ and $\underline{P}_{E_q}$ are associated to the maximum and minimum power available of each microgrid q. Then, the optimization problem for each microgrid is to optimize (3-14) while constraints (3-4), (3-8), (3-13), (3-15a) and (3-15b) are fulfilled as follows

$$\underset{P_M}{\text{maximize}} \quad U\left(P_M\right) \tag{3-16a}$$

subject to

$$\sum_{a_q=1}^{A^q} P_{d_{a_q}} + \sum_{s_q=1}^{S^q} P_{L_{s_q}} - \sum_{n_q=1}^{N^q} P_{g_{n_q}} - \sum_{r_q=1}^{R^q} P_{p_{r_q}} = P_{E_q}, \tag{3-16b}$$

$$\overline{P}_{g_{n_q}} \geq P_{g_{n_q}} \geq \underline{P}_{g_{n_q}} \quad \forall n_q \in \mathcal{N}^q, \tag{3-16c}$$

$$\overline{P}_{p_{r_q}} \geq P_{p_{r_q}} \geq \underline{P}_{p_{r_q}} \quad \forall r_q \in \mathcal{R}^q, \tag{3-16d}$$

$$\overline{P}_{d_{a_q}} \geq P_{d_{a_q}} \geq \underline{P}_{d_{a_q}} \quad \forall a_q \in \mathcal{A}^q, \tag{3-16e}$$

$$\overline{P}_{L_{s_q}} \geq P_{L_{s_q}} \geq \underline{P}_{L_{s_q}} \quad \forall s_q \in \mathcal{S}^q, \tag{3-16f}$$

$$\overline{P}_{E_q} \geq P_{E_q} \geq \underline{P}_{E_q}, \tag{3-16g}$$

where (3-14) is a convex function. We assume that the local objective function and the local constraint set are only known to the agent. Then, problem (3-16) is strongly convex. Local constraints are closed convex sets. Finally, it is assumed that exists at least a feasible point and a solution to this problem.

3.2 Power Losses

Network losses in a distribution system are factors that can be determinant to reach the optimum social welfare [62]. Network losses could be reduced making an optimal power flow, e.g., sometimes could be better generate with a generator more expensive if the load is near to the generator. Furthermore, losses varying through time depending on the load used by consumers. It is not possible to solve the system in all possible cases of demand and power generation needed to solve the problem analytically [63]. We assume that the network losses are constant in the range of operation and estimated by the Kron's formula as follows

$$P_l = \sum_{k=1}^{Z} \sum_{l=1}^{Z} p_k B_{kl} p_l + \sum_{k=1}^{Z} B_{k0} p_k d + B_{00}, \tag{3-17}$$

where $B = [B_{kl}]$, B_0, and B_{00} are the loss factors, N_l is the number of generator considered in the estimation algorithm. In order to calculate an estimation of the network losses, the algorithm in [49] is used. This algorithm can calculate loss factors which give us a closeness estimation of the losses. However, it is important to highlight that the lines will be considered but only considering active power, i.e., only the resistive component of the line is considered. Besides, non-technical losses in the distribution systems are not considered. As a result of this estimation algorithm, it is obtained a constant power P_l to be included in the algorithm as a part of the load to be dispatched.

3.3 Social Welfare Problem

The problem of social welfare is related to the optimization of the state of each agent of the electricity system, in other words, to minimize the cost of generators, maximize the utility of consumers and microgrids and it is obtained from (3-3), (3-5), (3-9), (3-12) and (3-14) as follows

$$S_W(P) = \sum_{a=1}^{A} U(P_{d_a}) + \sum_{s=1}^{S} U(P_{L_s}) + \sum_{q=1}^{Q} U(P_{M_q}) - \sum_{n=1}^{N} C(P_{g_n}) - \sum_{r=1}^{R} C(P_{p_r}). \tag{3-18}$$

The main goal is to maximize (3-18) as it is shown in (3-19), taking into account constraints (3-4), (3-8), (3-13) , (3-15). Let P be a stacked vector that contains all power of any agent in the system such that $P = [P_a, P_L, P_g, P_p, P_M]^\top$. Then, the optimization problem of the power system is

$$\underset{P}{\text{maximize}} \quad S_W(P) \tag{3-19a}$$

subject to

$$\sum_{a=1}^{A} P_{d_a} + \sum_{s=1}^{S} P_{L_s} + \sum_{q=1}^{Q} P_{M_q} - \sum_{n=1}^{N} P_{g_n} - \sum_{r=1}^{R} P_{p_r} = P_l, \tag{3-19b}$$

$$\overline{P}_{g_n} \geq P_{g_n} \geq \underline{P}_{g_n} \qquad \forall n \in \mathcal{N}, \tag{3-19c}$$

$$\overline{P}_{d_a} \geq P_{d_a} \geq \underline{P}_{d_a} \qquad \forall a \in \mathcal{A}, \tag{3-19d}$$

$$\overline{P}_{p_r} \geq P_{p_r} \geq \underline{P}_{p_r} \qquad \forall r \in \mathcal{R}, \tag{3-19e}$$

$$\overline{P}_{L_s} \geq P_{L_s} \geq \underline{P}_{L_s} \qquad \forall s \in \mathcal{S},, \tag{3-19f}$$

$$\overline{P}_{M_q} \geq P_{M_q} \geq \underline{P}_{M_q} \qquad \forall q \in \mathcal{Q}, \tag{3-19g}$$

where (3-18) is a convex function. We assume that the local objective function and the local constraint set are known to the agent only. Then, problem (3-19) is strongly convex. Local constraints are closed convex sets. Finally, it is assumed that exists at least a feasible point and a solution to this problem. In the following section, we will present the algorithms for solving the optimization problem proposed in this section.

4 Transactive Control Approaches

In this section, we present four algorithms developed to solve the problem proposed in the previous section. These algorithms are based on convex optimization theory and game theoretical methods. Initially, we present two algorithms based on convex optimization theory. The first and second algorithms are formulated in order to solve problem (3-16). The others two algorithms are formulated to solve problem (3-19).

In our first algorithm, we develop an algorithms combination in order to obtain a fully distributed transactive algorithm that without *a priori* knowledge can dispatch generators and loads in a microgrid optimally. As a result, this algorithm was presented in [64].

Second, we design a centralized algorithm that can optimize the resource allocation of power in a microgrid, and this algorithm was based on the following works [33, 48, 29]. Our main contribution to this algorithm has been made that this algorithm can address optimization problems with linear constraints. As a result, this algorithm was presented in [53].

Our third algorithm has the same base of our third algorithm, projected gradient method, and duality gradient method are used to build an asynchronous algorithm that can optimize the social welfare problem (c.f. Equation (3-18)) asynchronously, which gives to the algorithm an advantage over the others algorithms that requires a synchronous communication. In the microgrids inside the proposed power system, we use the optimization algorithm presented in Section 4.2. This algorithm was submitted to publication [65].

Finally, the last algorithm was developed taking into account our second algorithm. We made a distributed replicator dynamics where we can include several restrictions to the power system. This algorithm allows taking advantage of the algorithms made in distributed form, as each node performs small calculations instead of a single entity performs all the processing. This algorithm was submitted to publication [66].

Table 4-1: Algorithms characteristics

	Properties					Power System		Theoretical Method	
	Asynchronous	Duality	Distributed	Centralized	Hierarchical	MG	Distribution Systems with Microgrids	Game Theoretical Methods	Convex Optimization Methods
Algorithm 1			x			x			x
Algorithm 2		x		x		x		x	
Algorithm 3	x	x	x	x	x		x		x
Algorithm 4		x	x		x		x	x	

We define the following assumptions in order to solve Problem (3-19), these are about minimal conditions of utility functions, constraints sets, and initial points. X_x denote a constraint set of any agent x.

Assumption 2 *The graph $\mathcal{G}$ is connected, static, undirected, and unweighted.*

Assumption 3 *Links are assumed lossless, without delays and synchronous.*

Assumption 4 *For all agent $x \in V$, the set X_x is not-empty compact and convex.*

Assumption 5 *Consumers utility functions, generators cost functions and constraints are twice continuously differentiable.*

Assumption 6 *The feasible sets of P_{g_n} $\forall\ n \in \mathcal{N}$, P_{P_r} $\forall\ r \in \mathcal{R}$, P_{d_a} $\forall\ a \in \mathcal{A}$, P_{L_s} $\forall\ s \in \mathcal{S}$, and P_{M_q} $\forall\ q \in \mathcal{Q}$ are not-empty.*

Assumption 7 *All agents start their power inside their feasible set, i.e. $P_{g_n}(0)$, $P_{p_r}(0)$, $P_{L_s}(0)$ and $P_{d_a}(0)$ satisfy* (3-19c),(3-19d), (3-19e) *and* (3-19f).

Assumption 8 *The agents know their set of neighbors and hence their own cardinality, r_a, where $r_a = |\mathcal{N}_a|$.*

Assumption 9 *The set of feasible points for problem* (3-19) *is non-empty.*

4.1 Distributed Transactive Control in Power Systems Based on the Projected Consensus Algorithm

We propose a distributed transactive control algorithm based on the projected distributed gradient algorithm [45, 21]. In order to find a solution we consider the Assumptions 2, 3, 4, 5 and 8. These assumptions are related to graph properties, communications and agents initial knowledge. This section is based on [64].

To solve (3-16), it is necessary for this algorithm that all agents have information about the global state of the network. Agents need the following data: total power demanded (P_D), total power delivered (P_G) and finally the number of generators and consumers in the network ($N + R$) and ($A + S$) respectively. Where P_D and P_G are calculated such as

$$P_D = \sum_{a=1}^{A} P_{d_a}(k) + \sum_{s=1}^{S} P_{L_s}(k) \qquad\qquad P_G = \sum_{n=1}^{N} P_{g_n}(k) + \sum_{r=1}^{R} P_{p_r}(k)$$

Results in the recent literature assume that global parameters can be obtained *a priori* by agents. This implies that global parameters are acquired by a central agent that send the global parameters to the agents in the power system as in [67, 68, 69, 70, 48]. To relax the requirement of the central agent, we formulate a fully distributed algorithm that can find the global parameters in a distributed way efficiently. We use the **Minimum Diameter Spanning Tree** (MDST) algorithm and a **Finite Time Average** algorithm to reach consensus in a finite number of steps, these algorithms are explained in subsection 4.1.1 and 4.1.2 respectively.

The MDST algorithm is used to find a spanning tree of the graph $\mathcal{G}$ that allows to finite time averaging algorithm share the global parameters with all the agents in the power system. Later, the same spanning tree is used to execute the distributed projection gradient algorithm to compute the iterations of the optimization variables.

4.1.1 Distributed Algorithm for the Minimum Diameter Spanning Tree

We use the finite time consensus algorithm to achieve a common knowledge of the global parameters of the system at each node [71]. However, this algorithm only works on spanning tree graphs. Finding a spanning tree in a graph is a problem heavily studied in recent years [72, 73, 74, 75, 76]. We use the approach in [72] to guarantee the best convergence time for the finite-time distributed algorithm, because this algorithm converges in maximum d steps, being d the graph diameter.

Step 1 Computation of All-Pairs Shortest Paths (APSP) Algorithm.
In this step, each node calculates the minimum distance to every node in the graph.

Step 2 Computation of Absolute Center of $\mathcal{G}$.
The absolute center of $\mathcal{G}$ is a node where the eccentricity of a graph is the minimum. The information used to compute the absolute center is a list generated in ASPS algorithm where all the minimum distances between each pair of nodes are made.

Step 3 Construction of Minimum Diameter Spanning Tree.
The center ID is sent through the minimum distance path to each node in the graph. When sending the ID information, also sends other data, a counter that in each step of communication is updated until reach the diameter of $\mathcal{G}$, this procedure can be made grace a $\mathcal{G}$ is an unweighted graph, and it is a synchronous system. Finally, when all nodes in the graph know which one is the absolute center of the graph the spanning tree is constructed thought of community knowledge.

Lemma 1 *[72, Theorem 5]: Consider a graph $\mathcal{G}$ and let Assumptions 2 and 8 hold on $\mathcal{G}$. Then, the distributed algorithm for the MDST proposed in [72, Theorem 5] finds a MDST of $\mathcal{G}$, in $O(n)$ iterations.*

Proof. The distributed algorithm for the MDST has to calculate the All-Pairs Shortest Path (APSP) of the network. The time of execution of APSP is $O(n)$ [72, Lemma 4]. Then, each node knows which node is the shortest path to another node. After that, the absolute center of the graph is calculated, given a node with the lowest eccentricity x_{min}, the information about which node is the center of the graph is sent to other nodes in at most $O(n)$. Now consider the collection of all paths produced by APSP that begins in any node in the network and end in x_{min}, the set of path forms a tree rooted in x_{min}, which is the MDST of $\mathcal{G}$. Therefore each node knows a route to x_{min} and the MDST is built through of knowledge of the shortest path to x_{min} for all nodes [72, Subsection 2.1.5]. ■

Once the MDST is created, the global parameters are calculated through the distributed algorithm for the MDST as described in the following subsection.

4.1.2 Finite-time Distributed Averaging

To calculate the global parameters P_D, P_G, G and D the algorithm proposed in [71] is used. This algorithm has several characteristics that allow being used to this application such as its convergence in d steps at maximum. However, this algorithm only works in spanning tree graphs. In the beginning, each agent x in the system has initial values for the global parameters as follows

$$P_{G_x}(0) = \begin{cases} P_{g_x}(0), & \text{if } x \in \mathcal{N}, \\ P_{p_x}(0), & \text{if } x \in \mathcal{R}, \\ 0, & \text{otherwise}, \end{cases} \tag{4-1a}$$

$$P_{D_x}(0) = \begin{cases} P_{d_x}(0), & \text{if } x \in \mathcal{A}, \\ P_{L_x}(0), & \text{if } x \in \mathcal{S}, \\ 0, & \text{otherwise,} \end{cases} \tag{4-1b}$$

$$G_x(0) = \begin{cases} 1, & \text{if } x \in \mathcal{N}, \\ 1, & \text{if } x \in \mathcal{R}, \\ 0, & \text{otherwise,} \end{cases} \tag{4-1c}$$

$$D_x(0) = \begin{cases} 1, & \text{if } x \in \mathcal{A}, \\ 1, & \text{if } x \in \mathcal{S}, \\ 0, & \text{otherwise,} \end{cases} \tag{4-1d}$$

where G_x and D_x are the number of generators and consumers respectively. Let h_x be any global parameter which is displayed in (4-1), for each of the above values, each agent performs the following update action in parallel

$$h_x(q+1) = \begin{cases} h_x(0) + \sum_{y \in \mathcal{N}_x} h_y(0), & \text{if } q = 0; \\ \sum_{y \in \mathcal{N}_x} h_y(q) + (1 - r_x) h_x(q-1), & \text{if } q \geq 1. \end{cases} \tag{4-2}$$

Variable q is used to represent steps. Algorithm 1 explains how (4-2) is used to calculate the global parameters.

