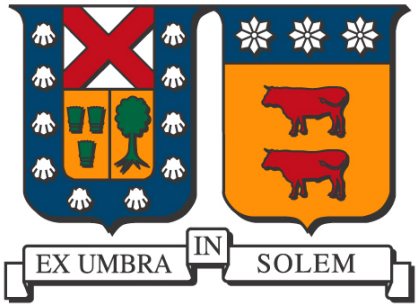


Universidad Técnica Federico Santa María

Department of Electrical Engineering

An Exact Bienstock-Zuckerberg-based algorithm for solving the Convex Hull Pricing problem with ramps constraints

Author:
Lucas Francisco
Bórquez Cerda

Thesis Director:
Dr. Alejandro Alberto
Angulo Cárdenas

A thesis submitted in partial fulfillment of the requirements for the degree of

Magíster en Ciencias de la Ingeniería Eléctrica

May, 2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): x Memoria o trabajo de título x Tesis de Postgrado

Título del trabajo: _An Exact bienstock-zuckerberg based algorithm for solving the convex hull pricing problema with ramps constraints

Nombre del candidato(a): Lucas Francisco Borquez Cerda

Carrera / Grado: Magíster en ciencias de la ingeniería eléctrica

Campus: Casa Central Valparaíso **Departamento:** Departamento de ingeniería eléctrica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Alejandro Alberto Angulo Cardenas, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

X El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: __29-05-2026 **Firma:**

Estudiante o Candidato(a): Lucas Borquez Cerda

Fecha: __29-05-2026 **Firma:** _

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Abstract

This thesis addresses the problem of price distortions and uplift payments in non convex electricity markets by proposing an efficient methodology for Convex Hull Pricing computation. An extended network-flow–based Unit Commitment formulation is developed, explicitly incorporating intertemporal ramping constraints and warm-up periods, and solved through an adaptation of the Bienstock–Zuckerberg algorithm. This column-generation and partition-refinement framework enables an accurate representation of generator operational flexibility and the computation of prices that minimize unrecovered costs, while progressively tightening the feasible solution space across iterations. Computational experiments on real-world systems from California, FERC, RTS-GMLC and Belgium demonstrate that the proposed approach outperforms traditional methods such as Dantzig–Wolfe decomposition and the Level Method in terms of robustness and convergence speed. Moreover, the results show that omitting ramp constraints leads to an underestimation of equilibrium prices, and that preprocessing techniques are critical for the computational viability of the model.

Acknowledgements

En primer lugar, agradecer a mis padres, Carmen y Francisco por todo el apoyo, cariño y confianza que recibí de ustedes en este camino que hicieron posible este logro. A mi hermano Alexander, que siempre me apoyó y siempre confió en mi en este camino.
A mi pareja Fernanda, que siempre estuvo apoyándome, haciendo que mi estadía en la universidad y en la V región fuese muy grata y bien acompañada, también por todos los momentos que compartimos y por los que nos falta compartir, gracias por todo lo que has hecho por mi. Agradecer también a su familia que siempre me tuvieron presente y que en parte, me acompañaron a lo largo de mi etapa universitaria.
A mi tía Nelly y tío Nelson por su apoyo y su compañia también aquí en Viña del Mar, muchas gracias por haberme acompañado y por hacerme parte de su familia. A Fabían, Paulina y Nelson que también me hicieron parte y me acompañaron en esta etapa. También a Reynaldo y Camila que se transformaron en una gran amistad y siempre me impulsaron a ser mejor.
A mis amigos, Cristian, Matías, Fernando, Nicolás, Jorge y Joaquín que también siempre estuvimos apoyándonos para estudiar, realizar trabajos y por todos los momentos que compartimos. También a Miguel, Pablo, Lorena, Ilan, entre tantos otros que tuve la oportunidad de conocer y compartir momentos especiales.
Finalmente, agradezco al profesor Alejandro Angulo por su constante apoyo y orientación durante el desarrollo de este trabajo. Asimismo, agradezco a la Agencia Nacional de Investigación y Desarrollo (ANID), cuyo financiamiento, a través de los proyectos ANID AC3E CIA 250006 y ANILLO ACT250059, contribuyó al desarrollo de esta investigación.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The present chapter gives an overall summary of the whole project. Global and national contexts of the operation of power systems under high renewable generation penetration levels are first presented. Then, the state–of–the–art of the disciplines related to the core of the thesis is shortly reviewed. Finally, the central hypothesis, objectives, used methodologies, and contributions of the present project are described.

1.1 Context and motivation

In the 21st century, environmental concerns have gained significant importance both socially and politically, driving substantial transformations across various sectors, particularly within the energy sector. In this context, power systems—historically characterized by high dependence on highly polluting fossil fuel-based generation technologies (coal, gas, and diesel)—are currently undergoing a transition towards cleaner energy matrices aimed at significantly reducing pollutant gas emissions. This transition has promoted an increasing integration of variable renewable energies (VRE), with solar and wind generation playing particularly prominent roles. Due to considerable reductions in investment costs observed in recent years, these renewable technologies have become more accessible and have emerged as key elements within modern electric systems. Nevertheless, the high penetration of VRE introduces new challenges arising from their intermittent and unpredictable nature, thereby increasing complexity in the planning and efficient operation of power systems.

In the particular case of the Chilean power system, this transition process has been accelerated by the national decarbonization plan, which aims to retire or reconvert coal-based thermoelectric plants, simultaneously driving a considerable increase in VRE participation. As depicted in Fig. 1.1, installed renewable generation capacity—particularly solar and wind—has experienced exponential growth in recent years. Additionally, Fig. 1.2 illustrates the geographical distribution of renewable generation capacity across the different regions of the country (from the northern to the southern region). It clearly demonstrates a predominance of solar plants in the northern region, taking advantage of excellent solar radiation conditions, whereas the southern region primarily features wind generation due to favorable wind conditions.

In this context, characterized by a significant increase in uncertainty associated with power system planning and operation due to the high penetration of variable renewable energies, the need to establish appropriate pricing policies for determining energy prices in the spot market.

In electricity markets, a pricing model based on marginal costs is commonly employed. The marginal cost at a specific point in time refers to the incremental cost incurred to supply an additional unit of energy. This cost can be defined by either the variable cost of the generating unit in audited-cost markets or by the bid price offered by participants in offer-based markets. Generally, this pricing methodology has been effectively applied across various electricity markets globally, including prominent markets such as PJM, MISO, and CAISO [1].

Although marginal pricing theory has been successfully applied in the previously mentioned markets, it only ensures adequate revenue for system units if the marginal cost is determined by the most expensive operating unit. However, in non-convex markets such as current electricity markets, the market-clearing price does not accurately reflect startup and shutdown costs nor other fixed costs associated with generation units. Therefore, whenever a generating unit is started or

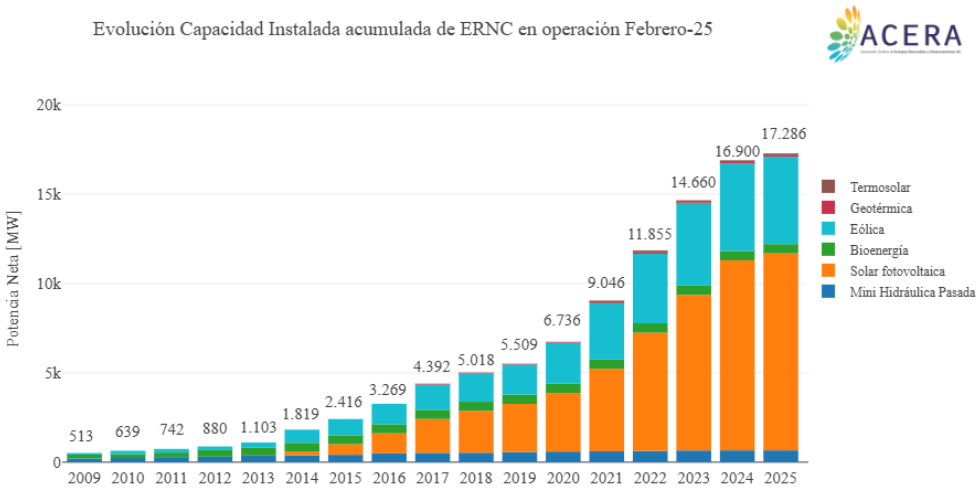


Figure 1.1: Evolution of the installed capacity accumulated of ERNC between 2009 and february 2025. Source: ACERA (`https://acera.cl/estadisticas/`)

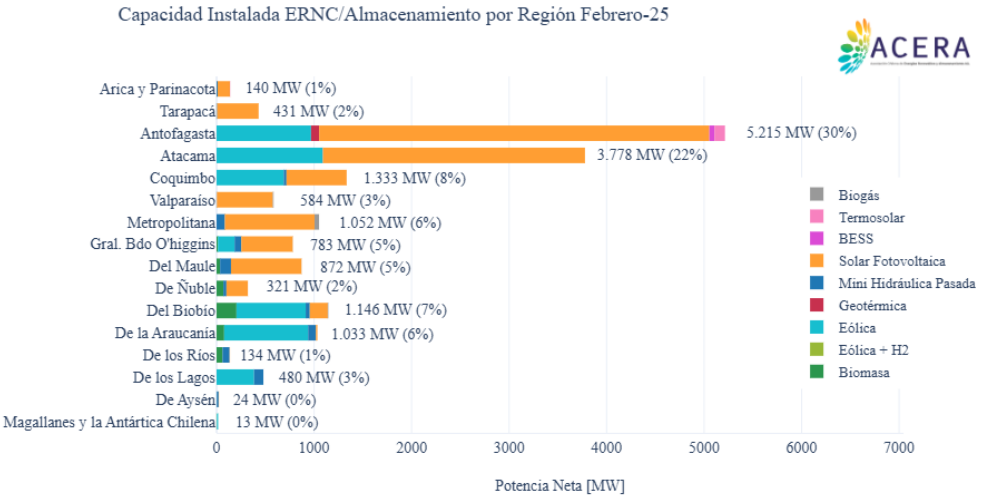


Figure 1.2: Installed capacity of renewable energy in february 2025 by region of Chile Source: ACERA (`https://acera.cl/estadisticas/`)

shut down, additional payments (uplift) must be provided to cover these respective costs. Another aspect not captured by marginal cost pricing is the operational scheduling of inflexible units. These cases include units that cannot be shut down due to prolonged shutdown periods and must operate at their technical minimum levels without impacting the market-clearing price (potentially being units more expensive than the marginal unit). As a result, the revenues these units obtain from the electricity market are insufficient to cover their operational costs, leading to economic losses for these generators. When such situations occur, the system operator must provide uplift to compensate for the economic losses incurred by units operating under these conditions.
On the other hand, in recent years, the participation of Non-Conventional Renewable Energy (NCRE) in the energy matrix has increased considerably. In the Chilean electrical system, as of March 2024, the installed capacity stands at 33.5 GW, with 64.9% from NCRE and another 7.7 GW of renewable projects are under construction [2]. With the growing prominence of renewable energies in electrical systems, it has become increasingly common to observe days in which the marginal cost during periods of high solar generation is equal to zero. Under these operational conditions, certain thermal units must remain online at their technical minimum levels even when the marginal price is zero, primarily due to technical constraints such as minimum shutdown and startup times. This implies operating with marginal costs insufficient to cover even the variable costs of these units. Another common phenomenon is the frequent startup and shutdown of thermal units during hours associated with solar ramps, resulting in additional fixed startup and shutdown costs that are also not captured by marginal pricing. Both scenarios inevitably lead to an increase in uplift payments, which are expected to grow further in the future due to the continued expansion of NCRE. According to [3], these additional payments introduce distortions in the electricity market and reduce transparency in price formation, negatively affecting investment signals and the overall competitiveness of the electricity market.
In light of the aforementioned context, efficient pricing policies have been explored to reduce uplift payments. One proposed alternative to reduce uplift payments is presented in [4], known as the Convex Hull Pricing (CHP) problem, which consists of setting prices from the slope of the convex hull of the cost curve parameterized by demand. This new pricing policy has been widely studied in the state of the art for its ability to reduce uplifts and even eliminate them at certain operation points. By closely aligning market-clearing prices with actual dispatch costs, CHP promotes economic efficiency, transparency, and fairness among market participants. Moreover, the convex hull approach addresses non-convexities arising from unit commitment constraints and generator start-up costs, which traditional marginal-cost pricing methods often fail to capture effectively.

1.2 Summary

This section comprises the principal foundations and methods of the present thesis, from basis hypothesis to main contributions.