Lemma 2 *[71, Theorem 1]: Suppose $\mathcal{G}$ is a tree graph with diameter equal to d. Algorithm 1 makes it possible for each agent x to get the global parameters at a maximum of d steps.*

The maximum path between two agents in the MDST obtained will be the maximum number of iterations in which the algorithm converges to global values (P_D, P_G, G and D).

4.1.3 Distributed Projected Consensus Gradient

To solve the problem presented in (3-16) we use the projected consensus algorithm proposed in [21]. Our algorithm does not require private information from the neighboring agents such as its incremental cost or gradient of its utility function. The algorithm only needs the power set points of the agents in the power system in iteration k to estimate the power set points in iteration $k + 1$.

For the initialization step we let Assumptions 2, 6, 7 and 8 hold. The x-th agent updates its global parameters by using the information produced by Algorithm 1, then taking a gradient

Algorithm 1 Finite-time Distributed Averaging

Executed by: Agents $x \in \mathcal{V} = \{1, 2, 3, \ldots, V\}$
Require: Spanning Tree Neighbors
Initialize: $P_{D_x}(0)$, $P_{G_x}(0)$, $G_x(0)$, $D_x(0)$ and Set $q = 0$
$P_{D_x}(1) = P_{D_x}(0) + \sum_{y \in \mathcal{N}_x} P_{D_y}(0)$
$P_{G_x}(1) = P_{G_x}(0) + \sum_{y \in \mathcal{N}_x} P_{G_y}(0)$
$G_x(1) = G_x(0) + \sum_{y \in \mathcal{N}_x} G_y(0)$
$D_x(1) = D_x(0) + \sum_{y \in \mathcal{M}_x} D_y(0)$
while $q \leq d$ **do**
    Send $P_{D_x}(q)$, $P_{G_x}(q)$,$G_x(q)$ and $D_x(q)$ to Spanning Tree neighbors
    $P_{D_x}(q+1) = \sum_{y \in \mathcal{N}_x} P_{D_y}(q) + (1 - r_x) P_{D_x}(q-1)$
    $P_{G_x}(q+1) = \sum_{y \in \mathcal{N}_x} P_{G_y}(q) + (1 - r_x) P_{G_x}(q-1)$
    $G_x(q+1) = \sum_{y \in \mathcal{G}_x} N_y(q) + (1 - r_x) G_x(q-1)$
    $D_x(q+1) = \sum_{y \in \mathcal{N}_x} D_y(q) + (1 - r_x) D_x(q-1)$
    Set $q = q + 1$;
end while
return $P_{D_x}(q)$, $P_{G_x}(q)$, $G_x(q)$ and $D_x(q)$
▷ $P_{D_x}(q)$=P_D, $P_{G_x}(q)$=P_G, $G_x(q)$=$N + R$, $D_x(q)$=$A + S$

step to minimize the cost or maximize the utility function, and then projecting the result onto its constraint set X_x, where X_x is the set of feasible solutions for each agent (cf. Equations (3-4), (3-8), (3-11) and (3-13)).

Initially, we seek for reach an average consensus, for this we use the Algorithm 1. We define the stacked vector of power (generated and demanded) as $P^\top = [P_g, P_p, P_L, P_d]^\top$, P_x is the power of agent $x \in \mathcal{V}$, P_y is the power of the neighbors of agent x, with $y \in \mathcal{N}_x$, we use the variable v_x to store the sum of the powers in the iterative system as follows

$$\begin{cases} v_x(1) = & P_x(0) + \sum_{y \in \mathcal{N}_x} P_y(0) \\ v_x(2) = & \sum_{y \in \mathcal{N}_x} P_y(1) + (1 - r_x) P_x(0) \\ & \vdots \\ v_x(d) = & \sum_{y \in \mathcal{N}_x} P_y(d-1) + (1 - r_x) P_{D_x}(d-2). \end{cases} \tag{4-3}$$

When (4-3) has been executed, every agent $x \in \mathcal{V}$ has a value $v_x(d) = \sum_{x \in \mathcal{V}} P_x$. Once $v_x(d)$ is obtained, it is possible to take the gradient step. For this step, we use $z(k)$ which contains the power average consensus minus the gradient of cost or utility function such as

$$z(k) = \frac{v_x(d)}{G_x + D_x} - \alpha_k d_x(k), \tag{4-4}$$

where $\alpha_k > 0$ is the stepsize, $d_x(k)$ is the gradient of the utility function depending on each agent (cf. Equations (3-3), (3-5), (3-9) and (3-12)). Finally, $z(k)$ is projected onto the feasible sets X_x. Each agent makes projections taking into account the constraints to which it is subjected. Consumers and generators are subject to constraints associated to its maximum and minimum load and generation, respectively. The projection onto the feasible set is defined as follows:

$$\mathbb{P}_{X_x}[z(k)] = z_1(k) = \begin{cases} \overline{P_x}, & \text{if} \quad z(k) > \overline{P_x}, \\ \underline{P_x}, & \text{if} \quad z(k) < \underline{P_x}, \\ z(k) & \text{Otherwise.} \end{cases} \tag{4-5}$$

Furthermore, generators have to maintain the equality constraint shown in (3-15a), the projection onto the feasible set X_g, where X_g is the constraint set where constraint (3-15a) is held, is shown in (4-6).

$$\mathbb{P}_{X_g}[z_1(k)] = \begin{cases} z_1(k), & \text{if (3-15a) is hold,} \\ z_1(k) - \dfrac{P_{G_x}(k) - P_{D_x}(k)}{G_x} & \text{Otherwise.} \end{cases} \tag{4-6}$$

Finally, the power update law $P_x(k+1)$ is given by for consumers and generator agents as follows

$$P_x(k+1) = \begin{cases} \mathbb{P}_{X_x}[z(k)] & \text{if } x \in \mathcal{A} \\ \mathbb{P}_{X_x}[z(k)] & \text{if } x \in \mathcal{S} \\ \mathbb{P}_{X_x}[z_1(k)] & \text{if } x \in \mathcal{N} \\ \mathbb{P}_{X_x}[z_1(k)] & \text{if } x \in \mathcal{R} \end{cases} \tag{4-7}$$

4.1.4 Distributed Transactive Algorithm

We now state the distributed transactive algorithm, that is, the main contribution of this paper. We use k to denote the iterations in the algorithm.

Algorithm 2 Distributed Transactive Control Algorithm

Executed by: Agents $x \in \mathcal{V} = \{1, 2, 3, \ldots, V\}$
Require: d_x, α_x and r_x
Initialize: Assumption 7 is hold $\forall\, a$. Set $k = 0$
Execute Algorithm MDST in Subsection 4.1.1.
while $k \geq 0$ **do**
  Send $P_x(k)$ to all $y \in \mathcal{N}_x$
  Execute Algorithm 1
  Obtain $P_G(k)$, $P_D(k)$, $G(k)$ and $D(k)$
  Execute For all $x \in \mathcal{V}$
    Equation (4-3) and then (4-4)
  Execute For all $x \in \mathcal{V}$
    Equation (4-5)
  Execute For all $x \in \mathcal{N} \;\vee\; x \in \mathcal{R}$
    Equation (4-6)
  Obtain $P_x(k+1)$
  Set $k = k + 1$;
end while

4.1.5 Convergence

In this subsection, we analyze and prove the convergence of the proposed distributed transactive algorithm.

Theorem 1 *Assume that the stepsize α satisfies that $\sum_k \alpha_k = \infty$ and $\sum_k \alpha_k^2 \geq \infty$. Furthermore, let $P_x(k)$, with $x \in \mathcal{V}$, be the set points generated by Algorithm 2 and $X = \cap_{x=1}^{V} X_x$ be the intersection set between all feasible sets of the agents. Then, $P_x(k)$ with $x \in \mathcal{V}$ converges to the optimal solution P_x^* with $P_x^* \in X$, that is*

$$\lim_{k \to \infty} P_x(k) = P_x^*.$$

Proof. Without loss of generality, all agents can be listed such as in (3-16). The optimization problem defined in (3-19) can be generalized as

$$\underset{P_x}{\text{minimize}} \quad \sum_{x=1}^{V} U_x(P_x) \tag{4-8a}$$

$$\text{subject to} \quad \sum_{x=1}^{V} P_x = 0, \tag{4-8b}$$

$$\overline{P_x} \geq P_x \geq \underline{P_x} \quad \forall x \in \mathcal{V} \tag{4-8c}$$

with $P_x(k)$ and $U_x(\cdot)$ defined as follows

$$P_x(k) = \begin{cases} P_{d_a}, & a = 1, \dots, A \\ P_{L_{s+A}}, & s = A+1, \dots, S+A \\ -P_{g_{n+S+A}}, & n = S+A+1, \dots, N+S+A \\ -P_{p_{r+N+S+A}}, & p = N+S+A+1, \dots, V \end{cases} \tag{4-9a}$$

$$U_x(\cdot) = \begin{cases} U_j(\cdot), & j = 1, \dots, A+S \\ -C_i(\cdot), & i = A+S+1, \dots, V \end{cases} \tag{4-9b}$$

The constraints (4-8b) and (4-8c) can be written such as $X = \cap_{x=1}^{V} X_x \cap X_g$, where X_g is the set of feasible solutions for (4-8b). It is possible to write (4-8) such as

$$\underset{P_x}{\text{minimize}} \quad \sum_{x=1}^{V} U_x(P_x) \tag{4-10a}$$

$$\text{subject to} \quad X = \cap_{x=1}^{V} X_x \cap X_g \tag{4-10b}$$

It is assumed that X_x and X_g are compact sets. Given that all utility functions $U_x(P_x)$ are continuous, and based on Weierstrass' Theorem (4-10) has an optimal solution $P_x^* \in X$. Considering that $\frac{v_x(d)}{N_x+M_x}$ in (4-4) is equal to

$$\frac{v_x(d)}{N_x + M_x} = \lim_{k\to\infty} \frac{1}{V} \sum_{x \in \mathcal{V}} P_x$$

and according to [21, Proposition 5] agents executing Algorithm 2 converges to $P_x^* \in X$. ■

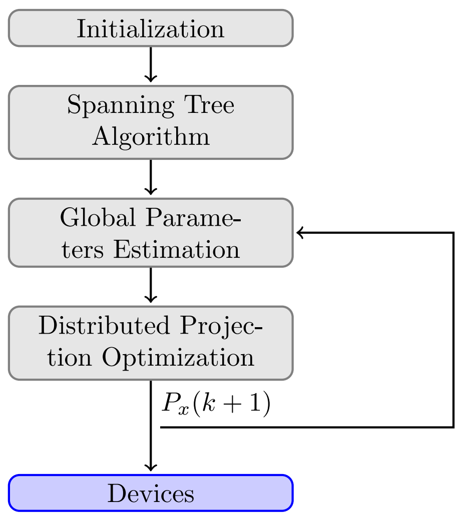

Figure 4-1: Flowchart of Distributed Transactive Control in Power Systems based on the Projected Consensus Algorithm

4.2 Centralized Transactive Control Based on Replicator Dynamics

The main contribution of this algorithm is to improve the centralized replicator dynamics algorithm including the constraints through Lagrange multipliers. This section is based on [53]. Transactive control finds the optimum equilibrium point based on the power which must be generated by each generator. Moreover, the power which should be consumed by the end user in order to minimize the generators cost and maximize the users' utility. We denote any agent in the system as x. The main goal is to maximize 3-16, as it is shown in the following problem:

$$\underset{P}{\text{maximize}} \quad U(P) \tag{4-11a}$$

$$\text{subject to} \quad \mathcal{R}P = \mathbf{r} \tag{4-11b}$$

Constraints could be represented by a matrix as $\mathcal{R}P = \mathbf{r}$, where $P = [P_g, P_p, P_d, P_L]^\top$ and $\mathcal{R} \in \mathbb{R}^M$, where M is the number of system constraints, this is made possible by transforming inequalities through slack variables. In order to resolve the optimization problem (4-11), we have used population dynamics models [48, 29]. Population dynamics represent how a pure population strategy progress through time. The replicator dynamics consider a V finite number of generators and loads, who adopt a ith strategy from a finite set of pure strategies. Accordingly, to achieve an appropriate performance in the steady state, the load demanded power should be the sum of all power set points [33].

The payoff function is associated with the chosen strategy; therefore, we also use a continuous time implementation. In its general form, the replicator dynamics can be represented as

$$\dot{p_x} = \beta p_x(f_x - \bar{F}) \tag{4-12}$$

where $p = [p_1, p_2, p_3, \ldots, p_V]^\top = [P_g, P_p, P_d, P_L]^\top$, β is a speed parameter for the convergence of the replicator dynamics, f_x is the fitness function associated with each strategy, and $\bar{F}$ is the average fitness function [29]. In this work, we use replicator dynamics (2-17) to solve the economic dispatch problem dynamically. In order to integrate the restrictions within the replicator dynamics use the Lagrange function

$$\mathcal{L}(P, \mu) = S_W + \mu^\top(\mathcal{R}P - \mathbf{r}) \tag{4-13}$$

Where μ is the Lagrange multipliers, the fitness functions of the game are defined as $f(p) = [f_1, f_2, f_3, \ldots, f_V]^\top = \nabla_P \mathcal{L}$ and $f(\mu) = \nabla_\mu \mathcal{L}$. Thus, the fitness functions are

$$\nabla_P \mathcal{L}(P, \mu) = \nabla S_W(P) + \mathcal{R}^\top \mu \tag{4-14}$$

$$\nabla_\mu \mathcal{L}(P, \mu) = -\mathcal{R}P + \mathbf{r} \tag{4-15}$$

The average of these fitness functions is

$$\bar{F}_{P_G} = \sum_{j=0}^{N+R} p_j * F_j \tag{4-16}$$

$$\bar{F}_{P_D} = \sum_{j=0}^{A+S} p_j * F_j \tag{4-17}$$

$$\bar{F}_\mu = \sum_{j=0}^{M} p_j * F_j \tag{4-18}$$

Here p_j is the power in each agent and F_j is the fitness function associated with this variable. In fact, this process is made two times, i.e, for each fitness equation (4-14) and (4-15) we have a fitness average function. Furthermore, we obtain two average fitness functions (4-16) to generator agents and (4-17) to consumer agents. This optimization can be made it due to the fitness functions used to represent the model are convex functions.

With the above optimization problem, the microgrid can be controlled optimally taking into account not only maximizing the profits of the generators but also maximizing the profits of the users.

4.2.1 Centralized Transactive Algorithm

We now state the centralized transactive algorithm based on replicator dynamics. We use k to denote the iterations in the algorithm.

Finally, in order to clarify the proposed algorithm the following flowchart is shown.

Algorithm 3 Centralized Transactive Control Algorithm

Executed by: Central Controller
Initialize: Assumption 2 and 7 are hold $\forall\, x$. Set $k = 0$
while $k \geq 0$ **do**
    Execute For all $x \in \mathcal{V}$
        Equation (4-14) and (4-15)
    Execute For all $x \in \mathcal{V}$
        Equation (4-16) and (4-17)
    Execute For all $x \in \mathcal{V}$
        Equation (4-12)
    Obtain $P_x(k+1)$
    Set $k = k + 1$;
end while

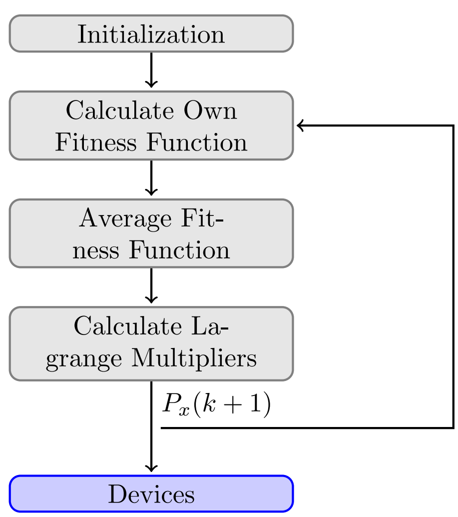

Figure 4-2: Flowchart of Distributed Transactive Control in Power Systems based on the Replicator Dynamics Algorithm

4.3 Asynchronous Transactive Control Algorithm

This algorithm has been developed to control a distribution system with microgrids inside. This algorithm uses as controller of the microgrids the Algorithm proposed in Section 4.2 and the algorithm presented in this section presents the second control layer that controls the distribution system, in this control layer is proposed an asynchronous control algorithm which is executed by the agents belonging to the distribution system. This section is based on [65].

Synchronization is a cornerstone to execute a wide variety of algorithms proposed for control of multi-agent systems [67, 68, 69, 70, 48, 27, 45]. It is common to assume that all devices in the grid obtain the data by other devices instantaneously. Nevertheless, these characteristics are unable to be accomplished [77, 78] due to the huge variety of devices in the grid [5, 6].

After microgrids execute Algorithm 2 they know if they need to buy power from the system or if they can sell it. In other words, once Algorithm 2 is executed $\overline{P}_{M_q}$ and $\underline{P}_{M_q}$ will be known, where $\overline{P}_{M_q}$ and $\underline{P}_{M_q}$ are the maximum and minimum power that an agent q can sell or buy to the network respectively. Once this data is revealed, the asynchronous algorithm starts to run. It is assumed that while this algorithm is running the values of $\overline{P}_{M_q}$ and $\underline{P}_{M_q}$ will not change.

The proposed algorithm is based on a primal-dual algorithm which uses the Lagrange multipliers μ. Lagrange multipliers are updated by a virtual agent which is connected with all agents in the power system. Power system agents send their estimates to the virtual agent at any time. Once the virtual agent has all agents estimations, it updates the Lagrange multiplier, and it is sent to the power system agents.

Initially, some minimal assumptions are made in order to execute this algorithm. We suppose that Assumption 3, 4, 5, 6, 7 and 9 are held. Control signals in power systems have to cover a large number of kilometers. Usually, it is assumed that communications and computations of algorithms are made synchronously. In practice, this is almost impossible because communication may interfere with each other, there may be slow data transmission, even loss of information when the channel data is over saturated.

In order to relax the traditional assumption that the control algorithm in the system is made synchronously, an asynchronous transactive control algorithm is proposed. In order to give a detailed overview of the algorithm an introduction to primal-dual optimization is presented first. Then, the algorithm used in this approach is presented.

4.3.1 Primal-Dual Optimization

Constraints in (3-19) can be organized as $\mathcal{R}P = \mathbf{r}$. It is important to highlight that constraint with the form $\mathcal{T}P \geq t$ can be written as $\mathcal{R}P = \mathbf{r}$ by using slack variables. Then, consider the problem stated below, it is equivalent to Problem (3-19)

Problem 3

$$\begin{aligned} \underset{P}{\text{minimize}} \quad & S_W(P) \\ \text{subject to} \quad & \mathcal{R}P = \mathbf{r}, \\ & P \in X \end{aligned}$$

where $\mathcal{R}$ is a matrix that includes global and local constraints and X is the set where P is feasible, i.e., $X = \cap_{x=1}^{V} X_x$, P is a vector where all the decision variables are stacked $P = [P_g, P_p, P_d, P_L, P_M]^\top$, $x \in \mathcal{V}$ represents any agent that executes the control algorithm in the power system. Also, P_x represents the power of agent x in the power system. Then, if Assumptions 2 and 4 are hold the Lagrangian associated with Problem 3 is

$$\mathcal{L}(P, \mu) = S_W(P) + \mu^\top(\mathcal{R}P - \mathbf{r}), \tag{4-19}$$

where μ is a vector that contains the Lagrange multipliers, $\mu \in \mathbb{R}^M_{\geq 0}$, where $\mathbb{R}^M_{\geq 0}$ is the nonnegative hyperorthant of $\mathbb{R}^M$ i.e., $\mathbb{R}_{\geq 0} = \{\mu \in \mathbb{R} \mid \mu \geq 0\}$, where M is the number of constraints of the power system. It is well defined that $\mathcal{L}(\cdot, \mu)$ is convex for all $\mu \in \mathbb{R}^M_{\geq 0}$ and $\mathcal{L}(P, \cdot)$ is concave for all $P_x \in X$.

Assumption 10 *(Slater's condition) Following [46], Exist a vector $\overline{P} \in \mathbb{R}^M_{\geq 0}$ such that $\mathcal{R}\overline{P} \leq 0$. $\overline{P}$ is called a Slater vector.*

A saddle point of $\mathcal{L}(P, \mu)$ is the maximization of $\mathcal{L}(P, \mu)$ over μ for all $\mu \in \mathbb{R}^M_{\geq 0}$ and the minimization $\mathcal{L}(P, \mu)$ over P_x for all $x \in \mathcal{V}$. Under Assumptions 4 and 5 a point denoted as $\left(\widehat{P}, \widehat{\mu}\right) \in X \times \mathbb{R}^M_{\geq 0}$ is a solution of Problem 3if and only if it is a saddle point of $\mathcal{L}(P, \mu)$ [43]. The saddle point condition satisfies that

$$\mathcal{L}(\widehat{P}, \mu) \leq \mathcal{L}(\widehat{P}, \widehat{\mu}) \leq \mathcal{L}(P, \widehat{\mu}) \tag{4-20}$$

Under Assumptions 4, 5 and 10 it is guaranteed that a saddle point $\left(\widehat{P}, \widehat{\mu}\right)$ exists [79]. In order to solve Problem 3, projection methods are used. Projection methods are useful to our problem because it is adaptable to decentralized problems [80]. However, this method requires that

$\nabla_x \mathcal{L}(\cdot, \mu)$ and $\nabla_\mu \mathcal{L}(P_x, \cdot)$ be Lipschitz. To be Lipschitz implies that the function domain is bounded, however it is possible to see that $\nabla_x \mathcal{L}(\cdot, \mu)$ is not bounded because initially $\mu \in \mathbb{R}^M_{\geq 0}$. To adjust this situation and use projection methods, a convex, compact and nonempty subset of $\mathbb{R}^M_{\geq 0}$ is proposed, i.e., $X_\mu \subset \mathbb{R}^M_{\geq 0}$.

Lemma 3 *Let* $\overline{P} \in X$ *a Slater vector. Then,*

$$\widehat{\mu} \in X_\mu := \left\{ \mu \in \mathbb{R}^M_{\geq 0} : \|\mu\|_1 \leq \frac{S_W(\overline{P}) - q(\widehat{\mu})}{\min_{1 \leq i \leq M}\{-f_i(\overline{P})\}} \right\}$$

Proof. (Inspired in [81]). It is possible define the dual function of $\mathcal{L}(P, \mu)$ taking into account [43, Proposition 2.6.1] as

$$\begin{aligned} \underset{\mu}{\text{maximize}} \quad & q(\mu) \\ \text{subject to} \quad & \mu \in \mathbb{R}^M_{\geq 0}, \end{aligned}$$

where $q(\mu) = \inf_{P \in X} \left(S_W(P) + \mu^\top (\mathcal{R}P) \right)$. Let $\overline{P}$ a Slater vector. Using Slater property,(cf. Equation (4-20)), we have

$$q(\mu) \leq \mathcal{L}(\widehat{P}, \widehat{\mu}) \leq \mathcal{L}(\overline{P}, \widehat{\mu})$$

Considering that $\widehat{\mu}^\top (\mathcal{R}P)$ can be rewrote as $\sum_{i=1}^{M} \widehat{\mu}_i f_i(P)$, where $f_i(P)$ are the constraints $(\mathcal{R}P)$. It follows that

$$\begin{aligned} q(\mu) &\leq S_W(\overline{P}) + \sum_{i=1}^{M} \widehat{\mu}_i f_i(\overline{P}) \\ q(\mu) &\leq S_W(\overline{P}) + \left(\sum_{i=1}^{M} \widehat{\mu}_i \right) \max_{1 \leq i \leq M} \{f_i(\overline{P})\} \\ \left(\sum_{i=1}^{M} \widehat{\mu}_i \right) \min_{1 \leq i \leq M} \{-f_i(\overline{P})\} &\leq S_W(\overline{P}) - q(\mu) \end{aligned}$$

Therefore, considering that

$$q(\widehat{\mu}) = \inf_{P \in X} \left(S_W(P) + \sum_{i=1}^{M} \widehat{\mu}_i f_i(P) \right) \leq S_W(\overline{P}) + \sum_{i=1}^{M} \widehat{\mu}_i f_i(\overline{P})$$

It follows that

$$\|\widehat{\mu}\|_1 \leq \sum_{i=1}^{M} \widehat{\mu}_i \leq \frac{SW(\overline{P}) - q(\widehat{\mu})}{\min_{1 \leq i \leq M}\{-f_i(\overline{P})\}} < \infty \tag{4-21}$$

Using $\widehat{\mu_i} \geq 0$ for all $i \in M$ is possible define

$$X_\mu = \left\{ \mu \in \mathbb{R}^M_{\geq 0} : \|\mu\|_1 \leq \frac{SW(\overline{P}) - q(\widehat{\mu})}{\min_{1\leq i\leq M}\{-f_i(\overline{P})\}} \right\} \tag{4-22}$$

Once defined X_μ. $\nabla_a \mathcal{L}(\cdot, \mu)$ is Lipschitz due to $\mu \in X\mu$ is a bounded convex set. ■

Thus, with the definition of the set limit, it is possible to use the projection operator in our algorithm since X_μ is a closed convex set.

4.3.2 Asynchronous Transactive Control Algorithm

Now, we present the asynchronous transactive control algorithm. This algorithm achieves the optimum social welfare for the power system asynchronously through the use of primal-dual optimization. We assume that agents compute primal updates of each one and the virtual agent computes the dual updates. The algorithm states that first the agents compute its updates and then the virtual agent computes the dual update. As it is an asynchronous algorithm we state the following clocks for the agents (k^x), each agent has a different clock, and for the virtual agent (k^v).

Initially, it is known that each agent can update its utility or cost function by having only its previous status and the previous status of dual updates as shown in (4-23). If the agent has not received an update of the dual variable when running (4-23), then it should stop its update and keep the previous value. The update law executed by agents in the power system is

$$P_x(k^x+1) = \begin{cases} \mathbb{P}_{X_x}\left[P_x(k^x) - \alpha \nabla_x \mathcal{L}(P,\mu)\right] & \text{if } \mu(k^x) \neq \mu(k), \\ P_x(k^x) & \text{if } \mu(k^x) = \mu(k), \end{cases} \tag{4-23}$$

where α is the stepsize, and ∇_x represents the partial derivative with respect to x. Be defined $[P_x(k^x) - \alpha \nabla_x \mathcal{L}(P,\mu)] = \beta(k^x)$. Then, the projection operation is

$$\mathbb{P}_{X_x}\left[\beta\left(k^x\right)\right] = \begin{cases} \overline{P}_x, & \text{if} \quad \beta\left(k^x\right) > \overline{P}_x, \\ \underline{P}_x, & \text{if} \quad \beta\left(k^x\right) < \underline{P}_x, \\ \beta\left(k^x\right) & \text{Otherwise.} \end{cases} \tag{4-24}$$

where $\underline{P}_x$ and $\overline{P}_x$ are the minimum and maximum power allowed by each agent x. Once each agent calculates (4-23) is expected that they sends to the cloud its estimates at time arbitrary $k^x + \tau$ and its expected to arrive to the virtual agent in $k^x + \tau^v$, where τ and τ^v are arbitrary delay times. It is important to highlight that $\tau < \tau^v$.

After the virtual agent has received all the new estimates for all agents in the network, the virtual agent computes the following update

$$\mu(k^v + 1) = \mathbb{P}_{X_\mu}\left[\mu(k^v) - \alpha_\mu\left(\nabla_\mu \mathcal{L}(P,\mu) - \mu(k^v)\right)\right], \tag{4-25}$$

where

$$\mathbb{P}_{X_\mu}\left[\mu(k^v) - \alpha_\mu\left(\nabla_\mu \mathcal{L}(P,\mu) - \mu(k^v)\right)\right] = \begin{cases} \mu_{max}, & \text{if} \quad \mu\left(k^v\right) > \mu_{max}, \\ 0, & \text{if} \quad \mu\left(k^v\right) < 0, \\ \mu(k^v) - \alpha_\mu\left(\nabla_\mu \mathcal{L}(P,\mu) - \mu(k^v)\right), & \text{Otherwise.} \end{cases} \tag{4-26}$$

where μ_{max} is the upper bound of the set X_μ. Then, the virtual agent sends its estimates to each agent and the algorithm start again. The algorithm of the transactive control in the power system is summarized in Algorithm 4.

Algorithm 4 Distributed Transactive Control Algorithm for Power Systems

Executed by: Agents $x \in \mathcal{V} = \{1, 2, 3, \ldots, V\}$
Require: $\nabla_x \mathcal{L}(P,\mu)\ \forall\ x \in \mathcal{V}$ and $\nabla_\mu \mathcal{L}(P,\mu)$
Initialize: Assumptions 6 and 9 are hold $\forall\ x \in \mathcal{V}$. Set $k^v = 0$ and $k^x = 0\ \forall\ x \in \mathcal{V}$
Initialize: $P_x(0)\ \forall\ x \in \mathcal{V}$ and $\mu(0)$
while $k^v \geq 0$ **do**
  for all $x \in \mathcal{V}$ **do**
    Execute Equation (4-23)
    if $x \in \mathcal{Q}$ **then**
      Execute Algorithm 2
    end if
    Send $P_x(k^x)$ to the virtual agent
    $k^x = k^x + 1;$
  end for
  Execute Equation (4-25)
  Send $\mu(k^v)$ to each agent
  $k^v = k^v + 1;$
end while

Figure **4-3** shows the flowchart of the Algorithm 4 .First an initialization is required, then each agent in a time (k^x) updates its estimates using (4-23). Once (4-23) is update, each agent sends its estimates to the virtual agent. The virtual agent executes (4-25) when it receives all estimates from the agents $x \in \mathcal{V}$. Finally the dual variable update is sent to all agents $x \in \mathcal{V}$. If the power generated does not fulfill constraint (3-19b) the algorithm returns to step two and it executes again the algorithm until constraint (3-19b) is fulfilled. If the constraint is fulfilled, the optimization Algorithm has reached the optimum social welfare for the power system and ends.