1.2.1 Hypothesis of work

As the electricity market is inherently non-convex, determining prices that reduce market distortion presents a significant challenge. This has driven state-of-art research focusing on approaches to solve the Convex Hull Pricing problem, a method aimed at minimizing additional payments and increasing market transparency. In this context, the following working hypothesis is formulated:

> *It is feasible to implement the Bienstock-Zuckerberg algorithm to efficiently solve the Convex Hull Pricing problem, including ramp constraints, formulated as a network flow problem, and achieve results that are similar to and more efficient than state-of-the-art methodologies.*

1.2.2 Problem statement and objectives

In this work, a formulation of the Bienstock–Zuckerberg (BZ) algorithm applied to the Convex Hull Pricing problem is proposed, focusing on a network-flow representation. The CHP problem is first expressed through a network-flow formulation that incorporates operational and startup ramp constraints, which directly influence the feasible solution space. Then, based on the impact of these constraints, the CHP problem is decomposed into a master problem and subproblems (one

per generator) and solved using the BZ algorithm. This algorithm iteratively solves the master problem to update the market-clearing price and the subproblems, whose solutions restrict the feasible operation of the generators in the master problem.
In light of the foregoing, the main objective of this thesis is to implement the BZ algorithm for the convex hull pricing problem represented as a network-flow formulation that incorporates operational ramp constraints. These constraints affect the feasible solution polyhedron of the problem, making it unsuitable for shortest-path algorithms typically used in network problems and instead requiring the resolution of a MILP through MIP solvers. The specific objectives are:

1. In the context of Convex Hull Pricing, review the current literature on: Algorithm for solving the Convex Hull Pricing Problem and Network-Flow-based optimization models and algorithms to solve them.

2. Implement the main algorithms proposed in the literature in small-scale test systems to compare algorithm performances.

3. Design a Network-Flow-based model of the Convex Hull Pricing Problem that includes generator ramp constraints.

4. Design a Bienstock-Zuckerberg-Based Algorithm to Solve the Convex Hull Pricing Problem Formulated as a Network-Flow problem.

5. Perform tests on large-scale systems (such as *FERC system*) to validate the scalability of the proposed algorithm.

6. Analyze and benchmark the quality of the solution by applying the proposed algorithm to large-scale systems.

1.2.3 Methodology

The above presented specific objectives are addressed by applying the following methodology.

1. Formulation of a CHP problem based on a network-flow model that includes operational ramp constraints, where the inclusion of ramping affects the feasible solution polyhedron of the CHP problem.

2. Implementation of a solution approach based on the BZ algorithm for the CHP problem with operational ramp constraints. The method decomposes the model into a restricted master problem and per-generator subproblems. Because these subproblems include ramp constraints, they cannot be solved by shortest-path algorithms; instead, MIP solvers are required due to the structure of their feasible-solution polyhedra.

3. Execution of computational experiments on both small and large-scale systems to demonstrate the scalability of the proposed approach. In addition, algorithms based on the literature are implemented to benchmark the proposed method against state-of-the-art techniques.

1.2.4 Contributions

The contributions of the present thesis are summarized in the following:

1. This work proposes a formulation of the CHP problem through a network-flow model with the direct inclusion of operational ramp constraints.

2. Implementation of a BZ–based approach for the network-flow convex hull pricing model, including operational ramp constraints in the formulation.

3. This thesis presents computational experiments with the proposed approach in both small and large scale systems. In addition, the computational performance of the proposed method is compared with that of relevant algorithms recently introduced in the state of the art. In particular, comparisons are made with the Dantzig–Wolfe (DW) decomposition and the Level Method (LM) algorithms reported in the literature.

1.2.5 Document structure

Initially, Chapter 2 establishes the theoretical foundations, covering the principles of mixed-integer linear optimization and network flow problems , alongside a detailed review of electricity market operations and the Convex Hull Pricing problem aimed at mitigating uplift payments. Subsequently, Chapter 3 contains the core of the proposed methodology, beginning with the formulation of the Unit Commitment (UC) problem as a network-flow model. Then, this formulation is extended to incorporate critical operational constraints, specifically the inclusion of warm-up periods and intertemporal ramping limits, which necessitate a transition to mixed-integer programming. Following this, the specific resolution framework is presented based on the BZ algorithm, detailing the iterative partition refinement strategy used to solve the restricted master problem. Lastly, strategies to enhance computational performance are described, including network preprocessing techniques and an initialization method based on a relaxed CHP model to accelerate algorithm convergence. Chapter 4 presents the computational experiments conducted. It reports the results obtained on representative FERC, Californian, RTS-GMLC and Belgian systems using the proposed algorithm and other state-of-art approaches. Finally, a comparative analysis of these methodologies is carried out. Finally, chapter 5 presents the conclusions of the present thesis, and some discussion about future research directions is developed.

Chapter 2

Background

This chapter establishes the theoretical and mathematical foundations of this research. First, it reviews the core concepts of linear optimization and mixed-integer programming, with emphasis on network-flow formulations and decomposition techniques suitable for large-scale instances. It then contextualizes the technical and economic operation of electricity markets by introducing standard Unit Commitment (UC) formulations and explaining how operational non-convexities give rise to uplift payments. Finally, it presents Convex Hull Pricing (CHP) as a pricing paradigm aimed at mitigating these distortions, discussing both primal and dual viewpoints and summarizing state-of-the-art solution approaches from the literature, including the Level Method and DW decomposition.

2.1 Principles of linear and mixed-integer optimization

2.1.1 Linear programming

Linear programming (LP) is a core area of operations research that focuses on optimizing a linear objective function subject to a set of linear constraints [5]. In general terms, an LP problem can be written as in (2.1)–(2.3):

$$\min_{x} \quad c^T x, \tag{2.1}$$

$$Ax \leq b, \tag{2.2}$$

$$x \geq 0 \tag{2.3}$$

where x is a vector of non-negative real decision variables, c^T is the vector of objective-function coefficients, A is a coefficient matrix and b is a vector of constants defining the linear constraints (2.2). Solving the LP yields an optimal vector x that minimizes (or maximizes, depending on the formulation) the objective function while satisfying all constraints.

2.1.2 Mixed-integer linear programming

In contrast to LP problems, which involve only real-valued variables, mixed-integer linear programming (MILP) extends LP by incorporating discrete decision variables. In particular, an MILP includes (i) continuous variables and (ii) integer variables, which in many applications are binary and restricted to take values in $\{0, 1\}$. This structure enables the explicit representation of on/off decisions, indivisible resources, and logical relationships. A generic MILP formulation is given in (2.4)–(2.7):

$$\min_{x} \quad c^T x + h^T y, \tag{2.4}$$

$$Ax + Gy \leq b, \tag{2.5}$$

$$y \geq 0, \tag{2.6}$$

$$x \in \mathbb{Z}, \tag{2.7}$$

where x represents the integer variables, y the continuous variables, and (2.5) couples both sets of variables. While LP problems can be solved efficiently with polynomial-time methods, MILP problems are NP-hard; consequently, worst-case solution times can grow exponentially with problem size.
It is important to note that classical LP algorithms (such as the simplex method or interior point methods) cannot be directly applied to solve MILPs to optimality due to the integrality constraints [6]. As a result, an exact MILP solution requires systematic exploration of the discrete feasible set.
The standard approach for solving MILP problems involves the use of MIP solvers, which explore the solution space by repeatedly solving LP relaxations and using them to guide branching and bounding decisions. In practice, these solvers typically combine the following techniques:

- Branch-and-bound: This method builds a search tree by partitioning the feasible set through fixings or bounds on the integer variables. At each node, the LP relaxation is solved to obtain a bound; when the relaxation yields an integer-feasible solution, the current best solution is updated. If a bound node does not improve the best available solution, that branch of the search tree is pruned [5].
- Cutting planes: This approach strengthens the LP relaxation by adding valid inequalities to the model that eliminate fractional regions of the LP relaxation solutions while preserving all integer solutions. By tightening the relaxation, cutting planes reduce the number of nodes explored and improve pruning efficiency.
- Branch and cut: This method integrates branch-and-bound with cutting planes dynamically generating cuts at nodes in the search tree to accelerate convergence to an optimal integer solution.

2.1.3 Network flow optimization problem

Many optimization problems in electrical engineering and other disciplines can be formulated as network-flow problems, which benefit from specialized polynomial-time algorithms and strong polyhedral properties [7]. A network is modeled as a directed graph composed of nodes (vertices) and arcs (directed edges) connecting them, where nodes and arcs may have different interpretations (for instance, in a shortest-path problem, nodes represent locations, and arcs represent the paths between them). The flow represents the amount of a commodity traversing each arc, subject to conservation and capacity constraints.
A generic formulation of a network-flow problem on a directed acyclic network is given in (2.8)-(2.11):

$$\min_{x} \quad c^T x, \tag{2.8}$$

$$Ax \geq b, \tag{2.9}$$

$$0 \leq x \leq u, \tag{2.10}$$

$$x \in \mathbb{Z}, \tag{2.11}$$

here, x denotes the flow variable and c^T represents the cost parameter. Equation (2.9) represents flow conservation (divergence constraint) and (2.10) establishes capacity limits through the vector u. The formulation in (2.8)–(2.11) is exact because no additional constraints beyond conservation and arc bounds are imposed, which yields a totally unimodular (TU) constraint matrix A. This property implies that the integrality restriction on the variable x to be relaxed to a continuous domain while preserving the integrality of the optimal solution, so the problem can be solved as an LP problem. Consequently, network-flow models can often be addressed with significantly lower computational effort than general MILPs.
In particular, a matrix is TU when every square submatrix has a determinant in the set $\{-1, 0, 1\}$, which guaranties that all vertices of the associated polyhedron are integral.

2.1.4 Shortest-path problem

Let $\mathcal{A}$ denote the set of arcs and $\mathcal{N}$ the set of nodes. The pair $(i, j) \in \mathcal{N}$ represents the nodes of an arc that departs from node i and arrives at node j. It is also standard for these formulations

to specify an initial node s and a terminal node e, which define the boundary conditions of the problem. The formulation of the shortest-path problem is therefore given by (2.12)–(2.14):

$$\min_{\mathbf{x}} \quad \sum_{(i,j)\in\mathcal{A}} c_{ij}\, x_{ij} \tag{2.12}$$

$$\text{s.t.} \quad \sum_{j:\,(i,j)\in\mathcal{A}} x_{ij} - \sum_{j:\,(j,i)\in\mathcal{A}} x_{ji} = b_i, \quad b_i = \begin{cases} 1, & \text{si } i = s, \\ 0, & \text{si } i \in \mathcal{N} \setminus \{s,t\}, \\ -1, & \text{si } i = t, \end{cases} \qquad \forall i \in \mathcal{N}, \tag{2.13}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{A}, \tag{2.14}$$

where x_{ij} denotes a binary variable that indicates whether the arc (i,j) is used, and c_{ij} is the corresponding arc cost. However, due to the TU property in (2.13), the integrality of x_{ij} is guaranteed at extreme points, so the binary restriction can be relaxed to $0 \leq x_{ij} \leq 1$ without loss of optimality [6]. This allows the problem to be solved efficiently using LP based methods, such as the simplex algorithm [5], or using dedicated shortest-path algorithms (e.g., Dijkstra's method) [8].

2.1.5 Side-constraints

Although network-flow models offer substantial advantages, many real-world applications require additional modeling features that go beyond flow conservation and arc capacities. These additional requirements disrupt the exact structure of the model and are known as side-constraints, i.e., constraints added to the network-flow model beyond the divergence constraints and arc bounds/capacities. Side-constraints arise when arc flows must satisfy global limits (e.g., budgets), arc-category restrictions, or intertemporal relationships (e.g., limited variation between consecutive time periods). A network-flow problem with side constraints can be written as in (2.15)–(2.18):

$$\min_{x} \quad c^T x, \tag{2.15}$$

$$(2.13), \tag{2.16}$$

$$0 \leq x \leq 1, \tag{2.17}$$

$$Px = D. \tag{2.18}$$

In this formulation, (2.18) represents the side-constraint introduced in the network-flow model.

Impact of Side-Constraints on Network Models

When side constraints are incorporated, the constraint matrix acquires additional rows that are no longer pure node-arc incidence relations, and the TU property is generally lost. As a result, integrality of the LP relaxation is not guaranteed, and the problem typically becomes an MILP that must be solved with MIP techniques (MIP solvers).

Since specialized shortest-path algorithms and min-cost flow algorithms can no longer be used to solve the problem, branch-and-bound-based methods or decomposition strategies are commonly required. An complementary approach is to strengthen the relaxation through valid inequalities (e.g., cuts derived from binary flow decisions or flow cover inequalities). These enhance LP relaxation, which becomes weaker when side-constraints are added [9].

In the context of electricity markets, several generator operating characteristics can be embedded in a network-flow formulation without necessarily breaking TU, such as minimum/maximum output limits and minimum up/down times. However, other essential operating characteristics do impact the TU structure and must therefore be treated as side-constraints, most notably intertemporal ramping restrictions.

2.1.6 Large-scale optimization problems

Many large-scale LP and MILP models in power systems and related applications exhibit a block-angular structure, i.e., a collection of mostly independent substructures coupled through a relatively

small set of linking constraints. A generic representation is given by (2.19)-(2.21):

$$\min_{x} \quad c_1^T x_1 + c_2^T + ... + c_n^T x_n, \tag{2.19}$$

$$\textbf{s.t} \quad B_1 x_1 + B_2 x_2 + ... + B_n x_n = d, \tag{2.20}$$

$$\mathrm{A}\boldsymbol{x} = \mathrm{B}. \tag{2.21}$$

In this formulation, (2.20) represents the coupling (linking) constraints between variable blocks, while (2.21) encodes the local constraints, which apply independently to each block. The local constraint matrix A often has a diagonal or block-diagonal form, as illustrated in (2.22).

$$\mathrm{A} = \begin{pmatrix} a_1 & 0 & 0 & 0 \\ 0 & a_2 & 0 & 0 \\ & & \ddots & \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & a_n \end{pmatrix}, \qquad \mathrm{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \tag{2.22}$$

This block-angular structure enables tailored algorithmic strategies that exploit separability across blocks while coordinating feasibility through the coupling constraints. The main approaches relevant to this work are briefly reviewed below.