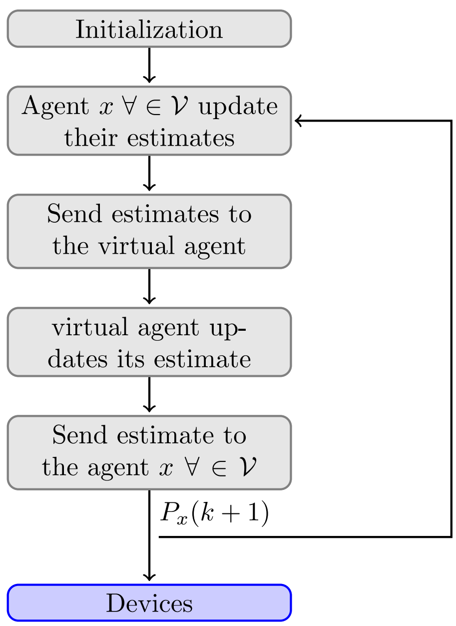

Figure 4-3: Flowchart of the asynchronous transactive algorithm used by the power system

4.4 Distributed Constrained Transactive Optimization Control Based on Population Dynamics

In this section, it is presented the last algorithm of this work. A distributed control algorithm considering renewable energy integration and flexible power demand using transactive control based on population games is developed. This section is based on [66], transactive control is responsible for the determination of how much energy each user will consume and how much power each generator have to generate in order to satisfy the demand while operating the system most economically.

Constraints in Problem (3-19) could be represented in the form $\mathcal{R}^\top P = \mathbf{r}$, where the decision variable are stacked in a vector as $P = [P_g, P_p, P_d, P_L, P_M]^\top = [P_1, P_2, P_3, \dots, P_V]^\top$, and $\mathcal{R}$ is a vector indicating the sign depending if the entry is a generator (+) or a smart load (-). In this work, we use a distributed replicator dynamics equation [29] to solve the social welfare problem dynamically associated with a transactive control problem between microgrids, distribution system, and consumers. This model is a dynamic resource allocation problem among V agents (generator and smart loads) given a finite amount of power $\mathbf{r}$.

The feasible set is defined as the set of possible trajectories of the system restricted by Problem (3-19) constraints as

$$\Delta = \{p \in \mathbb{R}^n : \mathcal{R}^\top P = \mathbf{r}.\}$$

Some assumptions have to be considered to assure optimality conditions and the existence of solutions of problem stated in (3-19). In this algorithm we consider that assumptions 3, 7 and 9. We use the Lemma 4 in order to establish the conditions to optimal points

Lemma 4 *A solution of problem* (3-19)*, P^* belonging to the feasible set Δ, is an optimal solution if and only if $\nabla_x S_W(P_x^*) = \nabla_y S_W(P_y^*)$ for all x, y.*

Notice that ∇ stands for the Jacobian of a function. In order to relate the optimality condition in Lemma 4 with the Distributed Replicator Dynamics (2-20), we use the Lagrangian function associated to optimization problem (3-19) as

$$\mathcal{L}(P, \mu) = S_W(P) - \mu^\top \left(\mathcal{R}^\top P - \mathbf{r}\right) \tag{4-27}$$

where μ are the Lagrange multipliers, S_W is the social welfare function, and $\mathcal{R}^\top P - \mathbf{r}$ are the constraints in the system. The optimal solution (P^*, μ^*), which can be found by the Kuhn-Tucker first-order conditions for maximization establishes that P^* is a unique solution to (3-19) if (P^*, μ^*) is a saddle point of $\mathcal{L}(P, \mu)$ [43].

4.4.1 Distributed Replicator Dynamics and Potential Games

One of the main attributes of potential games is the existence of a single scalar-valued function, called potential function, which captures all relevant information about the payoffs of the agents.

Assuming there exists a continuously differentiable potential function $h : \mathbb{R}^n_{>0} \to \mathbb{R}$, then a potential game satisfies the following relationship for the fitness function for each agent

$$\frac{\partial h(p)}{\partial p_x} = F_x(p) \quad \text{for all } \; x. \tag{4-28}$$

Equation (4-28) implies that the population game must satisfy the external symmetry defined as

$$\frac{\partial F_x}{\partial p_y} = \frac{\partial F_x}{\partial p_y} \quad \text{for all } \; x, y. \tag{4-29}$$

In potential games, there is a property that relates Nash equilibrium to local maximizers of potential function [50], it is stated in the following proposition.

Proposition 5 *A potential game with potential function h satisfies that Nash equilibrium of the potential game is equal to the solution of the Kuhn-Tucker conditions of the optimization problem maximize $h(p)$ subject to feasible set Δ [50].*

On the other hand, we are interested in the stability of the game. A significant result in population games about stability states that a population game satisfying (4-28) is a stable game if the potential function h is concave [50].
Having introduced the main concepts of potential population games, it is possible to present a definition for the fitness function relating the population game as a potential game guaranteeing optimality and stability conditions. Therefore, the fitness functions are defined as follows

$$F_x(P_x) = \nabla_P \mathcal{L} = \nabla_P S_W - \mathcal{R}\mu \tag{4-30}$$

therefore the dynamic equation is

$$\dot{p_x} = \left(\frac{1}{P_L}\right) p_x \left(F_x(p_x) \sum_{y \in \mathcal{N}_x} p_y - \sum_{y \in \mathcal{N}_x} p_y F_y(p_y) \right), \tag{4-31}$$

where x and y are agents in the power system, which introduces an extended system including the Lagrange multipliers dynamics with fitness function defined as

$$F_m(\mu_m) = \nabla_\mu \mathcal{L} = -(\mathcal{R}^\top P - \mathbf{r}), \tag{4-32}$$

and dynamic equation

$$\dot{\mu}_m = \mu_m \left(F_m(\mu_m) \sum_{b \in \mathcal{N}_m} \mu_b - \sum_{b \in \mathcal{N}_m} \mu_b F_b(\mu_b) \right) \tag{4-33}$$

where $m \in \mathcal{M} = \{1, 2, 3, \ldots, M\}$ is the number of constraints and b is any constraint in the neighborhood of m. In the next subsection it is presented the optimality and stability analysis of the extended dynamical system including $p_x(t)$ and $\mu_m(t)$.

Algorithm 5 Distributed Constrained Transactive Optimization Control based on Population Dynamics

Executed by: Agents $x \in \mathcal{V} = \{1, 2, 3, \ldots, V\}$
Require: β
Initialize: Set $k = 0 \; \forall \; x \in \mathcal{V}$
Initialize: $P_x(0) \; \forall \; x \in \mathcal{V}$ and $\mu(0) \; \forall \; m \in \mathcal{M}$
while $k \geq 0$ **do**
    for all $m \in \mathcal{M}$ **do**
        Execute Equation (4-33)
        Send to $x \in \mathcal{N}_m$
    end for
    for all $x \in \mathcal{V}$ **do**
        Execute Equation (4-31)
        Send to $m \in \mathcal{N}_x$
    end for
    Set $k = k + 1$
end while

4.4.2 Optimality and Stability Analysis

Consider the extended population game defined by distributed replicator dynamics (4-31) and Lagrange multiplier dynamics (4-33) with fitness functions (4-30) and (4-32), respectively. It is necessary to show that the extended population game is a potential game to guarantee optimality through Lemma 4 and Proposition 5. Then, to show the game is stable, we have to verify that the potential function h is concave and twice continuously differentiable.

Theorem 2 *Assume a convergence constant $\beta \in \mathbb{R}$. Furthermore, let $P_x(k)$, with $x \in \mathcal{V}$, be the set points generated by Algorithm 5. Then, $P_x(k)$ with $x \in \mathcal{V}$ converges to the optimal solution P_x^* with $P_x^* \in X$, that is*

$$\lim_{k \to \infty} P_x(k) = P_x^*.$$

Proof.

Since we have defined fitness functions as (4-30) and (4-32), by definition it is clear that a potential function for the population game (2-20) is $h(P, \mu) = \mathcal{L} = S_W(P) - \mu^\top(\mathcal{R}P - \mathbf{r})$ and considering the form of the social welfare function S_W defined in (3-18), it can be shown that the game satisfies the external symmetry (4-29). When the optimality condition in Lemma 4 is reached then $F_x(p_x) = F_y(p_y)$ for all x, y and it is noticed that in the distributed replicator dynamics (2-20) we have

$$\left(F_x(P_x)\sum_{y\in\mathcal{N}_x}P_y-\sum_{y\in\mathcal{N}_x}P_yF_y(P_y)\right)=\left(F_x(P_x)\sum_{y\in\mathcal{N}_x}P_y-\sum_{y\in\mathcal{N}_x}P_yF_x(P_x)\right)$$
$$=\left(F_x(P_x)\left(\sum_{y\in\mathcal{N}_x}P_y-\sum_{y\in\mathcal{N}_x}P_y\right)\right)$$
$$=0,$$

which implies that (2-20) reaches an equilibrium point. ■

For the stability analysis, we need to verify that the potential function $h(P,\mu)$ is concave to assure the stability of the algorithm.

To prove that the function is concave we need to check that the Hessian matrix of $h(P,\mu)$ is negative semidefinite , i.e, $\nabla^2 h(P,\mu)\leq 0$. The Jacobian is obtained as

$$\nabla h(P,\mu)=\begin{bmatrix}\nabla_P h\\ \nabla_\mu h\end{bmatrix}=\begin{bmatrix}\nabla S_W-\mu^\top\mathcal{R}\\ -(\mathcal{R}\mu+\mathbf{r})\end{bmatrix}.$$

Hence, the Hessian is obtained as

$$\nabla^2 h(p,\mu)=\begin{bmatrix}\nabla_p^2 h\\ \nabla_\mu^2 h\end{bmatrix}=\begin{bmatrix}\nabla^2 S_W\\ 0\end{bmatrix}.$$

The problem to check the semidefiniteness of the Hessian of h reduces to check the Hessian of $\nabla^2 S_W$. Recall that S_W is as in (3-18), and deriving twice it is obtained the Hessian as

$$\nabla^2 S_W=\left(\sum_{a=1}^{A}\beta_{d_a}+\sum_{s=1}^{S}\beta_{L_s}\right)-\left(\sum_{r=1}^{R}\beta_{P_r}+\sum_{n=1}^{N}\beta_{g_n}\right).$$

As a result to guarantee that $\nabla^2 S_W\leq 0$, we obtain a relationship between the coefficients of cost functions of the generators and utility functions of the smart loads as follows

$$\left(\sum_{a=1}^{A}\beta_{d_a}+\sum_{s=1}^{S}\beta_{L_s}\right)\leq\left(\sum_{r=1}^{R}\beta_{P_r}+\sum_{n=1}^{N}\beta_{g_n}\right)\tag{4-34}$$

If condition (4-34) is satisfied then the algorithm based on the DRD is stable.

In this section, it has been analyzed the main features of the proposed algorithm in term of optimality and stability.The flowchart of the proposed algorithm is shown in Figure **4-4**.

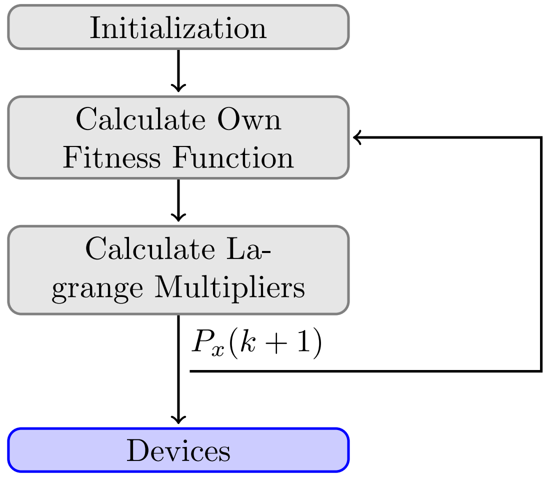

Figure 4-4: Flowchart of the Distributed Constrained Transactive Optimization Control based on Population Dynamics

In the next section, several case studies are presented to illustrate the effectiveness of the algorithms proposed to solve the social welfare problem.

5 Simulation Results

In this chapter, we test the effectiveness of all the proposed algorithms stated in Chapter 4. We test the algorithms presented in Section 4.1 and Section 4.2 in a WSCC 9 bus. Furthermore, we test the algorithms presented in Section 4.3 and Section 4.4 in a modified IEEE 30 bus system, where we include microgrids inside this system.

5.1 Simulations of Microgrids

To simulate and make an effective comparison of the different algorithms is necessary to make two study cases. First of them is dedicated to algorithms that are executed in microgrids,i.e., Algorithms in Sections 4.1 and 4.2.

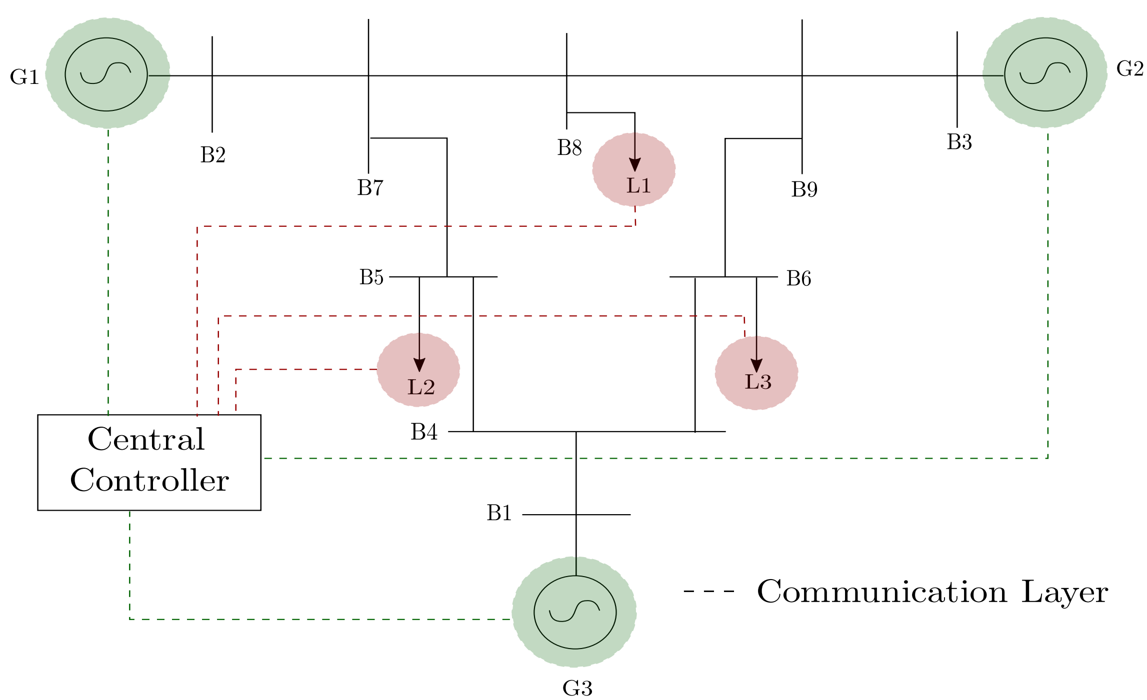

Figure 5-1: Illustrative general scheme of a microgrids adopted from a WSCC 9-bus.