Lagrangian relaxation

For large-scale linear and mixed-integer problems with a block-angular structure, Lagrangian relaxation dualizes the coupling constraints while retaining the local constraints within each block. Given the formulation in (2.19) to (2.21), the problem resulting from relaxing (2.20) with a fixed Lagrange multiplier π^* is presented in (2.23).

$$\max_{\pi} \quad \left\{ \min_{x} \sum_{i=1}^{n} c_i^T x_i + \pi^T \left(d - \sum_{i=1}^{n} I_i x_i \right) \right\}, \tag{2.23}$$

$$\textbf{s.t} \quad \mathrm{A}_i x_i = B_i \qquad \forall i = 1, 2, ..., n. \tag{2.24}$$

$$\min_{x} \quad \mathcal{L}(x, \pi^*) = \sum_{i=1}^{n} c_i^T x_i + \pi^* \left(d - \sum_{i=1}^{n} I_i x_i \right), \tag{2.25}$$

$$\textbf{s.t} \quad \mathrm{A}\boldsymbol{x} = \mathrm{B}. \tag{2.26}$$

The problem described in (2.23)–(2.26) is formulated as a max–min problem, which is typically solved through an iterative process. First, the inner minimization problem is addressed, which can be decomposed into n fully independent subproblems with shifted costs $c_i^T - I_i \pi^*$. These subproblems can generally be solved using specialized algorithms and, due to their independence, the parallel computing capabilities of MIP solvers can be exploited. After solving all subproblems, the algorithm proceeds with the outer maximization step to find the optimal dual variable π. This outer problem is typically solved via subgradient methods or advanced techniques such as the Bundle Method or stabilized relaxation approaches. This framework allows the derivation of strong dual bounds, and for MILP cases, valid lower bounds [10, 7].

Dantzig-Wolfe decomposition

DW decomposition exploits block-angular structure by separating the original problem into independent subproblems—one per variable block—while aggregating the coupling constraints into a master problem. Each subproblem is solved over its own feasible polyhedron and contributes columns to the master, these columns correspond to extreme points of the individual polyhedra and represent operational plans that the master combines convexly to satisfy the linking constraints. This procedure is typically implemented via column generation: the master solves a restricted problem and provides dual information, which the subproblems use to identify new cost improving columns [11]. The procedure stops when no profitable columns remains. For mixed-integer cases, the framework is extended via branch-and-price.

DW is particularly effective in exploiting separability, reducing the size of the master problem, and enabling the use of parallel computation for subproblems. This often yield strong dual bounds due to tighter polyhedral representation at the block level [12]. However, some drawbacks include potential degeneracy and tailing in the master, which often require stabilization techniques (such as proximal terms or trust regions) to enhance convergence [13]. Additionally, difficult pricing problems or dense coupling constraints can offset computational gains, especially when integrated into a branch-and-price scheme [14, 12].

Interior points method

Interior points methods (IPM) can exploit block-angular structures through Schur complement decompositions and partial factorizations, enabling the solution of large sparse linear systems in a block-wise manner while preserving sparse coupling. This approach improves numerical efficiency through preconditioning and allows high-performance parallel processing [15]. Additionally, combining IPM with column generation has shown significant benefits in accelerating the solution of large-scale dense master problems [16].

Bienstock-Zuckerberg algorithm

The Bienstock-Zuckerberg (BZ) algorithm targets large-scale LP relaxations of block-angular problems by combining Lagrangian duality of the coupling constraints with an adaptive variable aggregation mechanism [17]. After dualizing the coupling constraints, the problem decomposos into subproblems by block, and from their solutions, partitions H that group variables according to their optimal values. Then an LP problem is formulated, which includes all original constraints but restricts the feasible region based on these partitions. The partitions are iteratively refined until the relaxation optimum is recovered using onlya compact set of aggregated variables, enabling the treatment of instances with very large numbers of variables and constraints while maintaining a tractable master problem [17].
In [18], the BZ procedure is reinterpreted as a column generation variant that takes advantage of the combinatorial structure of the problem to accelerate the closure of the gap. Compared to traditional methods, two main features stand out: (i) the generated columns correspond to mutually orthogonal directions, so adding them induces a controlled contraction/aggregation of variables, and (ii) this contraction is conservative, ensuring that no previously discovered optimal solutions are excluded.

2.2 Electricity markets

An electrical system consists of an interconnected set of infrastructure and equipment designed to produce, transport, distribute, and deliver electrical energy from generation centers (producers) to load centers (consumers) as shown in Figure 2.1. This process begins at generation plants, which can utilize various types of technology (hydroelectric, wind, solar, thermal, among others) to produce electrical energy. Subsequently, this energy is transferred to step-up substations, whose function is to raise the voltage to hundreds of kilovolts (typically ranging from 110 kV to 500 kV), reducing energy losses during long-distance transmission. Through high-voltage transmission lines, typically supported by metal towers, electrical energy is transported to consumption areas, which can correspond to residential or industrial demand At this stage, step-down substations lower the voltage to intermediate levels (typically between 10 kV and 38 kV) for local distribution. Finally, in distribution networks, the voltage is reduced again to safe levels (220 V) to satisfy residential, commercial, and industrial demand.
Although the previous description provides a general overview of the technical functioning of a power system, in practice there are several additional stakeholders who play key roles in its operation and development. Among them are investors, whose participation is essential for the implementation of generation, transmission, and distribution projects. Their participation enables the expansion and modernization of the electrical infrastructure, contributing to the development of safer, more efficient, and more sustainable systems.
Other relevant stakeholders responsible for the regulation and operation of the power system are

- Transmission System Operator (TSO): The TSO is responsible for ensuring the safe, reliable, and efficient operation of the transmission network, including the management of congestion.

To achieve this, the TSO continuously monitors and controls power flows, enabling real-time monitoring of the operating conditions of the system. In addition, the TSO plays an active role in transmission network expansion planning by anticipating future infrastructure needs and conducting N-1 contingency studies. These studies ensure that the system can withstand the outage/failure of any single component, thereby improving operational resilience.

- Distribution System Operator (DSO): The DSO is responsible for the safe and efficient management of the electric distribution network, which enables the delivery of electricity from substations to end users. Its main objective is to ensure the continuity and quality of the electricity supply from substations to end users. The DSO typically oversees the connection process for new consumers and, depending on the regulatory framework of each country, is also responsible for the technical integration of Distributed Energy Resources (DER), such as residential photovoltaic generation (PV), energy storage systems (ESS) and electric vehicles.

- Market Operator (MO): The MO is the entity responsible for managing and regulating the operation of the wholesale market with the objective of ensuring transparency, competition, and efficiency in market transactions. Among its main functions are receiving energy supply offers from generators and collecting demand information from consumers or retailers, in order to determine electricity prices based on the balance between supply and demand. The MO also manages financial settlement, coordinating payments and charges among market participants. The responsibilities of the MO may extend beyond the scope of the wholesale energy market, also covering the coordination of complementary markets, such as ancillary services, capacity markets, the allocation of operating reserves, and regulatory mechanisms, among other functions necessary to ensure the safe and efficient operation of the power system as a whole.

In some electricity markets, such as Chile, the functions traditionally assigned to the TSO and the MO are integrated into a single entity known as the Independent System Operator (ISO). In the Chilean power system, this role is carried out by the Coordinador Electrico Nacional (CEN), an autonomous and technical organization responsible for coordinating the operation of the national power system.
The main objective of the CEN is to ensure the real time balance between electricity generation and demand, with the aim of achieving this operation at the lowest possible cost while maintaining safe conditions for the electricity system. To accomplish this, it solves the economic dispatch (ED) problem of generation units based on their variable costs, verifies the technical feasibility of operational schedules, and coordinates power flows across the transmission network to prevent congestion. In addition, the Coordinator plays a key role in the medium and long term planning of the power system, contributing to the definition of network expansion and the assessment of future system performance under different operating scenarios.
To perform the ED problem, the ISO considers the cost associated with each generating unit that participates in the market. This cost may correspond to the actual variable cost of production (in the case of markets based on audited costs) or to the bid prices submitted by generating units (in markets organized under a competitive bidding scheme). Once the ED that meets demand and respects the system operational constraints is determined, the electricity price is established. This price corresponds to the value at which the dispatched generators will be compensated for supplying demand during that period.
As previously noted, the ISO (or alternatively the MO) is responsible for carrying out the financial settlement processes of the electricity market, managing payments and collections among the various participants. These agents may participate in multiple electricity markets that operate in parallel and serve complementary purposes, each with its own objectives and structures.
One of the main markets is the wholesale market, where energy is traded between generators and large consumers, and where the electricity price is determined based on the daily economic dispatch. Complementing this is the contract market, in which agents negotiate bilateral agreements or long-term contracts for the purchase and sale of electricity at pre-established prices. These types of contract help mitigate the exposure to the volatility of the spot price in the wholesale market, providing revenue certainty for generators and cost predictability for consumers.
Additionally, in many power systems, including the Chilean system, complementary mechanisms such as ancillary services markets coexist. These mechanisms allow for the allocation and compensation of essential services required for the proper operation of the power system, such as frequency

regulation, voltage control, and operating reserves, thus contributing to the security and quality of the electricity supply.

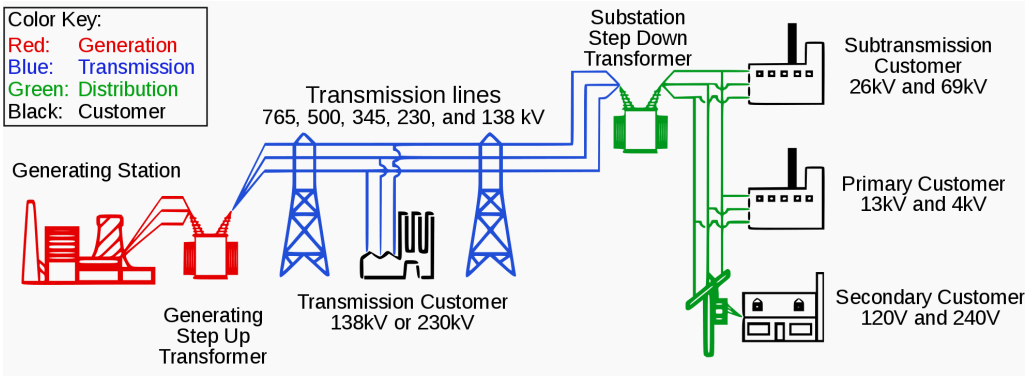


Figure 2.1: Power system basic schema. Different market agents are connected at different levels of the power system. Source: https://electrical-engineering-portal.com/electric-power-systems

2.3 Deterministic models of the unit commitment

The Unit Commitment problem is a mixed-integer optimization problem that determines the on/off status of generating units and their dispatched power for each time period t over a planning horizon. Its objective is to minimize total operating costs while meeting demand reliably and securely [19]. UC coordinates the operation of heterogeneous resources, including hydro and thermal generation, NCRE sources, and more recently, energy storage systems (ESS).
Accurate UC formulations must represent the main technical limits of each technology. For thermal units, these typically include minimum and maximum output limits, minimum up/down time constraints, ramp-rate limits, and (when applicable) nonlinear production costs. For ESS, the formulation introduces variables and constraints for the state of charge, charging/discharging power limits, conversion efficiencies, and related operating conditions. Incorporating these characteristics is essential to obtain schedules that are both feasible and operationally meaningful.

2.3.1 Short–term planning problems

The MILP formulation of the deterministic UC problem is a widely used optimization method in power systems. This formulation allows determining the operational status (on/off) and the generation of each unit, with the goal of minimizing the operational cost of the total system while the power demand meets the planning horizon and satisfying technical constraints of the generators [19], [20]. The MILP formulation includes binary variables for operational decisions (online status, startup decisions, and shutdown decisions) which are constrained by minimum up/down time requirements and continuous variables for power dispatch subject to generators technical constraints such as generation limits, ramping constraints, and non-linear cost approximated through piecewise linear [21], [22].
One of the main advantages of MILP formulation is the availability of powerful optimization solvers and exact algorithms that guaranty globally optimal solutions (as long as the model is properly formulated), using exact methods such as branch-and-bound and branch-and-cut [9],[6]. These methods make effective use of the structural features of MILP models to enhance computational efficiency and enable scalability in large-scale problem instances. Gurobi, CPLEX, HiGHS, and Xpress (among others) are among the most commonly used solvers in both the academic and industrial sectors. These solvers incorporate a variety of advanced features, such as automatic generation of valid cutting planes, preprocessing routines, parallel computing capabilities, and sophisticated search strategies for navigating the branch-and-bound tree [23], [24].
The growing academic interest in MILP formulations for the UC problem is closely linked to the significant improvements in solver efficiency observed in recent decades. These improvements are largely attributed to improved memory management, more effective parallelization, and optimized

exploration of high-dimensional discrete search spaces [25]. In addition, UC formulations themselves have evolved into more compact and efficient models, and the development of advanced mathematical decomposition techniques, such as Benders and Lagrangian decomposition among others, has further facilitated the solution of highly detailed real world systems. The most studied formulations are the 1-bin and 3-bin models, which are described in detail below.

2.3.2 UC 1-bin model

The 1-bin formulation of the UC problem is a MILP model that uses only one binary variable per generating unit and time period. This binary variable captures the on/off status of the unit and is used to model its startup and shutdown transitions. The model formulation follows the approaches proposed in [21] and [25]:

Sets

$g \in \mathcal{G}$: Generators set.
$t \in \mathcal{T}$: Time horizon.

Variables

$p_{g,t} \in \mathbb{R}_{\geq 0}$: Generation output of unit g in time t.
$u_{g,t} \in \{0, 1\}$: Binary variable equal to 1 when unit i is online at time t, and 0 when offline.

Parameters

c_g: Variable cost of generator g.
c_g^U / c_g^D: Start up/shutdown cost of generator g. D_t: load at time t.
$P_g^{\min} / P_g^{\max}$: Minimum/maximum power output of generator g.
RU_g / RD_g: Operational ramp-up/ramp-down limit of generator g.
SU_g / SD_g: Startup and shutdown ramping capability of generator g.
UT_g / DT_g: Minimum up/down time of generator g.

Objective function

$$\min \sum_{t \in \mathcal{T}} \sum_{g \in \mathcal{G}} \left(c_g^g p_{g,t} + CU_{g,t} + CD_{g.t} \right) \tag{2.27}$$

Constraints

1. Start up and shutdown cost:

$$CU_{g,t} \geq c_g^U (v_{g,t} - v_{g,t-1}) \tag{2.28}$$

$$CU_{g,t} \geq 0 \tag{2.29}$$

$$CD_{g.t} \geq c_g^D (v_{g,t-1} - v_{g,t}) \tag{2.30}$$

$$CD_{g,t} \geq 0 \tag{2.31}$$

2. Power balance:

$$\sum_{g \in \mathcal{G}} p_{g,t} = D_t \quad \forall t \in \mathcal{T} \tag{2.32}$$

3. Generation limits:

$$P_g^{\min} \cdot u_{g,t} \leq p_{g,t} \leq P_g^{\max} \cdot u_{g,t} \quad \forall g, t \tag{2.33}$$

4. Operational ramps and start up and shutdown ramps:

$$p_{g,t} - p_{g,t-1} \leq RU_g \cdot u_{g,t-1} + SU_g \cdot (u_{g,t} - u_{g,t-1}) \quad \forall g,\ t > 1 \tag{2.34}$$

$$p_{g,t-1} - p_{g,t} \leq RD_g \cdot u_{g,t} + SD_g \cdot (u_{g,t-1} - u_{g,t}) \quad \forall g,\ t > 1 \tag{2.35}$$

5. Minimum up time:

$$\sum_{\tau=t}^{t+UT_g-1} u_{g,\tau} \geq UT_g \cdot (u_{g,t} - u_{g,t-1}) \quad \forall g,\ t \tag{2.36}$$

6. Minimum down time:

$$\sum_{\tau=t}^{t+DT_g-1} (1 - u_{g,\tau}) \geq DT_g \cdot (u_{g,t-1} - u_{g,t}) \quad \forall g,\ t \tag{2.37}$$

The UC 1-bin model uses a single binary variable and therefore includes $N \times T$ binary variables for a system with N generators and T time horizon.
Regarding technical constraints, the 1-bin formulation does not exhibit fundamental drawbacks in terms of modeling capabilities. Although the presence of a single binary variable may complicate the implementation of certain technical constraints, this does not constitute a structural limitation of the formulation.

2.3.3 UC 3-bin model

In contrast to the 1-bin formulation, the 3-bin UC model introduces three binary variables per generator. These binary variables represent the on/off status of the generator, start-up decisions, and shutdown decisions. In general, the 3-bin model is currently the most widely used formulation in the literature due to its computational advantages over the 1-bin model [22], [26]. A general formulation of the UC problem based on the 3-bin model is presented in (2.38)-(2.45).

Sets

$g \in \mathcal{G}$: Generators set.
$t \in \mathcal{T}$: Time horizon.

Variables

$p_{g,t} \in \mathbb{R}_{\geq 0}$: power output of unit g in time t.
$u_{g,t} \in \{0, 1\}$: 1 if unit g is on in time t, 0 otherwise.
$v_{g.t} \in \{0, 1\}$: 1 if the unit g starts up at time t, 0 otherwhise.
$w_{g,t} \in \{0, 1\}$: 1 if the unit shut down at time t, 0 otherwhise.

Parameters

C_g^P: Production cost of generator g.
C_g^{NL}: No-Load cost of generator g.
C_g^{SU}/C_g^{SD}: Start up/sht down cost of generator g.

Objective function:

$$\min \quad \sum_{t\in\mathcal{T}} \sum_{g\in\mathcal{G}} \left(C_g^P p_{gt} + C_g^{NL} u_{gt} + C_g^{SU} v_{gt} + C_g^{SD} w_{gt}\right) \tag{2.38}$$

Constraints

1. Power balance:

$$\sum_{g\in\mathcal{G}} p_{gt} = D_t \quad \forall t \in \mathcal{T} \tag{2.39}$$

2. Power limits:

$$P_g^{\min} u_{gt} \leq p_{gt} \leq P_g^{\max} \cdot u_{gt} \forall g \in \mathcal{G}, t \in \tag{2.40}$$

3. Logical constraints:

$$u_{gt} - u_{g,t-1} = v_{gt} - w_{gt} \quad \forall g \in \mathcal{G}, t \in \mathcal{T} \setminus \{1\} \tag{2.41}$$

4. Ramps constraints:

$$p_{gt} - p_{g,t-1} \leq RU_g u_{g,t-1} + P_g^{\min} v_{gt} \quad \forall g, t \in \mathcal{T} \setminus \{1\} \tag{2.42}$$

$$p_{g,t-1} - p_{gt} \leq RD_g u_{gt} + P_g^{\min} w_{gt} \quad \forall g, t \in \mathcal{T} \setminus \{1\} \tag{2.43}$$

5. Minimum up time:

$$\sum_{\tau=t}^{t+UT_g-1} u_{g\tau} \geq UT_g v_{gt} \quad \forall g, t \in \mathcal{T} : t + UT_g - 1 \leq |\mathcal{T}| \tag{2.44}$$

6. Minimum down time:

$$\sum_{\tau=t}^{t+DT_g-1} (1 - u_{g\tau}) \geq DT_g w_{gt} \quad \forall g, t \in \tag{2.45}$$

To model start-up and shutdown transitions, the 1-bin formulation requires additional constraints that capture state changes between consecutive time periods [21]. One of the main advantages of this formulation lies in the reduction of the number of binary variables, which may simplify the model structure and, in some cases, reduce solution times. However, this structural simplification can compromise the strength of linear relaxation, yielding a larger feasible region and, consequently, a lower computational efficiency in large-scale or technically demanding systems [26].
In contrast, the 3-bin formulation introduces three binary variables for each generating unit: one representing the commitment status, one for start-up decisions, and one for shutdown decisions. Although this increases the dimensionality of the binary variable space, it provides a more explicit and structured representation of state transitions, thereby simplifying the modeling of minimum up/down time constraints and fixed cost components. Studies such as [22] and [26] have shown that the 3-bin formulation yields a tighter and more accurate linear relaxation, which reduces the search space and consequently, improves computational performance and solution quality.

2.3.4 Economic dispatch problem

The ED problem is used mainly for two reasons: (i) to compute the dispatch levels of generation units aiming at the same objectives as the UC problem, and (ii) to determine the electricity price (marginal cost) from the dual variable of the power balance constraint [27]. This is achievable because, unlike the UC formulation, the ED model does not include binary variables, given that the unit on/off status is assumed to be known in advance. A simplified version of the ED problem is outlined below:

$$\textbf{ED:} \quad \min_{p} \sum_{g \in \mathcal{G}} \sum_{t \in T} \left(C_g^P p_{g,t}\right), \tag{2.46}$$

$$\text{s.t.} \quad \sum_{g \in \mathcal{G}} p_{gt} = D_t \qquad : (\lambda_t) \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{2.47}$$

$$p_{g,t} \in \mathcal{X}_g \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{2.48}$$

where $\mathcal{X}g$ denotes the feasible operating region of generator g. The dual variable λt, associated with the power balance constraint (2.47), which can be interpreted as the cost incurred to supply one additional unit of demand. When transmission constraints are considered, this value is known as the Locational Marginal Price (LMP). Figure 2.2 illustrates the scheme for obtaining the LMP.

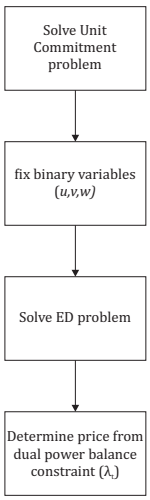


Figure 2.2: General scheme for determining marginal cost.

This pricing methodology has been widely adopted by numerous ISOs around the world [1]. However, because of the inherent non-convexity of electricity markets, it does not capture fixed generation costs such as start-up costs or costs associated with operating at minimum output levels in the resulting market price [28]. A detailed description of the characteristics of non-convex markets (as the electricity market) and their differences from traditional supply–demand markets.

2.4 Non-convex markets

In traditional general equilibrium theory, convexity in the behavior of economic agents and the possibility of continuous allocation proportions are fundamental conditions to ensure the existence of competitive equilibrium prices that equal supply and demand [29]. Under convex assumptions, there exists a set of linear prices (per unit) that are common to all agents, such that no one can unilaterally deviate and improve their situation, thus satisfying Walrasian equilibrium, i.e., Pareto optimality without the need for subsidies [29]. Nevertheless, many real-world markets feature non-convexities, leading to situations in which a unique equilibrium price that satisfies all participants may not exist.

2.4.1 Non-convex properties of electricity markets

There are multiple features of power system operation that introduce non-convexities into electricity markets. Generators, for instance, are subject to fixed start-up and no-load costs, as well as minimum generation thresholds and minimum up and down time constraints, among other discrete operational limits [30]. This means that they cannot adjust their output in a fully continuous manner without incurring additional costs (for example, the cost associated with starting up an additional turbine). On the demand side, large consumers or demand aggregators may require non-divisible energy blocks, for example, a fixed load over a number of hours to run an industrial process that cannot be partially executed. These non-convexities in both supply and demand imply that determining the optimal generation and consumption dispatch constitutes a non-convex optimization problem. This inherent non-convexity in power system operation lies at the core of many challenges in electricity market design.

2.4.2 Market-clearing price in electricity markets

In convex and competitive markets, the equilibrium price typically equals the marginal production cost of the last unit supplied and ensures that the quantity offered and demanded is optimal for all agents. Conversely, in a non-convex market, a single linear price may not yield such an equilibrium across the market, since some operational decisions are discrete (e.g., turning a generating plant fully on or off). This means that it may be impossible to establish a single energy price (typically in MWh) that ensures both that each generator and consumer is optimizing its position and that system balances are satisfied, due to the discrete aspects of some decisions.

2.5 Additiontal payments

Given the non-convex characteristics of electricity markets, a global equilibrium price that meets the conditions of all market participants may not exist. Consequently, at a market-clearing price, inconsistencies can arise between the quantity that the generators are willing to supply and the system total demand. In conventional markets, this may disincentivize participation by generators, who aim to maximize their profits. To address this shortcoming, non-convex markets typically introduce uplift payments to ensure that all dispatched units recover their costs. These payments are traditionally categorized into two types: (i) Make-Whole Payments (MWP), and (ii) Lost Opportunity Cost (LOC), which are explained below.

- **Make-Whole Payments**: MWP are additional payments made by the ISO to generation units to ensure that, once dispatched for economic or reliability reasons, they recover their full operational costs when the market price is insufficient to cover them. These costs, depending on the market design, may include start-up costs, no-load costs, and variable generation costs. Since it constitutes a compensatory payment, it is assessed, at a given dispatch level, as the difference between the generator's production costs and the revenues accrued whenever the latter do not result in a profitable outcome, according to (2.49).

$$\text{MWP}_g = f(p_g^*, u_{g,t}^*) - \phi_g^* \tag{2.49}$$

- **Lost Opportunity Cost**: LOC compensates generators that by, following the ISOs instructions to decrease or adjust their generation due to transmission constraints or other technical reasons, are unable to operate as they would under market conditions, thus losing

potential revenue [31]. For example, a generator is instructed to reduce its output for a given number of hours, and thereby loses the opportunity to sell additional energy during that period. Therefore, ϕ_g is defined as the maximum profit a generator could obtain if it were to self-dispatch rather than follow the dispatch instructions issued by the ISO.

$$\phi_g = \max_{p,u} \sum_{t \in T} \lambda_t^* p_{g,t} - f(p_{g,t}, u_{g,t}), \tag{2.50}$$

$$\textbf{s.t.} \quad (\boldsymbol{p_{g,t}}, \boldsymbol{u_{g,t}}) \in \mathcal{X}_g \qquad \forall t \in \mathcal{T}, \tag{2.51}$$

where (2.50) represents the objective function that maximizes the generators profit given a price λ_t^*, while (2.51) denotes the generators feasible operating region determined by its technical limitations.