We simulate a Microgrid using a WSCC 9-Bus System, as it is shown in Figure **5-1**. In this power system case, there are three generators, which we adopted two as traditional generators and one as photovoltaic generators. There are also three loads; we choose that two of them being adjustable loads and the other is a shiftable load as is shown in Table **5-1**.

Table 5-1: System parameters to study case in Microgrids.

Agents Parameters		
Agent	**Power Max [W]**	**Power Min [W]**
Conventional Generator 1	3000	100
Conventional Generator 2	4000	100
Photovoltaic Generator 1	3000	100
Shiftable Load 1	2000	1000
Shiftable Load 2	3000	1500
Adjustable Load 1	3000	1500

Notice that the base load is the minimum power that the load can consume and the set between the minimum or maximum load is where the load behavior is a smart load. In the simulation, the minimum power of all loads rise up 500 $[W]$ as is shown in Figure **5-4** and Figure **5-5**. However, the power maximum also rises 500 $[W]$, then the smart loads in the system do not vary its capacity to consuming power from the network.

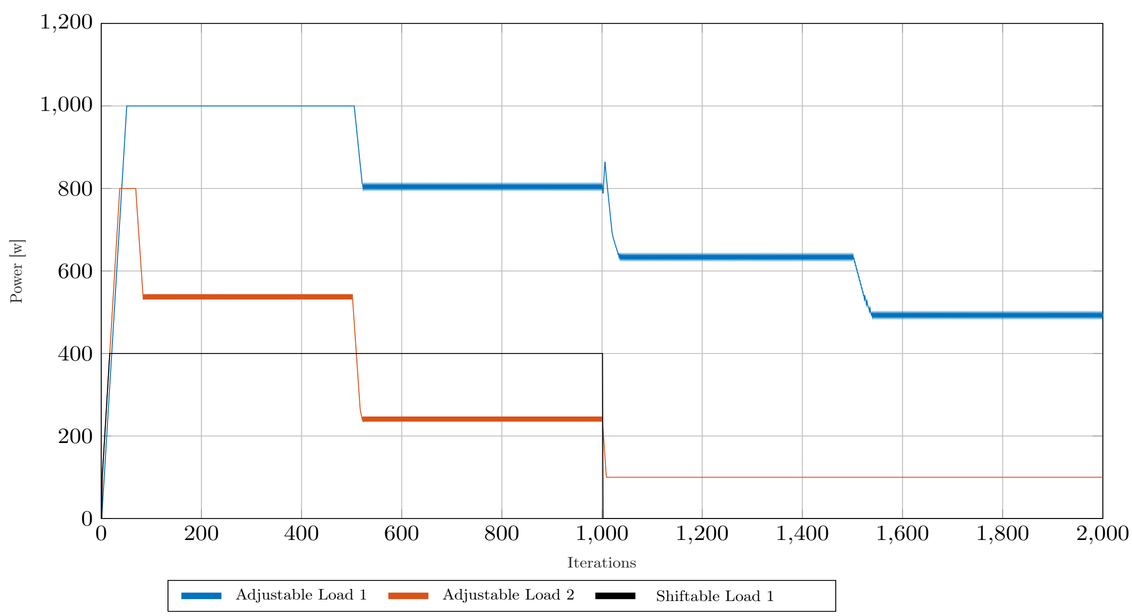

Figure 5-2: Algorithm 1: Smart Load results for simulation in WSCC 9 bus system

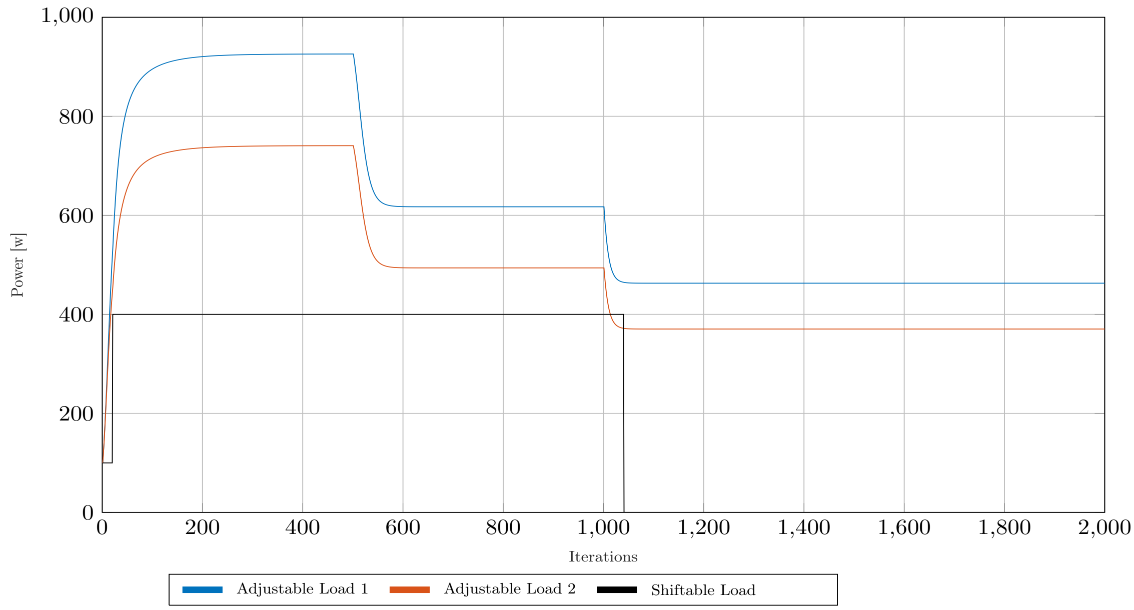

Figure 5-3: Algorithm 2: Smart Load results for simulation in WSCC 9 bus system

It is possible to see that adjustable load decreases its amount of power consumed in 500 and 1000 iterations due to the increase of base load which is a predictable behavior in both algorithms which is shown in Figure **5-2** and Figure **5-3**, because smart loads should consume less power when the power system is in a load peak . Also, it is possible to see that shiftable load after 1000 iterations does not consider viable to connect to the microgrid the shiftable load for both algorithms which is also shown in Figure **5-2** and Figure **5-3**.

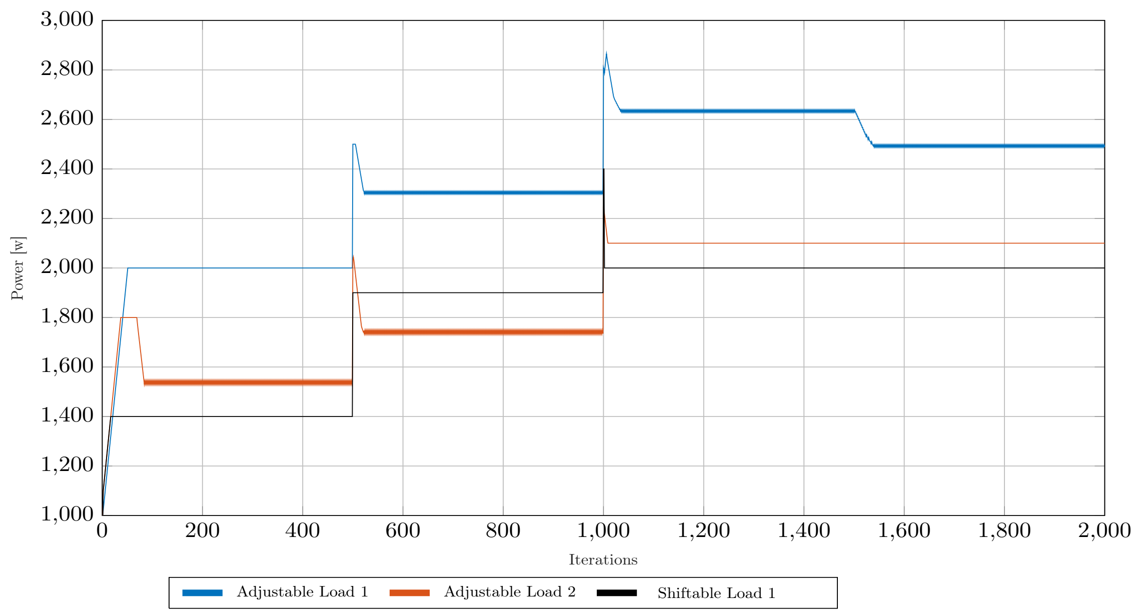

Figure 5-4: Algorithm 1: Total load results for simulation in WSCC 9 bus system.

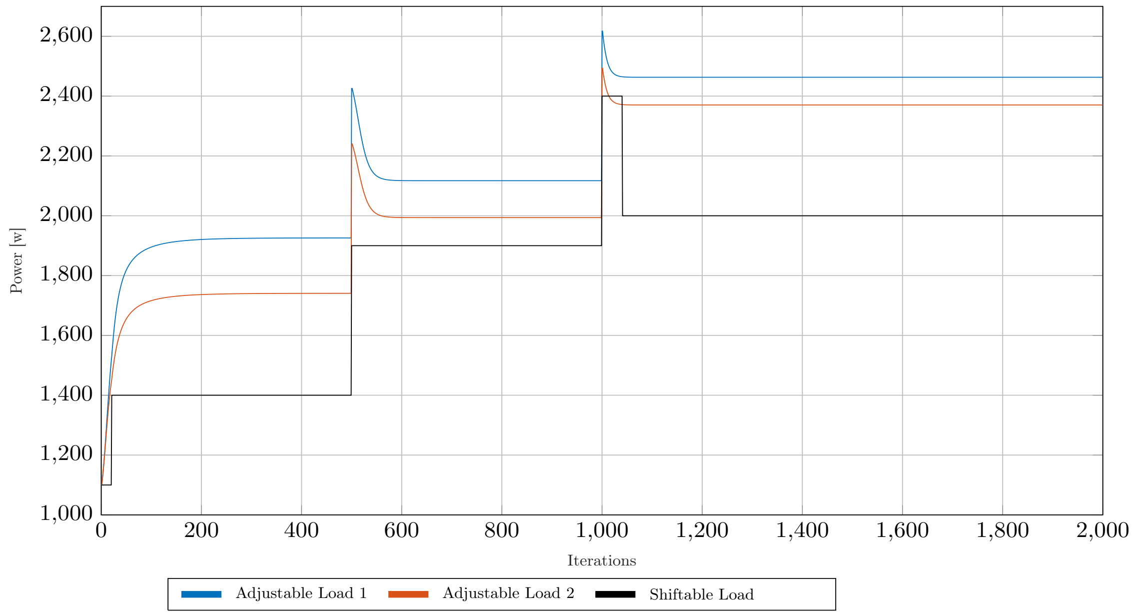

Figure 5-5: Algorithm 2: Total load results for simulation in WSCC 9 bus system.

In Figure **5-4** and Figure **5-5** is possible to see that the load associated with base load increases in 500 and 1000 iterations. Furthermore is shown that adjustable loads turn off some load every time this happens. Furthermore, the shiftable load only changes the load consumed to the power system when the threshold is reached. This occurs near to 1000 iterations in both algorithms. In Figure **5-4** is possible to see that shiftable load turn off the load, while in Figure **5-5** is possible to see that when adjustable load 1 and 2 reach the optimum state, shiftable loads change the load consumed.

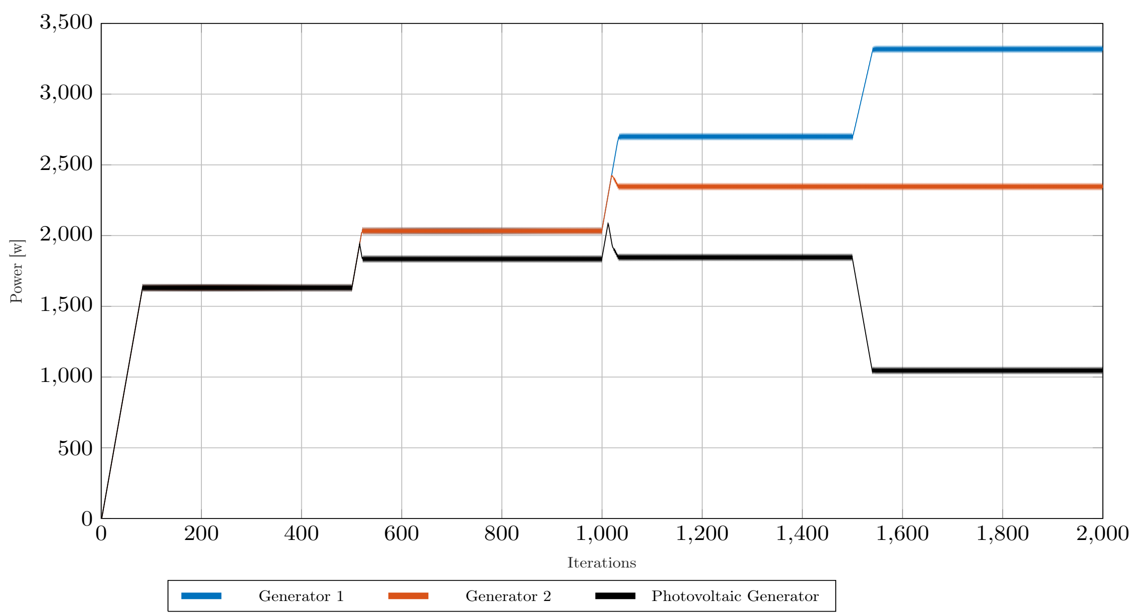

Figure 5-6: Algorithm 1: Generation results for simulation in WSCC 9 bus system.