However, in electricity markets, generators are required to follow the ISO dispatch instructions, even if these lead to financial losses. This creates an incentive misalignment that may compromise market efficiency and long-term viability. Therefore, the generator's actual revenue (ϕ_g^*) is defined as:

$$\phi_g^* = \sum_{t \in T} \left(\lambda_t^* \boldsymbol{p_{g,t}^*} - f(\boldsymbol{p_{g,t}^*}, \boldsymbol{u_{g,t}^*}) \right). \tag{2.52}$$

The operational variables of the generator correspond to the dispatch schedule determined by the ISO. The difference between the maximum profit the generator could achieve by ideal self-dispatch and the revenues obtained under ISO dispatch, $\phi_g - \phi_g^*$, defines the LOC uplift payments.

The presence of uplift payments is strongly associated with the fact that, under non-convex market conditions, the optimal dispatch determined by the ISO does not necessarily match the individual optimal operating point of each generating unit at the prevailing market price. Although these payments help recover costs not reflected in market-clearing prices, uplift payments introduce distortions that can undermine the transparency of electricity markets. The non-uniform nature of these payments—where generators receive unequal amounts can result in inconsistent price signals. According to [32, 33], this can encourage agents to engage in strategic behavior, especially in the disclosure of marginal costs or volumes offered, to improve their compensation.

2.6 Convex hull pricing problem

In response to the challenge of uplift payments that distort the energy market, the convex hull pricing problem has been formulated to establish an energy price inclusive of fixed generation costs, thereby lowering the system overall uplift payments. As a result, the market achieves enhanced transparency and better price signals.

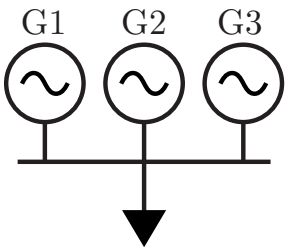


Figure 2.3: Test system.

The CHP problem addresses the market distortion introduced by uplift payments by convexifying the UC problem, such that the resulting prices correspond to the convex envelope of the cost function parameterized by demand. This involves reformulating the UC model such that the discrete decisions of the original problem (start-up/shut-down) and fixed costs are incorporated into an LP problem. As a result, the resulting prices internalize these fixed-cost components, thereby reducing total uplift payments. Consequently, convex hull prices reflect these non-convex elements, minimizing the duality gap between the relaxed LP and the full UC problem, and thereby reducing uplift payments [34, 35].

To illustrate the fundamental principles underlying the convex hull pricing methodology, consider a simplified single-bus system comprising three generating units and a fixed demand level, as shown in Figure 2.3. The technical and cost parameters of each generator are summarized in Table 2.1, and the corresponding optimization problem formulated for a specific time interval is presented in (2.53).

Table 2.1: Generator data in test problem

Gen	Pmax	Pmin	RU/RD	Var. Cost	Fix Cost	Start-up Cost
1	30	10	5	10	3	100
2	30	10	5	20	3	60
3	30	10	5	30	3	40

$$\begin{aligned} \min_{\boldsymbol{p},\boldsymbol{u}} \quad & 10p_1 + 20p_2 + 30p_3 + 103u_1 + 63u_2 + 43u_2 \\ \textbf{s.t} \quad & p_1 + p_2 + p_3 = D_t, \\ & P_{min}u_g \leq p_g \leq P_{max}u_g \qquad g = \{1,2,3\}, \\ & u_g = \{0,1\} \qquad\qquad\qquad g = \{1,2,3\}, \end{aligned} \tag{2.53}$$

The optimization problem defined in (2.53) is first solved by performing a demand sweep over the feasible operating range of the system, initially without incorporating ramp constraints into the formulation. This sweep allows the construction of the system cost function as a function of demand, from which the convex hull and the associated CHP prices can be derived. In Figure 2.4, the objective function cost of the UC problem is shown as a function of demand, along with the associated CHP objective function. The non-convexity of the UC is observed in the cost jump when demand exceeds 30 `MW` and 60 `MW`. In contrast, the function associated with CHP is convex and does not exhibit non-linearities, representing an approximation of the convex hull of the UC problem cost function.

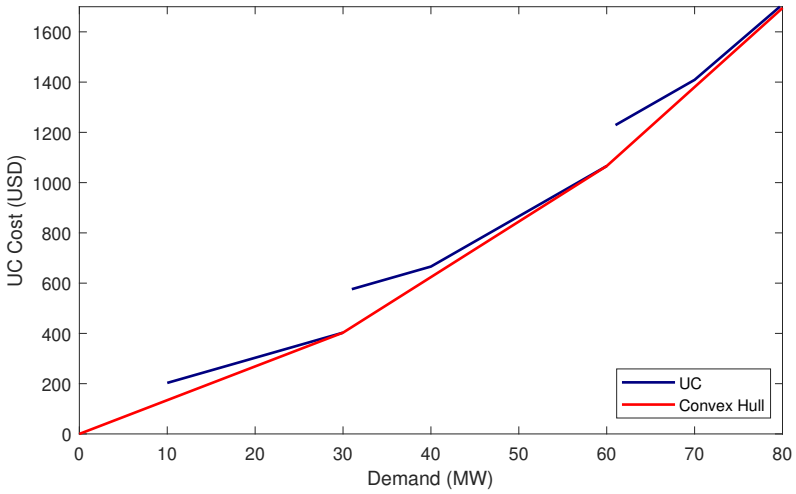


Figure 2.4: Cost function parameterized with demand for UC and CHP

When addressing the CHP problem in large-scale power systems, where the associated computational burden becomes significantly more demanding, two main solution frameworks are generally distinguished in the literature. The first corresponds to a Lagrangian dual-based approach. The second framework corresponds to a primal approach, which reformulates the CHP problem as a linear relaxation of the UC problem.

2.6.1 Lagrangian dual-based approach

The Lagrangian dual approach consists of solving the UC problem by relaxing the power balance constraint through the associated Lagrange multiplier. The vector of optimal multipliers obtained from this dual formulation lies within the subgradient of the convex hull of the cost function, which is parameterized with respect to demand. This implies that the gap between the optimal value of the ED problem and the value of the dual problem exactly represents the total uplift payments required to ensure full cost recovery for the generating units. Therefore, maximizing the dual

objective directly corresponds to minimizing total uplift and leads to convex hull prices that best internalize the non-convex characteristics of the dispatch model [34], [36].

$$\mathcal{L}(\lambda, p, u) = \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} f(p_{g,t}, u_{g,t}) - \sum_{t\in\mathcal{T}} \lambda_t \left(d_t - \sum_{g\in\mathcal{G}} p_{g,t} \right). \tag{2.54}$$

The dual formulation of the Lagrangian dual is given by (2.55)-(2.56):

$$\max_{\lambda} \quad \left\{ \min_{p,u} \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} f(p_{g,t}, u_{g,t}) - \sum_{t\in\mathcal{T}} \lambda_t (d_t - \sum_{g\in\mathcal{G}} p_{g,t}) \right\}, \tag{2.55}$$

$$\textbf{s.t.} \quad (p_{g,t}, u_{g,t}) \in \mathcal{X}_g \qquad \forall t \in \mathcal{T}. \tag{2.56}$$

The Lagrangian dual formulation of the Unit Commitment problem yields a non-smooth convex problem, characterized by a separable structure that allows decomposition by generating unit. This separability property enables the problem to be solved iteratively using subgradient methods, where, at each iteration and for a given price vector λ_t, each unit independently solves a profit maximization problem that accounts for its cost structure and technical constraints. This procedure effectively models decentralized economic decision making under marginal pricing schemes [36].
Despite the aforementioned advantages, although the separable structure of the Lagrangian dual problem allows its decomposition into individual MILP subproblem for each generating unit, this leads to a significant computational burden when scaling to real systems with a large number of units. This difficulty can be mitigated if constraints (2.28)-(2.37) (in UC-1bin model) or (2.40)-(2.45) (in UC-3bin model) precisely describe the convex hull of each units feasible region, thereby eliminating the need to directly solve the integer problems. However, even in such cases, the standard subgradient method tends to suffer from oscillations and lack of direction in convergence due to the shape of the dual Lagrangian function, as discussed in [37].
Several variants of the method have been proposed to improve computational efficiency. In [37], a hybrid approach is introduced, combining subgradient iterations with the progressive elimination of dominated solutions via cutting planes, effectively filtering out suboptimal multipliers and accelerating convergence. However, this approach exhibits limited robustness under certain problem topologies, as discussed in [38]. As an alternative, [39] proposes a method based on the iterative solution of a convex quadratic problem to determine more stable improvement directions. Although this technique ensures finite convergence, it requires maintaining a record of all extreme points visited, resulting in exponential complexity growth.
More recently, methodologies such as LM [40] have been proposed, which employ polyhedral subgradient approximations in combination with explicit stabilization, as well as cutting-plane stabilization techniques. These approaches have shown improvements in numerical stability and convergence times for large-scale CHP problems. Such methodologies represent a significant advancement by balancing computational efficiency and robustness in the presence of system non-convexities.

2.6.2 Primal approach

An alternative to the Lagrangian dual approach for solving the CHP problem is to directly solve the continuous relaxation of the primal UC problem. In this formulation, energy prices can be obtained from the dual variables of the demand balance constraint (2) (UC-1bin) or (1) (UC-3bin), provided that constraints (2.28)-(2.37) (UC-1bin) or (2.40)-(2.45) define the exact convex hull of the feasible region of the original MILP model [34]. When this condition is met, the optimal objective value and dual solutions of the continuous and Lagrangian relaxations are equivalent [41]. Since the primal approach corresponds to the continuous relaxation of the UC problem, it transforms the original MILP into an LP problem. This leads to computational benefits compared to the dual approach, as it allows for the application of standard linear programming algorithms without the need for decomposition techniques.
However, one of the main challenges of this approach lies in the accurate characterization of the convex hull of the feasible set, as any modification in the modeling of generation units may require the development of additional constraints or specialized valid inequalities. In this regard, [42] introduced a convex model that accounts for generation limits, minimum up and down times, linear operation costs, and time-independent start-up costs. Subsequently, [26] incorporated ramping

constraints for start-up and shut-down, and [43] proposed valid inequalities to represent ramping limits across consecutive periods, applicable for time horizons of up to three stages.
In addition to approaches based on the explicit characterization of the convex hull, alternative methodologies have emerged to address the Convex Hull Pricing problem through structurally equivalent reformulations that enable efficient solution procedures without directly representing the full convex polyhedron. These techniques avoid the need to directly model the full convex polytope by leveraging combinatorial and structural insights, along with advanced decomposition and reformulation methods. Among them, two recent proposals stand out due to their promising computational results: on one hand, an algorithm based on the BZ algorithm, which solves a network-flow formulation of the CHP problem and iteratively refines a partition-based structure [44]; and on the other hand, an approach based on DW decomposition, which represents the convex set of feasible solutions as a combination of extreme points derived from UC subproblems [45]. Both methods compute economically consistent prices that minimize uplift payments and are designed to scale efficiently for large-scale systems. The following sections present each algorithm in detail, including their mathematical formulations and main operational advantages.

2.7 State-of-art approaches for convex hull pricing

2.7.1 Level-Method algorithm

This section focuses on the LM algorithm, which is designed around a Lagrangian dual framework to address the CHP problem. The Lagrangian dual formulation introduced in [40] is first presented.

$$\max_{\lambda} \mathcal{L}(\lambda) = \left\{ \sum_{t\in\mathcal{T}} \lambda_t d_t - \sum_{g\in\mathcal{G}} \max_{\substack{p,u,c \\ \in X_g}} \left(\sum_{t\in\mathcal{T}} \lambda_t p_{g,t} - c_g \right) \right\}, \tag{2.57}$$

where equation (2.57) defines c_g as the cost function of generator g. According to [38], $\mathcal{L}(\lambda)$ is a non-smooth concave function, and allows for the decomposition into subproblems per generator, each of which is modeled as a profit-maximization problem (2.58)-(2.59).

$$\max_{p,u,c} \quad \sum_{t\in\mathcal{T}} p_g \lambda_t^{i(g)} - c_g, \tag{2.58}$$

$$\textbf{s.t.} \quad (p_{g,t}, u_{g,t}, c_g) \in \mathcal{X}_g \qquad \forall t \in \mathcal{T}. \tag{2.59}$$

Decomposing the Lagrangian into subproblems for each generator leads to improved computational efficiency, given that solving multiple smaller problems for $g \in \mathcal{G}$ is generally less demanding than solving a centralized problem involving all generating units simultaneously.
The profit maximization model outputs reflect the states of the units and their optimal generation levels, given that they are self-dispatched, which means these are not decisions made by a centralized model. The LM is based on the Kelleys algorithm [46], the central idea is to form an upper approximation of the Lagrangian function $L(\lambda)$ through its supergradients. Where the supergradient g_k and the Lagrangian $L(\lambda^k)$ at iteration k are updated based on the dispatched power of each unit as follows:

$$g_k = \left(D_t - \sum_{g\in\mathcal{G}} \left[P^{min} u^*_{g,t} + p^*_{g,t} \right] \right).$$

The supergradient update and the Lagrangian function evaluated at the energy price determine the cut coefficients a_k ($a_k = g_k$) and the constant coefficient b_k ($b_k = L(\lambda_k) - \langle g_k, \lambda_k \rangle$) at iteration k.
Then, the problem (2.60) is solved, which defines the cutting planes method of Kelley's algorithm. This model provides an update for the Upper Bound ($UB_k = \theta$). On the other hand, the Lower Bound update is obtained from the Lagrangian function evaluated at the energy price ($LB = \max_{j=0\dots k} L(\lambda_j)$).