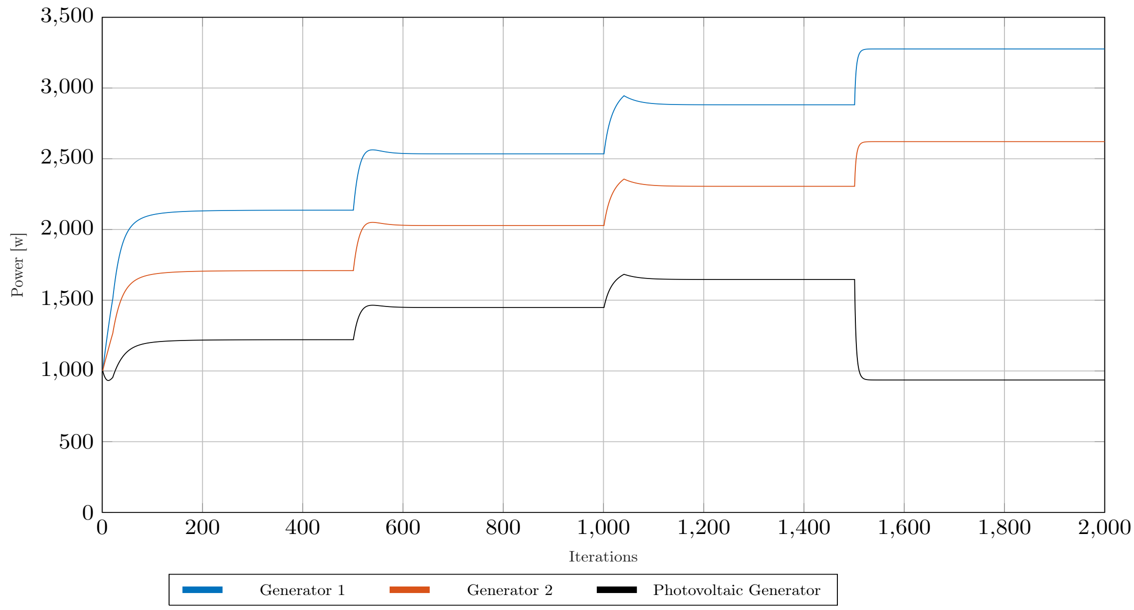

Figure 5-7: Algorithm 2: Generation results for simulation in WSCC 9 bus system.

With respect to the generators agent it is possible to see that all of them respond to the changes in the load balance, which in both cases is reached in 15 iterations for Algorithm presented in Section 4.1 and 25 iterations for Algorithm presented in Section 4.2 as is shown in Figure **5-6** and Figure **5-7** respectively. Furthermore, in 1500 iterations we change SR for the photovoltaic generator. This changes the maximum power available for the photovoltaic generator, reducing it power availability to 1000W. In the Algorithm presented in Section 4.1 this change the generators power set points and adjustable loads power set point. Adjustable loads change its consume due to the change in the average consensus used to set the power set point as is shown in Figure **5-6**. While in Algorithm presented in section 4.2 power set points for generators change, but power for smart loads do not change. This is due to average fitness function is only taken fitness function of smart loads, then the change in generators power set point does not affect the smart loads as is shown in Figure **5-7**.

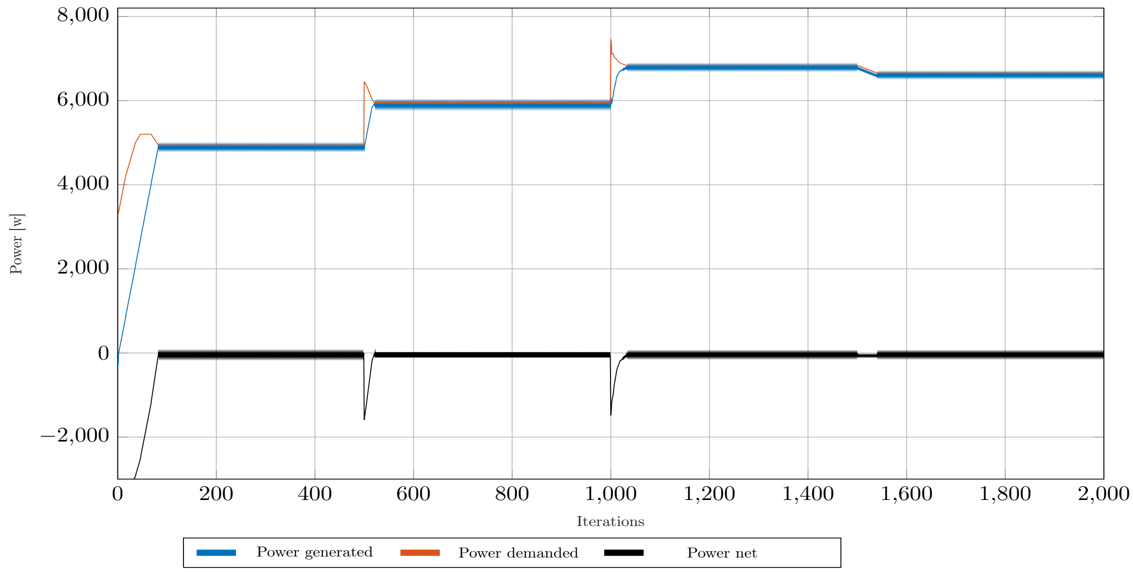

Figure 5-8: Algorithm 1: Power net in Microgrid for simulation in WSCC 9 bus system.

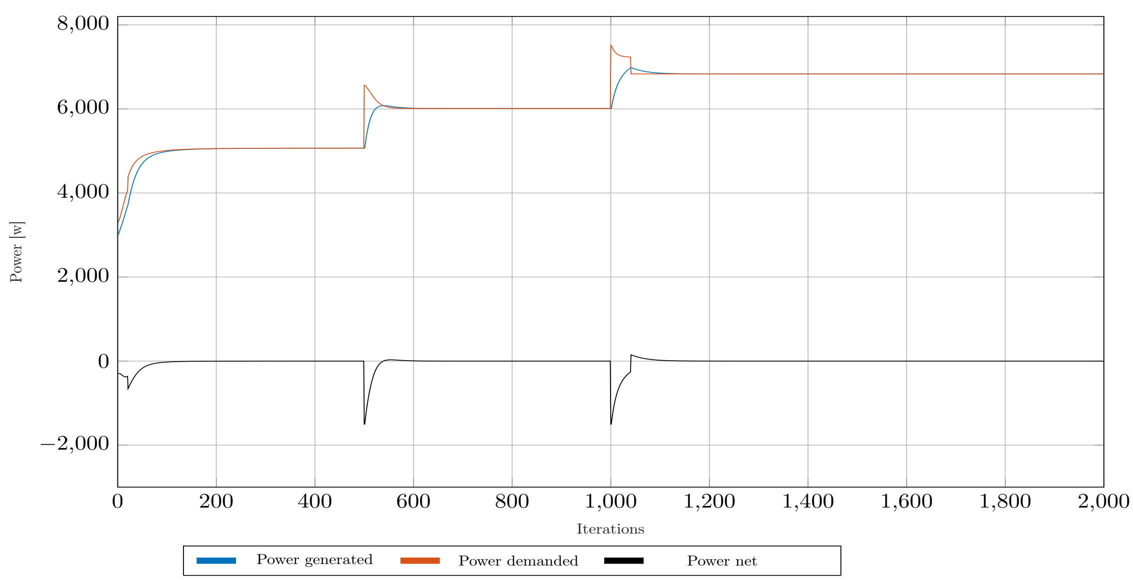

Figure 5-9: Algorithm 2: Power net in Microgrid for simulation in WSCC 9 bus system.

Finally, Figure **5-8** and Figure **5-9** show that both algorithms maintain the power balance in the network. Also, it is possible to see that the convex optimization based algorithm fulfill this constraint in less time despite their distributed nature. However, this algorithm has oscillations around the optimum point; this makes it have a difference of $\pm$ 1 % to power demanded. In Figure **5-9** is possible to see that the game theory algorithm has more smooth behavior, In consequence, the algorithm reaches the optimum point, i.e., the difference between the power demanded and power generated is $0[W]$.

5.2 Simulations of Distributed System with Microgrids

In this section, we simulate a power system based on an IEEE 30-bus system. A microgrid, five generators, and six smart loads. Microgrid agent has two traditional generators, one photovoltaic generator and three smart loads. We seek to maximize the social welfare of generators, consumers and the microgrid. Agents have limited power generation and demand as it is shown in Table **5-2**.

Table 5-2: IEEE 30 bus system Parameters

Agents Parameters	
Microgrid (MG1)	[W] [W]
Power (max, min) gen 1	5000, 100
Power (max, min) gen 2	6000, 100
Power (max, min) gen 3	7000, 100
Power (max, min) con 1	2000, 1000
Power (max, min) con 2	3000, 2000
Power (max, min) con 3	4000, 3000
Power System	[W] [W]
Power (max, min) gen 1	14000, 100
Power (max, min) gen 2	15000, 100
Power (max, min) gen 3	14000, 100
Power (max, min) gen 4	15000, 100
Power (max, min) gen 5	14000, 100
Power (max, min) con 1	6333, 3333
Power (max, min) con 2	8333, 3333
Power (max, min) con 3	6333, 3333
Power (max, min) con 4	8333, 3333
Power (max, min) con 5	6333, 3333
Power (max, min) con 6	8333, 3333
Power (max, min) MG 1	P_{M_q}, 10

Initially, each agent in the power system initializes its power states, i.e., $P_x(0)\ \forall\ x$. Then, Algorithm 4 starts, agents in the power system updates its states, if the agent is a microgrid, the agent q executes Algorithm 2. After the microgrid optimizes its state, the dual update is computed by the virtual agent.

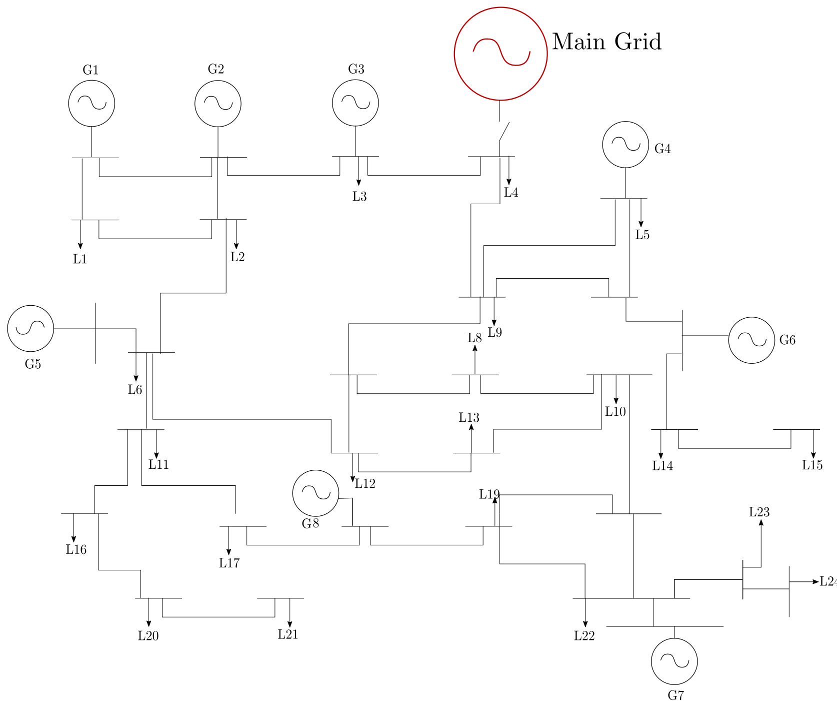

Figure 5-10: Illustrative general scheme of a Distribution system with microgrids adopted from a IEEE 30-bus

Notice that the base load is the minimum power that the load can consume and the set between the minimum or maximum load is where the load behavior is a smart load. We use several states in the power system, in the beginning, the loads request to the system 20.000 $[W]$ divided into each load. Once stabilized the power system, in 700 iterations we raise the load to 35.000 $[W]$, in other words, the base load of each load rises until 5.833 $[W]$. Finally, another load change is proposed raises the base load of each consumer to 9.166 $[W]$, for a total in the system of 55.000 $[W]$.

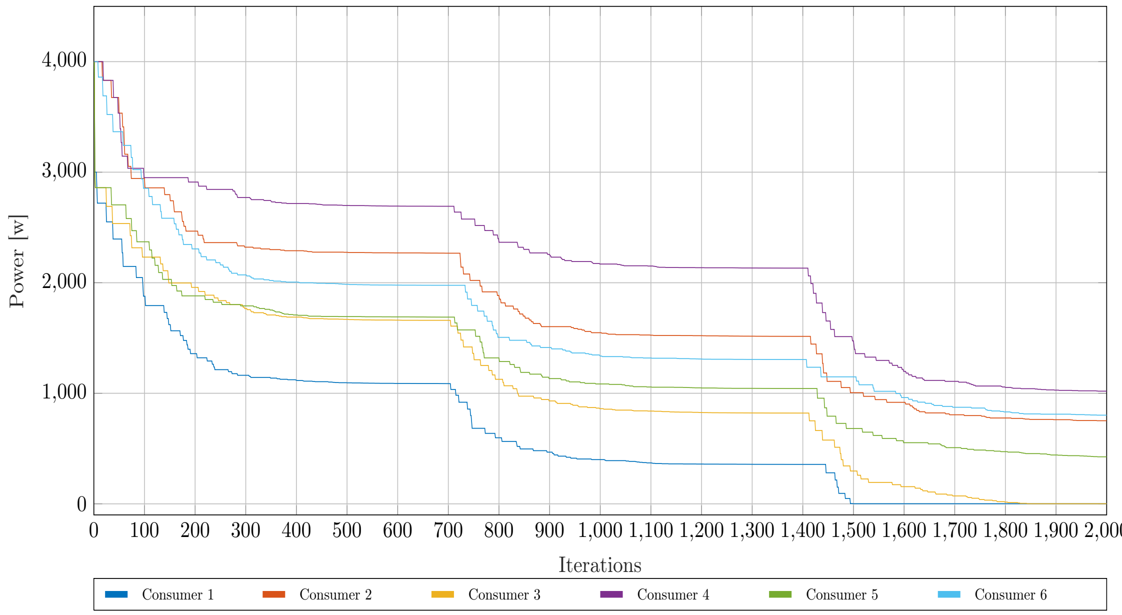

Figure 5-11: Algorithm 3: Smart Load results for simulation in IEEE 30 bus system

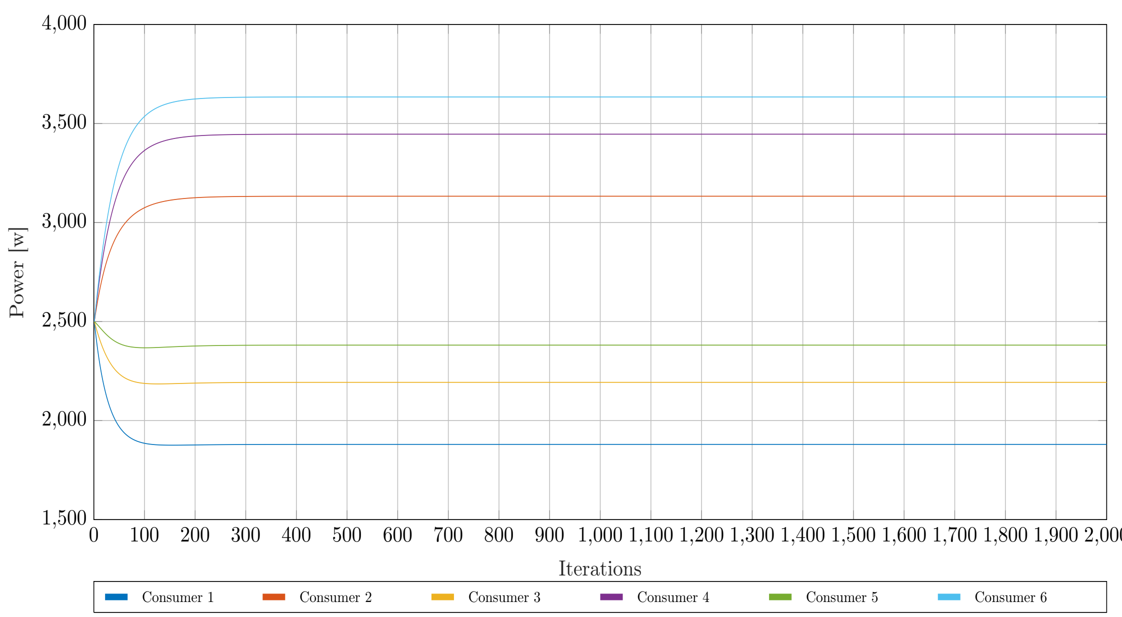

Figure 5-12: Algorithm 4: Smart Load results for simulation in IEEE 30 bus system

It is possible to see that adjustable load decreases its amount of power consumed in 500 and 1000 iterations due to the increase of the base load which is predictable behavior because smart loads should consume less power when the power system is in a load peak. However, this only occurs in Figure **5-11**. In Figure **5-12** is shown that load rises until a point and maintain its stability until the end of the simulation. This behavior is done because the algorithm starts within the set where the solution is feasible. When changing the optimization problem, that is, when changing the load that the power system has to supply, the algorithm needs to be reconfigured, which is a clear disadvantage against the asynchronous algorithm.

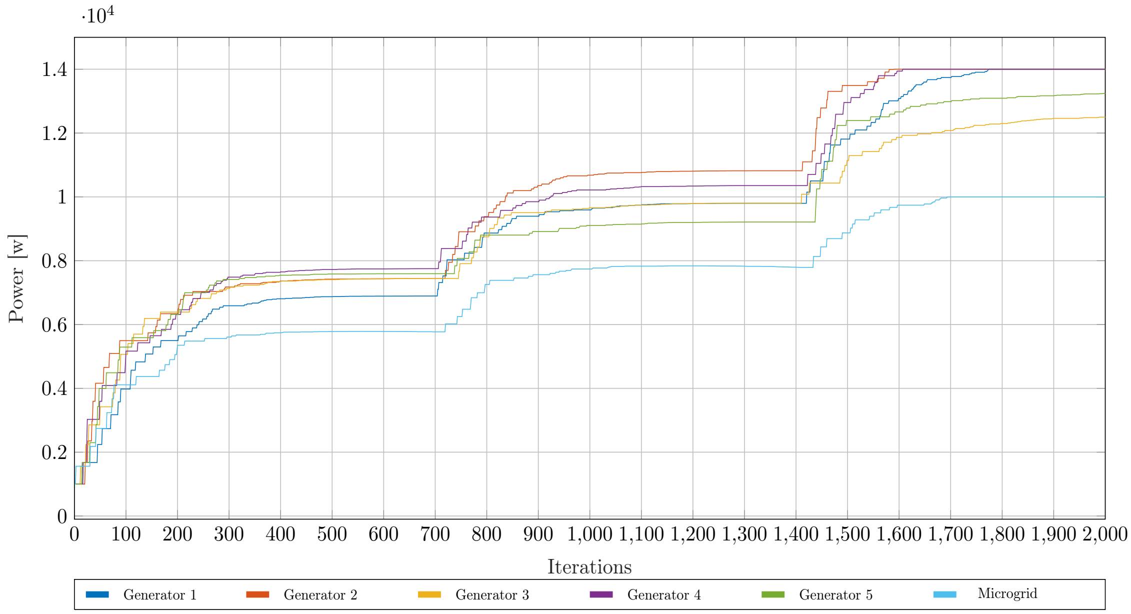

Figure 5-13: Algorithm 3: Generation results for simulation in IEEE 30 bus system

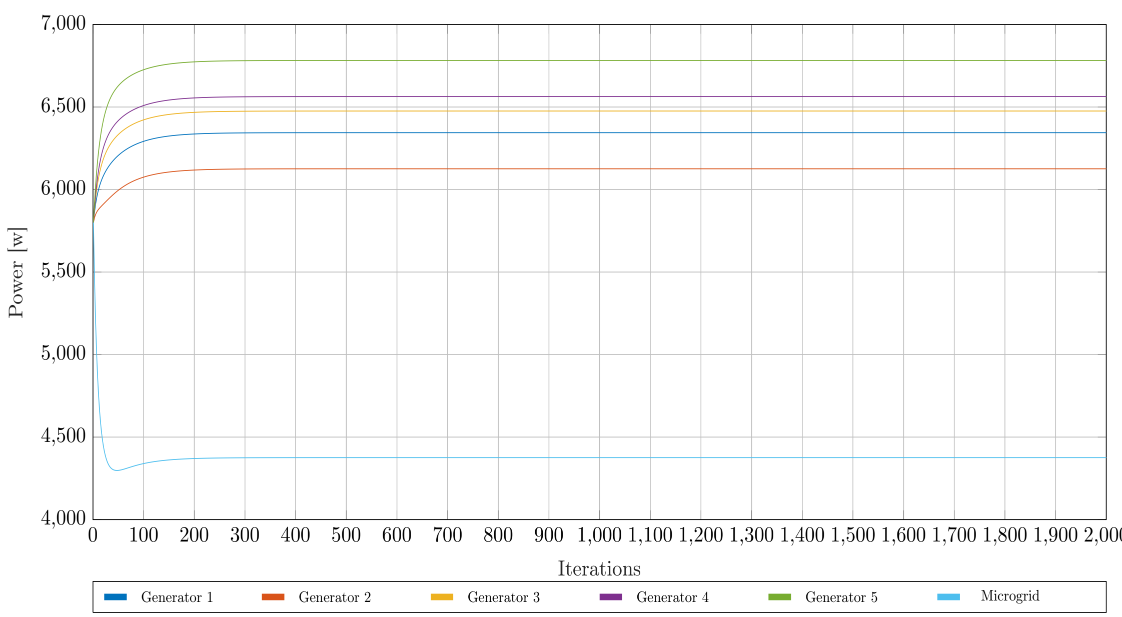

Figure 5-14: Algorithm 4: Generation results for simulation in IEEE 30 bus system

Concerning the generators agent, it is possible to see that all of them respond to the changes in the load balance, which in Figure **5-13** is reached in 300 iterations for Algorithm 3. As well as in Figure **5-12**, the generators in Figure **5-14** do not follow the power requested by the grid making an imbalance that is dangerous for the correct operation of a power system. However, Algorithm 4 reaches smooth to the global optimum taking into account the microgrid inside it. The microgrid agent is ever the less requested for power; however, both algorithms include it inside the economic dispatch making both algorithms meet their main objective, to include microgrids in the economic dispatch of a distribution system.

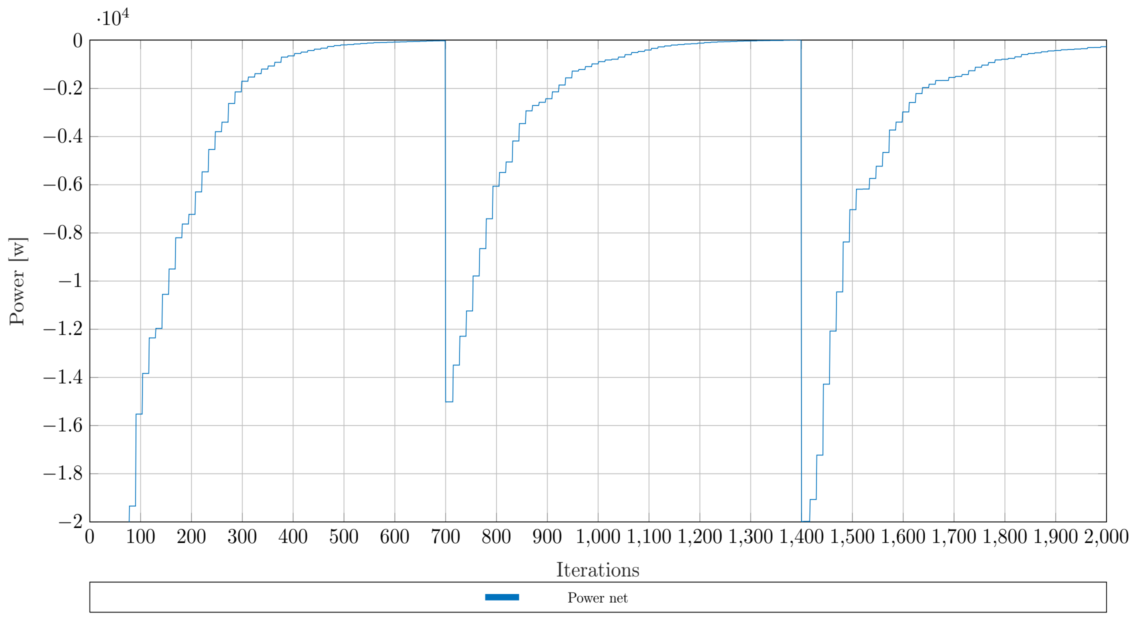

Figure 5-15: Algorithm 3: Power net in Microgrid for simulation in IEEE 30 bus system.

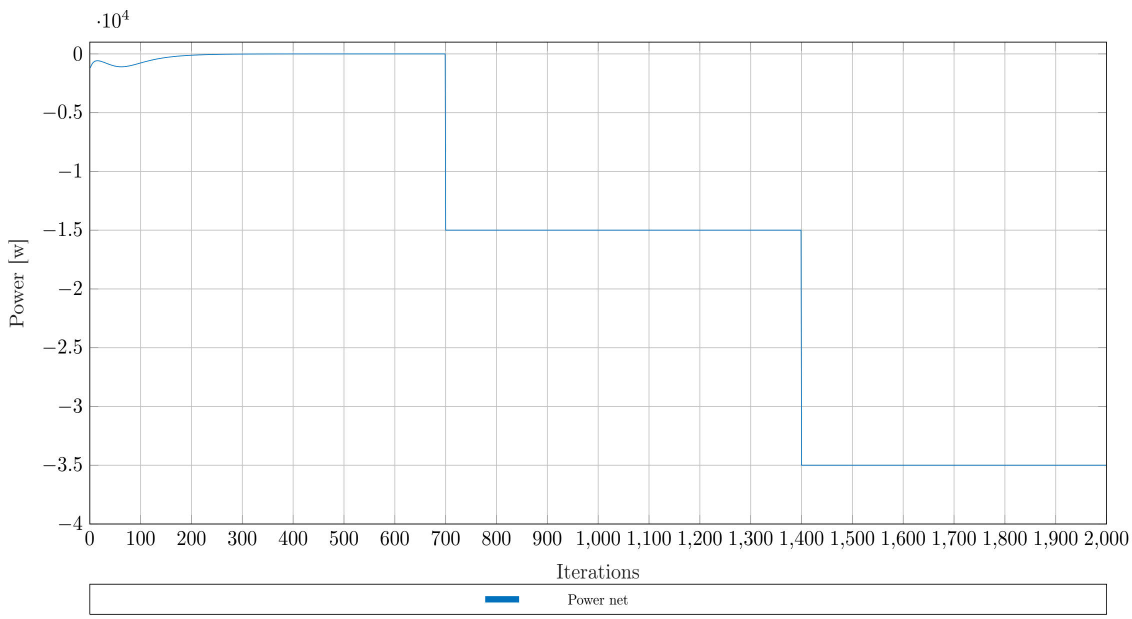

Figure 5-16: Algorithm 4: Power net in Microgrid for simulation in IEEE 30 bus system.

Finally, it is possible to see in Figure **5-15** that the asynchronous algorithm reaches a difference of 0[W] between power generated and power demanded in approximately 600 iterations each time the load is changed. While in Figure **5-16** it is shown how the first 700 iterations the algorithm reaches a difference of 0[W] between power generated and power demanded, but in both load rises the difference increase in the total of the increased load.

6 Conclusions

Based on transactive control framework, new economic strategies in the dispatch of distributed generators and the users in microgrids and distribution systems have been proposed. We have proposed four algorithms in order to achieve this objective. Initially, we propose a centralized population dynamics algorithm that can control a microgrid in order to optimize and guarantee the constraints. We have considered some constraints in the generators and consumers with satisfactory results. Then we have obtained an optimal solution that includes, technical, economical and ensures the maximum social welfare of generators and consumers dynamically.

The centralized algorithm to control microgrids is included inside our second algorithm. This algorithm is based on distributed replicator dynamics protocol, this algorithm solves a social welfare optimization problem between distributed generators, smart loads and microgrids in a transactive control framework. The proposed algorithm considers utility functions of generator and consumers in order to reach the optimum social welfare dynamically while maintaining some system constraints. However, this algorithm cannot solve the power system optimization problem if this problem changes through time. Despite this constraint, game theory rises as a robust framework theory to solve optimization problems applied to power systems.

In our third algorithm, we use the first algorithm in order to control the microgrids inside the distribution system, this algorithm is used as the first control layer of our algorithm. We present a hierarchical asynchronous transactive control strategy to operate a power system with microgrids in order to reach the optimal social welfare with satisfactory results. It is proved its behavior theoretically and in simulations. Moreover, the second control layer can address problems in power systems in an asynchronous way. The simulation results show that the hierarchical transactive control algorithm reaches optimal social welfare while maintaining system constraints in a power network. Our system has advantages over other proposed control systems as it relaxes the synchronism condition in the control systems applied to power systems.

Finally, As a future work, all algorithms need to include other constraints such as maximum generation rates. Furthermore, this algorithm could include a dynamical calculation of losses in the system and an algorithm running in parallel to estimate the energy valuation of each consumer agent. Finally, communication failures in the first control layer algorithm and storages system or wind generators agents could be analyzed and included in this algorithm.

Bibliography

[1] M Annaswamy Anuradha and Amin Massoud. *IEEE Vision for Smart Grid Controls: 2030 and Beyond*. 2013.

[2] Hassan Farhangi. The path of the smart grid. *IEEE Power and Energy Magazine*, 8(1):18–28, 2010.

[3] Michael Stadler, Gonçalo Cardoso, Salman Mashayekh, Thibault Forget, Nicholas DeForest, Ankit Agarwal, and Anna Schönbein. Value streams in microgrids: A literature review. *Applied Energy*, 162:980 – 989, 2016.

[4] David Anderson, Chuanlin Zhao, Carl H. Hauser, Vaithianathan Venkatasubramanian, David E. Bakken, and Anjan Bose. A Virtual Smart Grid. *IEEE Power and Energy Magazine*, 10(1):49–57, 2012.

[5] Siddharth Bhela, Vassilis Kekatos, and Sriharsha Veeramachaneni. Enhancing Observability in Distribution Grids using Smart Meter Data. *IEEE Transactions on Smart Grid*, pages 1–1, 2017.