$$\begin{aligned} \max_{\lambda,\theta} \quad & \theta, \\ \textbf{s.t} \quad & \theta \le \langle a_i, \lambda \rangle + b_i \qquad \forall i = 0 \dots k, \\ & \lambda \in Q. \end{aligned} \tag{2.60}$$

Finally, a price projection problem is solved to obtain the energy price for the next iteration k. As presented in (2.61), the variable in this problem is the price that minimizes the square of the ℓ_2 norm between the variable and the current iteration's price.

$$\begin{aligned} \min_{\lambda} \quad & \|\lambda - \lambda_k\|_2^2, \\ \text{s.t} \quad & \langle a_i, \lambda \rangle + b_i \geq \alpha UB_k + (1-\alpha) LB_k \quad \forall i = ..k. \end{aligned} \tag{2.61}$$

Algorithm 1 Level Method $(\rho^0, \epsilon, iter^{\max}, \alpha)$

Input: $[g_i]_{i=0}^k$ supgradient at $[\rho^i]_{i=0}^k$; L_k Lagrangian value at ρ^k
lower bound $LB \leftarrow -\infty$, upper bound $UB \leftarrow +\infty$
cut coefficients $[a_i]_{i=0}^k$ and constants $[b_i]_{i=0}^k$
for $k = 0$ **to** $iter^{\max}$ **do**
  solve profit maximization (2.58)–(2.59)
  $g_k \leftarrow \left(D_t - \sum_{g \in G}[P_g^{\min} u_{g,t}^* + p_{g,t}^*]\right)$
  $L_k \leftarrow \sum_t \rho_t^k D_t - \sum_t \rho_t^k \left(\sum_g [P_g^{\min} u_{g,t}^* + p_{g,t}^*] - f_g^*\right)$
  $a_k \leftarrow g_k$
  $b_k \leftarrow L(\rho_k) - \langle g_k, \rho_k \rangle$
  if $L_k \geq LB$ **then**
    $LB \leftarrow L_k$
  solve master problem (2.60)
  $UB \leftarrow \theta^*$
  solve projection (2.61) and obtain ρ^*
  $\rho_{k+1} \leftarrow \rho^*$
  if $\dfrac{UB - LB}{|UB|} \leq \epsilon$ **then**
    break

2.7.2 Dantzig-Wolfe decomposition

In [45] the authors propose a DW decomposition to compute the CHP primal approach.
The authors provide an appropiate UC formulation for the decomposition method (2.62)-(2.65):

$$\min_{\mathbf{z}} \quad \sum_{g \in \mathcal{G}} f(\hat{p}_g^k, \hat{u}_g^k) z^k, \tag{2.62}$$

$$\text{s.t} \quad \sum_{g \in \mathcal{G}, k \in \mathcal{K}} \hat{p}_g^k z_g^k = d_t, \quad \forall t \in \mathcal{T}, \tag{2.63}$$

$$\sum_{k \in \mathcal{K}} z_g^k = 1, \qquad \forall g \in \mathcal{G}, \tag{2.64}$$

$$z_g^k \in \{0, 1\}, \qquad \forall g \in \mathcal{G}, k \in \mathcal{K}. \tag{2.65}$$

In this UC formulation, $\hat{p}_g^k$ and $\hat{u}_g^k$ are known parameters associated with a feasible operation k of the generator g, where the first is the power output of the unit and the second is the state of the units (on/off), while z^k is the binary variable associated with the feasible operation schedule of the unit g to optimize. Since the parameters are associated with a feasible operation schedule for generator g, the operational constraints of the generators are included within these values.
To implement the DW algorithm, the proposed binary variables of the UC formulation are relaxed, and the optimization problem is decomposed into a master problem and a subproblem for each generator.

Master problem

The master problem aims to minimize systemic operational costs by considering all possible feasible operation schedules for the generators, ensuring that demand is met. The formulation of the master

problem are presented in (2.66)-(2.69):

$$\min_{\mathbf{z}} \; g^k(z) = \sum_{g\in\mathcal{G},k\in\mathcal{K}} \hat{c}_g^k z^k, \tag{2.66}$$

$$\text{s.t} \sum_{g\in\mathcal{G},k\in\mathcal{K}} \hat{p}_g^k z_g^k = d_t, \quad \forall t \in \mathcal{T}, \qquad : (\lambda_t^{(k)}) \tag{2.67}$$

$$\sum_{k\in\mathcal{K}} z_g^k = 1, \qquad \forall g \in \mathcal{G}, \qquad : (\pi_g^{(k)}) \tag{2.68}$$

$$z_g^k \geq 0, \qquad \forall g \in \mathcal{G}, k \in \mathcal{K}. \tag{2.69}$$

Subproblems

The subproblems are specified for each generator $g \in \mathcal{G}$. The objective function is the reduced cost function, and each problem is defined by the technical requirements of each generator contained in the set of constraints χ_g. These constraints consider generation limits, minimum times, and ramps.

$$\begin{aligned} \min_{\mathbf{z}} \; rc_g^{(k)}(\boldsymbol{p_g}, \boldsymbol{u_g}) = \quad & f_g(\boldsymbol{p_g}, \boldsymbol{u_g}) - \sum_{t\in\mathcal{T}} \lambda_t^{(k)} p_{g,t} - \pi_g^{(k)}, \\ \text{s.t} \quad & (\boldsymbol{p_g}, \boldsymbol{u_g}) \in \mathcal{X}_g \qquad \forall t \in \mathcal{T}. \end{aligned} \tag{2.70}$$

The subproblems generate new columns with new feasible schedules. These new columns are added to the master problem only if their reduced-cost function is a negative value.

Algorithm 2 Dantzig–Wolfe and Column-Generation decomposition for CHP problem

Input: sets $\mathcal{G}, \mathcal{T}$; demand d_t; tolerance $\varepsilon > 0$; maximum $K_{\max}$; feasible initial column sets $\{\mathcal{K}_g^{(0)}\}_{g\in\mathcal{G}}$
Output: prices $\lambda^\star$ (CHP), RMP solution, and final column set
Initialization of restricted RMP with (2.66)–(2.69) using $\mathcal{K}_g^{(0)}$
$k \leftarrow 0$
for $k = 0$ **to** $K_{\max}$ **do**
  1) Solve RMP
  Solve RMP $\Rightarrow$ obtain $z^{(k)}$, duals $\lambda^{(k)}$ from (2.67) and $\pi_g^{(k)}$ from (2.68)
  2) Pricing problems (2.31) per unit
  $\mathcal{N}_{\text{new}} \leftarrow \emptyset$
  for each $g \in \mathcal{G}$ **do**
    Solve (2.70) with $\lambda^{(k)}, \pi_g^{(k)}$
    obtain candidate column k_g^{new} and reduced cost $rc_g^{(k)}$
    if $rc_g^{(k)} < -\varepsilon$ and $k_g^{\text{new}} \notin \mathcal{K}_g^{(k)}$ **then**
      $\mathcal{N}_{\text{new}} \leftarrow \mathcal{N}_{\text{new}} \cup \{(g, k_g^{\text{new}})\}$
  3) Stopping criterion
  if $\mathcal{N}_{\text{new}} = \emptyset$ **then**
    // DW relaxation optimum: balance duals are CHP prices
    $\lambda^\star \leftarrow \lambda^{(k)}$
    break
  else
    4) Add improving columns
    for each $(g, k_g^{\text{new}}) \in \mathcal{N}_{\text{new}}$ **do**
      Add k_g^{new} to $\mathcal{K}_g^{(k+1)}$ and to the RMP

Chapter 3

Mathematical formulations

The present chapter describes the proposed methodology for addressing the Convex Hull Pricing problem, based on a network-flow Unit Commitment formulation. The operational behavior of generating units is initially represented using acyclic space–time graphs, which are subsequently extended to incorporate key technical constraints, including warm-up periods and ramping limitations. These modeling extensions result in the loss of total unimodularity, thereby necessitating the use of mixed-integer linear programming solvers. To efficiently solve the resulting extended formulation, the Bienstock–Zuckerberg algorithm is adopted. This algorithm is an iterative partition-refinement–based column-generation approach that alternates between solving a restricted master problem and separable generator-level subproblems, while also incorporating initialization and preprocessing techniques to accelerate convergence.

3.1 Network flow-based UC formulation

This section presents a formulation of the UC problem as a network flow model based on [44]. A graphical representation of the proposed network model for a generator with a minimum up/down time of 2 hours and for 4 time periods is shown in Figure 3.1. In this framework, nodes represent the operational states of the generator, namely the number of consecutive time periods the unit has been in the on or off state depending on the region (on state and off state). The arcs represent the feasible transitions through which the generator can move between these operational states.

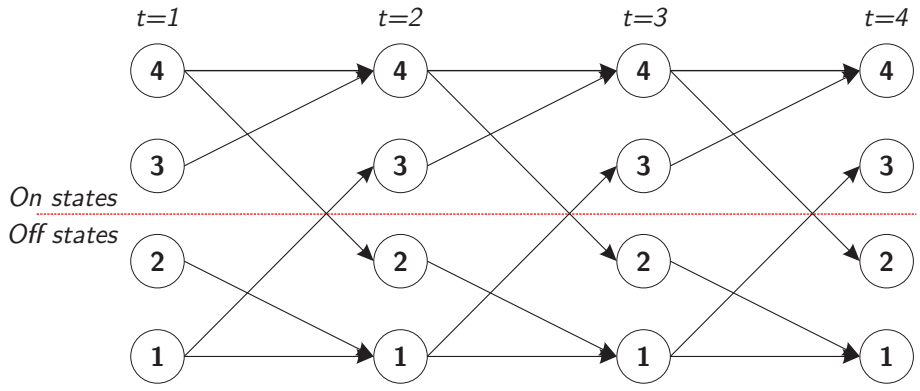


Figure 3.1: Network structure for generator with 2 hours of up/down time

3.1.1 Sets and variables definition

This subsection summarizes the main sets and variables used in the network-flow formulation. First, the sets are defined:

- $g \in \mathcal{G}$: set of generating units.
- $t \in \mathcal{T}$: set of time periods in the planning horizon.

- $e \in \mathcal{E}_g$: set of arcs in the space time network associated with generator g.
- $i \in \mathcal{V}_g$: set of nodes in the space time network associated with generator g.

To define the network variables that represent the generator operations, we consider a network for generator g, which includes $e \in \mathcal{E}_g$ arcs and $n \in \mathcal{N}_g$ nodes for some period of time $t \in \mathcal{T}$. As the network is acyclic, its arc and node structure replicate over time. Consequently, for a time horizon $\mathcal{T}$, the network will contain $\mathcal{E}_g \times \mathcal{T}$ arcs and $\mathcal{N}_g \times \mathcal{T}$ nodes. In the network-flow formulation, let $x_{e,t}$ be a variable associated with the use of arc e at time t, then the power generated through that arc is given by (3.1):

$$P_g^{\min} x_{e,t} \leq p_{g,t} \leq P_g^{\max} x_{e,t} \tag{3.1}$$

Based on (3.1), the variable $y_{e,t}$ can be introduced for each arc e, to represent the fraction of dispatch above the minimum level when the arc is selected. It is defined in (3.2) and (3.3).

$$0 \leq \frac{p_{g,t} - P_g^{\min} x_{e,t}}{P_g^{\max} - P_g^{\min}} \leq x_{e,t} \qquad \forall e \in \mathcal{E}_g, \tag{3.2}$$

$$0 \leq y_{e,t} \leq x_{e,t} \qquad \forall e \in \mathcal{E}_g. \tag{3.3}$$

Given how $y_{e,t}$ is defined, it is a semi-continuous variable bounded by the binary variable $x_{e,t}$. This means that if the arc is inactive ($x_{e,t} = 0$), then $y_{e,t}$ is forced to zero. On the other hand, if the arc e is active ($x_{e,t} = 1$), the variable $y_{e,t}$ can take values in the range $[0, 1]$ Accordingly, in the network-based model, the power generation of units results from a combination of the variables x and y, as described in (3.4).

$$\sum_{e \in \mathcal{E}_g} \left(P_g^{\min} x_{e,t} + (P_g^{\max} - P_g^{\min}) y_{e,t} \right) = p_{g,t} \qquad \forall t \in \mathcal{T}, g \in \mathcal{G}. \tag{3.4}$$

The parameters $q = P_g^{\min}$ and $f = P_g^{\max} - P_g^{\min}$ are defined for the variables x and y respectively. However, these parameters must be adapted to each arc, since arcs associated with the off-state region imply that the unit is not generating power. To capture this behavior, a preprocessing phase is performed to assign specific parameter values to each arc. These arc-dependent parameters are detailed in Table 3.1.