[6] Koen Kok and Steve Widergren. A Society of Devices: Integrating Intelligent Distributed Resources with Transactive Energy. *IEEE Power and Energy Magazine*, 14(3):34–45, 2016.

[7] Antonis G. Tsikalakis and Nikos D. Hatziargyriou. Centralized control for optimizing microgrids operation. *IEEE Transactions on Energy Conversion*, 23(1):241–248, 2008.

[8] Aj Wood and Bf Wollenberg. *Power Generation Operation and Control*. 2013.

[9] Norhamimi Hamdan, MM Ahmed, and I Hassan. New costs optimization concepts for unit commitment and economic load dispatch in large scale power systems. *Tencon 2004. 2004 Ieee ...*, pages 6–9, 2004.

[10] K.S. Hindi and M.R. Ab. Ghani. Multiperiod secure economic dispatch for large-scale power systems. *IEE Proceedings C Generation, Transmission and Distribution*, 136(3):130, 1989.

[11] M M a Abdelaziz, H E Farag, and E F El-Saadany. Optimum Droop Parameter Settings of Islanded Microgrids With Renewable Energy Resources. *Sustainable Energy, IEEE Transactions on*, 5(2):434–445, 2014.

[12] Hongjing Chen, Jianxue Wang, and Yao Zhang. Economic dispatch of hydro-thermal power system with large-scale wind power penetration. *Asia-Pacific Power and Energy Engineering Conference, APPEEC*, pages 0–3, 2012.

[13] Yin Cheng, Hai Xin Li, Jing Wei Liu, Yan Zhe Xu, Guo Qiang Sun, and Zhi Nong Wei. A Modified on-Line Rolling Generation Dispatch Model for Accommodating Large-Scale Wind Power. *Applied Mechanics and Materials*, 325-326:569–572, 2013.

[14] H. Kanchev, V. Lazarov, and B. Francois. Environmental and economical optimization of microgrid long term operational planning including PV-based active generators. *15th International Power Electronics and Motion Control Conference and Exposition, EPE-PEMC 2012 ECCE Europe*, pages 1–8, 2012.

[15] Carlos A Macana and Hermanshu R Pota. Optimal Energy Management System for Strategic Prosumer Microgrids: An average bidding algorithm for prosumers aggregators. *Asian Control Conference - ASCC 2017*, 2017.

[16] K. Christodoulopoulos, V. Sourlas, I. Mpakolas, and E. Varvarigos. A comparison of centralized and distributed meta-scheduling architectures for computation and communication tasks in Grid networks. *Computer Communications*, 32(7-10):1172–1184, 2009.

[17] Jay Giri, Manu Parashar, Jos Trehern, and Vahid Madani. The situation room: Control center analytics for enhanced situational awareness. *IEEE Power and Energy Magazine*, 10(5):24–39, 2012.

[18] Julian Barreiro-Gomez, German Obando, and Nicanor Quijano. Distributed Population Dynamics: Optimization and Control Applications. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–11, 2016.

[19] Na Li and Jason R. Marden. Designing Games for Distributed Optimization. pages 1–12, 2013.

[20] Jason R. Marden and Jeff S. Shamma. Game Theory and Distributed Control. *Handbook of Game Theory*, 4:861–899, 2015.

[21] Asuman Ozdaglar, Pablo A Parrilo, and Senior Member. Constrained Consensus and Optimization in Multi-Agent Networks. 55(4):922–938, 2010.

[22] Angelia Nedić, Alex Olshevsky, and César A Uribe. Fast convergence rates for distributed non-bayesian learning. *IEEE Transactions on Automatic Control*, 62(11):5538–5553, 2017.

[23] César A Uribe, Soomin Lee, Alexander Gasnikov, and Angelia Nedić. Optimal algorithms for distributed optimization. *arXiv preprint arXiv:1712.00232*, 2017.

[24] M. Reyasudin Basir Khan, Razali Jidin, and Jagadeesh Pasupuleti. Multi-agent based distributed control architecture for microgrid energy management and optimization. *Energy Conversion and Management*, 112:288–307, 2016.

[25] Emmanouil Loukarakis, Chris J. Dent, and Janusz W. Bialek. Decentralized multi-period economic dispatch for real-time flexible demand management. *IEEE Transactions on Power Systems*, 31(1):672–684, 2016.

[26] Seyedali Moayedi and Ali Davoudi. Distributed Tertiary Control of DC Microgrid Clusters. 31(2):1717–1733, 2016.

[27] Eduardo Mojica-Nava, Sergio Rivera, and Nicanor Quijano. Distributed dispatch control in microgrids with network losses. *2016 IEEE Conference on Control Applications (CCA)*, (October):285–290, 2016.

[28] Panagis N. Vovos, Aristides E. Kiprakis, A. Robin Wallace, and Gareth P. Harrison. Centralized and distributed voltage control: Impact on distributed generation penetration. *IEEE Transactions on Power Systems*, 22(1):476–483, 2007.

[29] Nicanor Quijano, Carlos Ocampo-martinez, Julian Barreiro-gomez, German Obando, Andres Pantoja, Eduardo Mojica-nava, and P O C C Ocampo-martinez. The Role of Population Games and Evolutionary Dynamics in Distributed Control Systems. *IEEE Control Systems Magazine*, 2017.

[30] Florian Dörfler, John Simpson-Porco, and Francesco Bullo. Breaking the Hierarchy: Distributed Control & Economic Optimality in Microgrids. *IEEE Transactions on Control of Network Systems*, 5870(c):1–1, 2014.

[31] Xiangjing Su, Mohammad A S Masoum, and Peter J. Wolfs. Multi-Objective Hierarchical Control of Unbalanced Distribution Networks to Accommodate More Renewable Connections in the Smart Grid Era. *IEEE Transactions on Power Systems*, 31(5):3924–3936, 2016.

[32] Wenchuan Wu, Weiye Zheng, Yibing Liu, Zhigang Li, and Boming Zhang. Fully distributed multi-area economic dispatch method for active distribution networks. *IET Generation, Transmission & Distribution*, 9(12):1341–1351, 2015.

[33] E. Mojica-Nava, C. Barreto, and N. Quijano. Population games methods for distributed control of microgrids. *IEEE Transactions on Smart Grid*, 6(6):2586–2595, Nov 2015.

[34] Jesper Knudsen, Jacob Hansen, and Anuradha M Annaswamy. A Dynamic Market Mechanism for the Integration of Renewables and Demand Response. *IEEE Transactions on Control Systems Technology*, 24(3):940–955, 2016.

[35] Junjie Hu, Guangya Yang, Koen Kok, Yusheng Xue, and Henrik W Binder. Transactive control: a framework for operating power systems characterized by high penetration of distributed energy resources. *Journal of Modern Power Systems and Clean Energy*, pages 1–14, 2016.

[36] Farrokh A. Rahimi and Ali Ipakchi. Transactive Energy Techniques: Closing the Gap between Wholesale and Retail Markets. *Electricity Journal*, 25(8):29–35, 2012.

[37] Arman Kiani Bejestani, Anuradha Annaswamy, and Tariq Samad. A hierarchical transactive control architecture for renewables integration in smart grids: Analytical modeling and stability. *IEEE Transactions on Smart Grid*, 5(4):2054–2065, 2014.

[38] Junjie Hu, Guangya Yang, Koen Kok, Yusheng Xue, and Henrik W Binder. Transactive control: a framework for operating power systems characterized by high penetration of distributed energy resources. *Journal of Modern Power Systems and Clean Energy*, pages 1–14, 2016.

[39] Dylan J. Shiltz, Miloš Cvetković, and Anuradha M. Annaswamy. An Integrated Dynamic Market Mechanism for Real-Time Markets and Frequency Regulation. 7(2):875–885, 2016.

[40] F. Bullo. *Lectures on Network Systems*. Version 0.96, 2018. With contributions by J. Cortes, F. Dorfler, and S. Martinez.

[41] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. 2010.

[42] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000.

[43] Dimitri P. Bertsekas, Angelia Nedić, and Asuman Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Belmont, Massachusetts, 2007.

[44] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 25. Cambridge University Press, 2010.

[45] a. Nedic and a. Ozdaglar. Distributed Subgradient Methods for Multi-Agent Optimization. *IEEE Transactions on Automatic Control*, 54(1):1–28, 2009.

[46] A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *Journal of Optimization Theory and Applications*, 142(1):205–228, 2009.

[47] C. D. Meyer. Matrix Analysis and Applied Linear Algebra. 2001.

[48] Eduardo Mojica-Nava, Carlos Andrés Macana, and Nicanor Quijano. Dynamic Population Games for Optimal Dispatch on Hierarchical Microgrid Control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(3):306–317, 2014.

[49] Eduardo Mojica-Nava, Sergio Rivera, and Nicanor Quijano. Game-theoretic dispatch control in microgrids considering network losses and renewable distributed energy resources integration. *IET Generation, Transmission & Distribution*, 11(6):1583–1590, 2017.

[50] William H Sandholm. *Population Games and Evolutionary Dynamics (Economic Learning and Social Evolution)*. 2010.

[51] J Hofbauer and K Sigmund. Evolutionary Games and Population Dynamics, 1998.

[52] Lingfeng Wang Lingfeng Wang and C Singh. Reliability-Constrained Optimum Placement of Reclosers and Distributed Generators in Distribution Networks Using an Ant Colony System Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(6):757–764, 2008.

[53] E. Baron-Prada, E. Osorio, and E. Mojica-Nava. Resilient transactive control in microgrids under dynamic load altering attacks. In *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, pages 1–5, Oct 2017.

[54] M. A. G. de Brito, L. Galotto, L. P. Sampaio, G. d. A. e Melo, and C. A. Canesin. Evaluation of the main mppt techniques for photovoltaic applications. *IEEE Transactions on Industrial Electronics*, 60(3):1156–1167, March 2013.

[55] Guiqiang Li, Yi Jin, M.W. Akram, Xiao Chen, and Jie Ji. Application of bio-inspired algorithms in maximum power point tracking for pv systems under partial shading conditions – a review. *Renewable and Sustainable Energy Reviews*, 81:840 – 873, 2018.

[56] Huanhai Xin, Deqiang Gan, Yun Liu, and Zhen Wang. Control of virtual power plant in microgrids: a coordinated approach based on photovoltaic systems and controllable loads. *IET Generation, Transmission & Distribution*, 9(10):921–928, 2015.

[57] Hongmei Tian, Fernando Mancilla-David, Kevin Ellis, Eduard Muljadi, and Peter Jenkins. A cell-to-module-to-array detailed model for photovoltaic panels. *Solar Energy*, 86(9):2695–2706, 2012.

[58] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3):381–388, 2011.

[59] Omid Palizban, Kimmo Kauhaniemi, and Josep M. Guerrero. Microgrids in active network management - Part I: Hierarchical control, energy storage, virtual power plants, and market participation. *Renewable and Sustainable Energy Reviews*, 36:428–439, 2014.

[60] Peter Asmus. Microgrids, Virtual Power Plants and Our Distributed Energy Future. *Electricity Journal*, 23(10):72–82, 2010.

[61] Ahmad Alzahrani, Mehdi Ferdowsi, Pourya Shamsi, and Cihan H. Dagli. Modeling and Simulation of Microgrid. *Procedia Computer Science*, 114:392–400, 2017.

[62] Dario Arango Angarita and Sergio Rivera. Robust loss coefficients: application to power systems with solar and wind energy. 9:351–383, 09 2018.

[63] Jizhong Zhu. *Optimization of power system operation*, volume 47. John Wiley & Sons, 2015.

[64] Eder Baron-prada, César A. Uribe, and Eduardo Mojica-nava. A Method for Distributed Transactive Control in Power Systems based on the Projected Consensus Algorithm. *7th IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys18)*, 2018.

[65] Eder Baron-prada and Eduardo Mojica-nava. Hierarchical Asynchronous Transactive Control in Power Systems with Microgrids and Virtual Power Plants. *IET Generation, Transmission & Distribution*, pages 1–11, 2019 (submitted).

[66] Eder Baron-prada and Eduardo Mojica-nava. Constrained Transactive Optimization Control in Power Systems based on Population Dynamics. In *2019 IEEE Milan PowerTech, PowerTech 2019*, pages 1–6, 2019 (submitted).

[67] A. Cherukuri and J. Cortés. Distributed coordination of ders with storage for dynamic economic dispatch. *IEEE Transactions on Automatic Control*, 63(3), March 2018.

[68] Ashish Cherukuri and Jorge Cortés. Decentralized Nash equilibrium learning by strategic generators for economic dispatch. *Proceedings of the American Control Conference*, 2016-July:1082–1087, 2016.

[69] Ashish Cherukuri and Jorge Cortés. Initialization-free distributed coordination for economic dispatch under varying loads and generator commitment. *Automatica*, 74:183 – 193, 2016.

[70] Fanghong Guo, Changyun Wen, Jianfeng Mao, Jiawei Chen, and Yong-Duan Song. Hierarchical Decentralized Optimization Architecture for Economic Dispatch: A New Approach for Large-scale Power System. *IEEE Transactions on Industrial Informatics*, 2017.

[71] Shaoshuai Mou and A S Morse. Finite-Time Distributed Averaging. *2014 American Control Conference (ACC)*, pages 5260–5263, 2014.

[72] Marc Bui, Franck Butelle, and Christian Lavault. A distributed algorithm for constructing a minimum diameter spanning tree. *Journal of Parallel and Distributed Computing*, 64(5):571–577, 2004.

[73] Michael Elkin. A faster distributed protocol for constructing a minimum spanning tree. *Journal of Computer and System Sciences*, 72(8):1282–1308, 2006.

[74] Beat Gfeller, Nicola Santoro, and Peter Widmayer. A distributed algorithm for finding all best swap edges of a minimum-diameter spanning tree. *IEEE Transactions on Dependable and Secure Computing*, 8(1):1–12, 2011.

[75] Refael Hassin and Asaf Levin. Minimum restricted diameter spanning trees. *Discrete Applied Mathematics*, 137(3):343 – 357, 2004.

[76] Seth Pettie and Vijaya Ramachandran. An optimal minimum spanning tree algorithm. In *Automata, Languages and Programming*, pages 49–60. Springer Berlin Heidelberg, 2000.

[77] M. Raynal. A short introduction to synchronous communication. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 1136–1143, March 2013.

[78] K. S. Stevens, P. Golani, and P. A. Beerel. Energy and performance models for synchronous and asynchronous communication. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(3):369–382, March 2011.

[79] Matthew T Hale and Angelia Nedic. Asynchronous Multi-Agent Primal-Dual Optimization. *IEEE Transactions on Automatic Control*, 9286(c):1–40, 2017.

[80] Bo Yang and Mikael Johansson. Distributed optimization and games: A tutorial overview. *Lecture Notes in Control and Information Sciences*, 406:109–148, 2010.

[81] Matthew T Hale and Angelia Nedić. Cloud-Based Centralized / Decentralized Multi-Agent Optimization with Communication Delays. In *Proceedings of the IEEE Conference on Decision and Control*, volume 54rd IEEE, pages 700–705, 2015.