Table 3.1: Network generation parameters per variable

Arc classification	x parameter	y parameter
on-on	$P_g^{\min}$	$P_g^{\max} - P_g^{\min}$
on-off	0	0
off-off	0	0
off-on	$P_g^{\min}$	$P_g^{\max} - P_g^{\min}$

Table 3.1 shows that only the arcs in the on-state region (or in the off-to-on transition) are associated with nonzero generation parameters. For arcs in the off-state region, all parameters associated with generation are set to zero as the generator is turned off.
In terms of generation costs, these include a variable component associated with the dispatched power, as well as fixed costs related to start-up/shut-down decisions and the no-load cost. Based on the network structure from Figure 3.1, start-up and shut-down costs can be assigned to arcs representing transitions between operational states (off-on and on-off arcs, respectively), while variable generation costs and the no-load cost can be allocated to arcs where the unit actually produces energy, namely the on-on and off-on arcs. Table 3.2 details how the operational costs of generator g are assigned based on the four different arc classifications. From Table 3.2, NLC denotes the no-load cost, SDC refers to the shutdown cost, SUC to the startup cost and VC to the variable generation cost. To simplify the notation, we denote the cost parameters for variables x and y as c_x and c_y respectively. With arc-specific parameters, the operational costs associated with each generator and time period t are determined by (3.5).

$$\sum_{e \in \mathcal{E}_g} (c_e x_{e,t} + f_e y_{e,t}) \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{3.5}$$

Table 3.2: Network cost parameters per variable.

Arc classification	x cost	y cost
on-on	$NLC + P^{\min}VC$	$VC(P^{\max} - P^{\min})$
on-off	SDC	0
off-off	0	0
off-on	$SUC + NLC + P^{\min}VC$	$VC(P^{\max} - P^{\min})$

The UC formulation expressed as a network flow model, based on [44], is presented in (3.6)–(3.10), and is called Network-Flow Unit Commitment (NFUC) .

$$\textbf{NFUC:} \quad \min_{x,y} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\sum_{t\in\mathcal{T}}\left(c_e x_{e,t} + f_e y_{e,t}\right), \tag{3.6}$$

$$\textbf{s.t.} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\left(q_e x_{e,t} + s_e y_{e,t}\right) = d_t \qquad \forall\, t\in\mathcal{T}, \tag{3.7}$$

$$\sum_{e\in\delta^+(i)} x_{e,t} \;-\; \sum_{e\in\delta^-(i)} x_{e,t} = b_i \qquad \forall\, i\in\mathcal{V}_g,\ \forall\, g\in\mathcal{G}, \tag{3.8}$$

$$0 \;\le\; y_{e,t} \;\le\; x_{e,t} \qquad \forall\, e\in\mathcal{E}_g,\ \forall\, g\in\mathcal{G}, \tag{3.9}$$

$$x_{e,t} \;\in\; \{0,1\} \qquad \forall\, e\in\mathcal{E}_g,\ \forall\, g\in\mathcal{G}, \tag{3.10}$$

where (3.8) enforces flow conservation and thus guaranties logical consistency of arc selections over time, as discussed in Section 2.3.2. The parameter b_i is only active in the first time period of the planning horizon and then takes a value of zero. This structure ensures that if an arc enters a node at time t, then an arc must depart from that node at time $t+1$, preventing inconsistent operating trajectories. In (3.8), $\delta^+(i)$ denotes the set of arcs leaving node i, while $\delta^-(i)$ denotes the set of arcs entering node i.

The **NFUC** problem is an MILP formulation that captures minimum generation requirements and minimum/maximum power limits. However, the constraints (3.8)–(3.10) define the convex hull of the operating features represented by the network, and the associated constraint matrix preserves a TU structure. Consequently, the relaxation of the power balance constraint leads to the **NFUC-Relaxed** problem of (3.11)-(3.14), which admits integer-optimal solutions from the linear relaxation of x. This provides a computational advantage by avoiding explicit integrality enforcement. Nevertheless, the base formulation remains incomplete from an operational standpoint, as it omits relevant technical constraints. Such features can be incorporated as side-constraints or embedded directly into the formulation, which in turn breaks TU.

$$\textbf{NFUC-Relaxed:} \quad \min_{x,y} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\sum_{t\in\mathcal{T}}\left(c_e x_{e,t} + f_e y_{e,t}\right) - \lambda_t \sum_{t\in\mathcal{T}}\left(d_t - \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\left(q_e x_{e,t} + s_e y_{e,t}\right)\right), \tag{3.11}$$

$$\textbf{s.t.} \quad \sum_{e\in\delta^+(i)} x_{e,t} \;-\; \sum_{e\in\delta^-(i)} x_{e,t} = b_i \qquad \forall\, i\in\mathcal{V}_g,\ \forall\, g\in\mathcal{G}, \tag{3.12}$$

$$0 \;\le\; y_{e,t} \;\le\; x_{e,t} \qquad \forall\, e\in\mathcal{E}_g,\ \forall\, g\in\mathcal{G}, \tag{3.13}$$

$$x_{e,t} \;\in\; \{0,1\} \qquad \forall\, e\in\mathcal{E}_g,\ \forall\, g\in\mathcal{G}. \tag{3.14}$$

3.1.2 Lagrangian relaxation of NFU and marginal-price interpretation

A key structural feature of **NFUC** is that, apart from the system-wide power balance constraint (3.7), all remaining constraints are generator-specific and are confined to each unit space–time network. This separability is central for both pricing and decomposition. In particular, relaxing the power balance constraint with a vector of Lagrange multipliers λ yields a Lagrangian formulation in which each generator becomes independent, because the coupling between units is removed and absorbed into the objective.

For a fixed multiplier vector λ_t, each generator solves an individual subproblem that can be interpreted as a self-dispatch (profit-maximization problem) under the price signal λ_t. The resulting

optimal objective value of the relaxed problem provides a valid dual bound, while multipliers λ_t represent the marginal value of supplying an incremental unit of demand at time t. This perspective motivates the **NFU-Relaxed** formulation introduced and provides the economic interpretation of the price updates used by the subsequent decomposition-based algorithms.

3.1.3 Warm-up periods

Generator warm-up and cool-down periods can be incorporated into the **NFUC** model through a preprocessing phase, without requiring structural changes to the core formulation. This is achieved by adding auxiliary nodes that represent the warm-up states. Although this increases the size of the network (by adding nodes and arcs), it preserves the underlying TU structure. Figure 3.2 illustrates a network with a warm-up time of $T^{\text{warm}} = 3$ hours and a minimum up/down time of 2 hours. To incorporate a cooling-down phase, the same procedure applies, with additional nodes placed in the shutdown region.

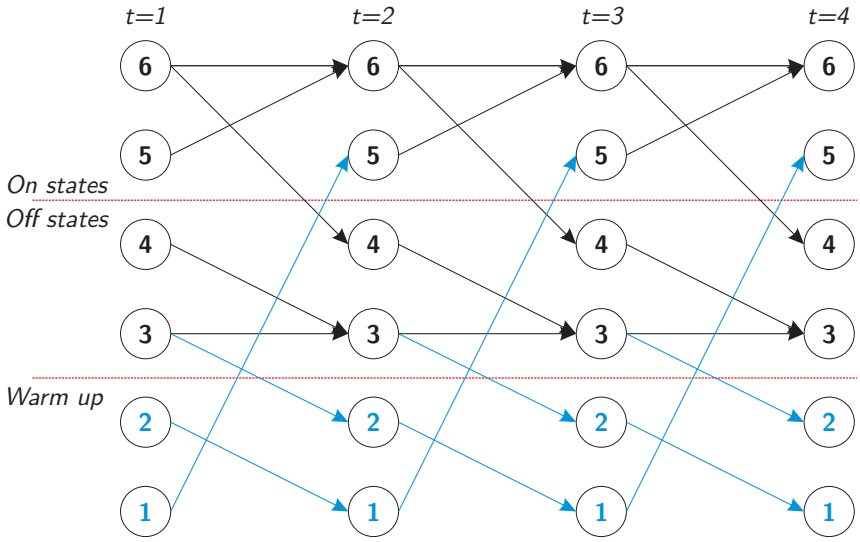


Figure 3.2: Network structure for generator with 2 hours of up/down time and 3 hours of T^{warm}.

The additional nodes are introduced by computing the number of periods required for the unit to reach its minimum output. The equivalent shutdown time is defined as $T^{\text{down}}_{\text{eq}} = T^{\text{down}} + T^{\text{warm}} - 1$. As illustrated in Figure 3.2, once the unit is shut down via an on-off arc, two periods are required to meet the minimum off-time. Thereafter, the unit can either: (i) remain offline without initiating the warm-up, or (ii) enter the warm-up sequence, which spans three periods before reaching the on-state at minimum generation. Once the unit enters the warm-up states, the network-flow structure prevents spurious warm-up trajectories, thereby eliminating false start-up representations. As a result, the commitment and dispatch patterns are more physically consistent and provide a closer approximation to the true cost-minimizing operating behavior of the generator under realistic operational constraints.
Introducing warm-up states requires updating cost parameters while leaving generation parameters unchanged, since the unit does not inject power during the warm-up stages. Table 3.3 presents the cost parameters for the additional "off-warm" and "warm-warm" arcs. Relative to Table 3.2, the cost previously tied to the "off-on" transition is now associated with the "warm-on" arc, which represents the effective start-up transition into the on-state.

Table 3.3: Parameters cost for warm status.

Arc clasifficataion	x cost	y cost
off-warm	$NLC + VCP^{\text{warm}}_{j}$	0
warm-warm	$NLC + VCP^{\text{warm}}_{j+1}$	0

In Table 3.3, P^{warm}_{j} denotes the increase in power associated with the unit heating process, discretized into $j \in \mathcal{J}$ segments until the minimum power level is achieved. The startup cost can

be assigned either at the end of the warm-up period (i.e., on the final "warm-on" arc) or on the "off-warm" arc. With the additional arcs, the corresponding generation parameters by variable and arc type are summarized in Table 3.4.

Table 3.4: Network generation parameters per variable including warm-up.

Arc classification	x parameter	y parameter
on-on	$P_g^{\min}$	$P_g^{\max} - P_g^{\min}$
on-off	0	0
off-off	0	0
off-warm	P_j^{warm}	0
warm-warm	P_j^{warm}	0
warm-on	$P_g^{\min}$	$P_g^{\max} - P_g^{\min}$

Conversely, the new costs by variable and arc type are summarized in Table 3.5, where the start-up cost is, for simplicity, assigned to the "off-warm" arc that initiates the warm-up of the generating unit.

Table 3.5: Network cost parameters per variable with warm-up.

Arc classification	x cost	y cost
on-on	$NLC + P^{\min}VC$	$VC(P^{\max} - P^{\min})$
on-off	SDC	0
off-off	0	0
off-warm	$SUC + NLC + P_j^{\text{warm}}VC$	0
warm-warm	$NLC + P_j^{\text{warm}}VC$	0
warm-on	$NLC + P^{\min}VC$	$VC(P^{\max} - P^{\min})$

3.1.4 Implementation of ramps constraints

To obtain a network representation of the CHP problem that more faithfully captures generator dynamics, the Network Flow Unit (NFU) formulation is extended to include ramping constraints. These constraints, derived from the 1-bin model in Section 2.3.2, limit the maximum change in output between consecutive periods for both upward and downward variations. Let $\mathcal{E}_g^{on}$ denote the set of arcs along which unit g can generate, i.e., arcs in the "on" region together with the "off–on" transition arc. In this representation, x selects the arcs associated with minimum output, while y provides continuous flexibility above the minimum level. Therefore, the intertemporal variation of y is constrained as in (3.15)–(3.16), ensuring that dispatch decisions respect ramp limits. This extension is consistent with standard UC modeling practices for non-convex operating constraints [22].

To achieve a network representation of the CHP problem that more faithfully captures generator dynamics, the **NFUC** formulation is extended to include ramping constraints. These constraints, derived from the 1-bin model described in section 2.3.2, limit the maximum change in output between consecutive periods for both upward and downward variations. Let $\mathcal{E}_g^{on}$ denote the set of arcs along which unit g can generate, i.e., arcs in the "on" region together with the "off–on" transition arc. In this network-based approach, the variable x represents only the selection of arcs associated with the minimum output level, indicating that when an arc in the start-up region is activated, the unit operates at this minimum level, while the variable y provides the flexibility to modulate the delivered power between periods. Therefore, the intertemporal variation of y is constrained as in (3.15)–(3.16), ensuring that dispatch decisions respect ramp limits. This enhancement allows the generation dynamics to be represented more realistically and is consistent with methodologies proposed in the literature to address non-convex Unit Commitment problems with operational ramping constraints [22].

$$\sum_{e \in \mathcal{E}_g^{on}} y_{e,t} - \sum_{e \in \mathcal{E}_g^{on}} y_{e,t-1} \leq r^{\text{up}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{3.15}$$

$$\sum_{e\in\mathcal{E}_g^{on}} y_{e,t} - \sum_{e\in\mathcal{E}_g^{on}} y_{e,t-1} \geq r^{\text{down}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}. \tag{3.16}$$

From (3.15) and (3.16), r^{up} and r^{down} correspond to the upward and downward ramp limits, respectively, normalized by the dispatchable range ($P^{\max} - P^{\min}$). Accordingly:

$$r^{\text{up}}/r^{\text{down}} = \frac{R^{\text{up}}/R^{\text{down}}}{P^{\max} - P^{\min}}. \tag{3.17}$$

For unflexible units (those with identical minimum and maximum output levels) the normalized ramp is assigned a value of 1, ensuring that no issues arise during data loading and that infeasibilities in the model are avoided.

3.1.5 Startup ramp and shutdown ramp constraints

To incorporate start-up and shutdown ramping, the corresponding ramp values are first normalized as:

$$r^{su/sd} = \begin{cases} \min\{1, \frac{R^{su/sd} - P^{min}}{P^{max} - P^{min}}\} & : \text{if} \quad R^{su/sd} > P^{min} \\ 0 & : \text{if} \quad R^{sa/sd} = P^{min} \end{cases}$$

The start up ramp can be incorporated directly by bounding y on the "off–on" transition arc, as shown in (3.18), where $\mathcal{E}_g^{off\text{-}on}$ denotes the set thata contains the "off–on" arc of generator g.

$$y_{e,t} \leq r^{su} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{off\text{-}on}, t \in \mathcal{T}. \tag{3.18}$$

By contrast, the inclusion of the shutdown ramp is less straightforward, shutdown ramping must enforce that the "on–off" transition is feasible only when the units dispatched power in the previous period lies within the shutdown ramp range. In order to represent this shutdown ramp behavior, the constraint presented in (3.19) is incorporated. Here, $\mathcal{E}_g^{on\text{-}off}$ is defined as the set that includes the "on—off" arc, that is, the arc that represents the transition from the on state to the off state.

$$\sum_{j\in\mathcal{E}_g^{on}} y_{j,t-1} \leq r^{sd} x_{e,t} + \sum_{j\in\mathcal{E}_g^{on}} x_{j,t} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{on\text{-}off}, t \in \mathcal{T} \tag{3.19}$$

This constraint can be interpreted as follows: the on–off arc ($x_{e,t}, e \in \mathcal{E}_g^{on\text{-}off}$) is activated only when the dispatched power in the preceding time period is within the shutdown ramp range. Otherwise, if this arc is not used, the dispatched power in the previous period is bounded by the unit maximum power.

Both constraints are included in the model to represent both operational ramping constraints and start-up and shutdown constraints within the convex hull pricing model formulated as a network-flow problem. With the ramp constraints incorporated, the problem that accounts for these technical characteristics is referred to as NFUC Ramps (**NFUCR**) and is presented in (3.20)-(3.28).

NFUCR:

$$\min_{x,y} \quad \sum_{g\in\mathcal{G}} \sum_{e\in\mathcal{E}_g} \sum_{t\in\mathcal{T}} (c_e x_{e,t} + f_e y_{e,t}), \tag{3.20}$$

s.t.

$$\sum_{g\in\mathcal{G}} \sum_{e\in\mathcal{E}_g} (q_e x_{e,t} + s_e y_{e,t}) = d_t \qquad \forall t \in \mathcal{T}, \tag{3.21}$$

$$\sum_{e\in\mathcal{E}_g^{on}} y_{e,t} - \sum_{e\in\mathcal{E}_g^{on}} y_{e,t-1} \leq r^{\text{up}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{3.22}$$

$$\sum_{e\in\mathcal{E}_g^{on}} y_{e,t} - \sum_{e\in\mathcal{E}_g^{on}} y_{e,t-1} \geq r^{\text{down}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{3.23}$$

$$y_{e,t} \leq r^{su} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{off\text{-}on}, t \in \mathcal{T}, \tag{3.24}$$

$$\sum_{j\in\mathcal{E}_g^{on}} y_{j,t-1} \leq r^{sd} x_{e,t} + \sum_{j\in\mathcal{E}_g^{on}} x_{j,t} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{on\text{-}off}, t \in \mathcal{T}, \tag{3.25}$$

$$\sum_{e\in\delta^+(i)} x_{e,t} - \sum_{e\in\delta^-(i)} x_{e,t} = b_i \qquad \forall i \in \mathcal{V}_g, \ \forall g \in \mathcal{G}, \tag{3.26}$$

$$0 \leq y_{e,t} \leq x_{e,t} \qquad \forall e \in \mathcal{E}_g, \ \forall g \in \mathcal{G}, \tag{3.27}$$

$$x_{e,t} \in \{0,1\} \qquad \forall e \in \mathcal{E}_g, \ \forall g \in \mathcal{G}. \tag{3.28}$$

The **NFUCR** problem with dual relaxation of the power balance constraint is presented in (3.29)-(3.30) and is called **NFUR-R** .

$$\textbf{NFUCR-R:} \quad \min_{x,y} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\sum_{t\in\mathcal{T}}(c_e x_{e,t} + f_e y_{e,t}) \tag{3.29}$$

$$-\sum_{t\in\mathcal{T}}\left(\lambda_t d_t - \sum_{g\in\mathcal{G},e\in\mathcal{E}_g}(q_e x_{e,t} + s_e, y_{e,t})\right),$$

$$\textbf{s.t} \quad (3.21) - (3.28). \tag{3.30}$$

The problem described in (3.29)–(3.30) can be addressed by decomposing it into individual subproblems for each generating unit, since the only coupling between their generation variables is through the power balance constraint. The separable structure of the model enables a more computationally efficient solution using an MIP solver, in contrast to the work of [44], where the problems are solved through a shortest-path algorithm, due to the inclusion of ramping constraints.

3.2 Effect of generator ramps constraints on BZ algorithm

Adding ramping constraints to the NFU model significantly alters the polyhedral structure of the underlying network formulation. In the base model without ramp constraints, the polyhedron defined by binary variables x, continuous variables y and constraints preserves the TU property, which allows the problem to be solved as an LP, yielding integer solutions without explicitly enforcing integrality on x [6]. Once rapming constraints are introduced, this property is generally lost, and the convex hull of the problem can no longer be recovered through a simple linear relaxation of the model as in the **NFUC-R** problem. As a result, shortest-path algorithms are no longer applicable, and the **NFUCR-R** formulation becomes an MILP that must be solved with MIP solvers. Key advantages of employing MIP solvers include:

- Advanced algorithms:
  - Branch-and-Bound / Branch-and-Cut: This class of algorithms used by MIP solvers systematically searches the integer solution space, merging a branching enumeration strategy with the generation of valid cuts to tighten the relaxation.
  - Presolve: This procedure removes redundant variables, identifies logical relationships within the model, and simplifies the system of constraints automatically before addressing the core optimization problem.
- Scalability and performance:
  - Parallelization: Modern optimization solvers are capable of using multiple processor cores to simultaneously explore the search tree, generate valid cuts in parallel, and speed up matrix factorizations. This leads to substantial reductions in computation time for separable problems.
  - Scalability: MIP solvers include sophisticated node structures, shared-memory management, and dynamic cut algorithms, allowing the solution of problems with a large number of variables and constraints without significantly compromising computational efficiency.
- Optimality and reliability: Using an MIP solver provides a formal guranty of optimality, whereas manual algorithms rarely offer such certainty. In addition, MIP solvers have been rigorously tested in numerous benchmark instances, which enhances confidence in both the reliability and reproducibility of the results and reduces the risk of errors that could emerge from extending a shortest-path procedure.

3.3 Characteristics and impact of the MIP solver presolve phase

As discussed previously , the preprocessing stage provided by MIP solvers is a powerful tool that allows redundant variables and constraints to be removed from the model, leading to a smaller

problem size and improved computational efficiency. The following section presents the conditions under which preprocessing contributes to reducing the number of variables and constraints.

3.3.1 Initial conditions preprocess

Consider the generator network illustrated in Figure 3.3, where the initial condition corresponds to node 4 (i.e., the generator is already online before the simulation starts). As shown, arcs associated with the initial periods become redundant because the initial operating state is fixed exogenously.

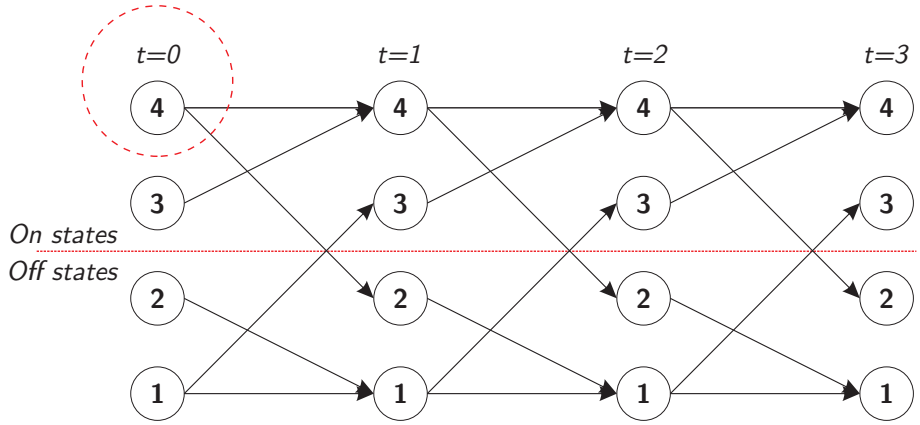


Figure 3.3: Network representation with initial condition.

The variables and constraints associated with redundant arcs and nodes are removed by the MIP solver preprocessing stage, resulting in a graph as shown in Figure 3.4. As observed, the arcs and nodes corresponding to the initial periods (depicted in light gray) are reduced, thus eliminating paths that are not feasible under the generator initial condition. For the generator illustrated, with a minimum up/down time of 2 hours, after a 4-hour horizon, no arcs or nodes are redundant and all are considered in the model decision process.

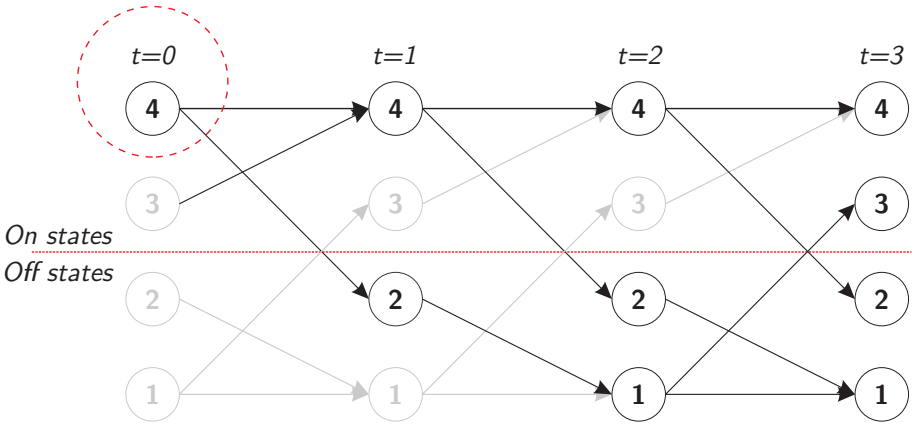
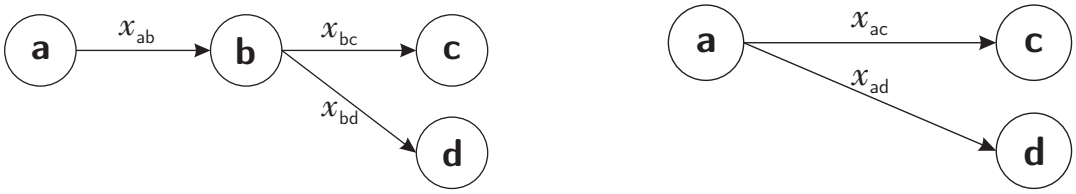


Figure 3.4: Network representation with preprocess of MIP solver at initial time horizon.

3.3.2 Arcs preprocess

A second preprocessing mechanism reduces variables (and associated constraints) at nodes with multiple outgoing arcs. Figure 3.5a illustrates an example network in which node b acts as a fork node. In this representation, three binary variables are associated with outgoing arcs, yet node b is redundant and can be eliminated without affecting the set of feasible paths. Consequently, preprocessing removes this node and reduces the number of arcs, as shown in Figure 3.5b. The resulting formulation uses only two binary variables to describe the same feasible trajectories, thereby reducing both variables and constraints.



(a) Arcos without preprocess.

(b) Arcs with preprocess.

Figure 3.5: Effect of MIP solver preprocessing: (a) Network without preprocessing and (b) Network after preprocessing.

In larger-scale instances, such reductions are particularly valuable: in MILP, fewer binary variables typically translate into smaller branch-and-bound trees and more effective pruning. Moreover, preprocessing can strengthen the linear relaxations solved at intermediate nodes by removing redundant structure, improving the overall efficiency of the search.

3.4 Bienstock-Zuckerberg based algorithm

The **NFUCR** problem is solved using a Bienstock–Zuckerberg-based algorithm, whose generic structure is summarized next.

The BZ algorithm relies on a specialized column-generation strategy. Instead of solving the full LP relaxation, BZ iteratively builds the solution by adding columns (solution patterns) induced by partitions [18]. Each column corresponds to a feasible extreme point of a network subproblem associated with an individual generator. The overall procedure alternates between: (a) solving a restricted master problem defined by the current set of columns, and (b) solving a pricing subproblem to identify a new candidate column that can improve the objective value.

To implement the algorithm, the feasible operating path of each generator is first obtained from the generator subproblems. This path identifies the set of arc-usage variables activated in the subproblem solution for generator g. Based on the activated pattern, a partition is defined as a set of indices that group variables associated with the same decision behavior in the generator subproblem solution (i.e. variables that must take identical values in the final solution). The solution of the subproblem for generator g is illustrated in Figure 3.6 (for simplicity, warm-up times are omitted).

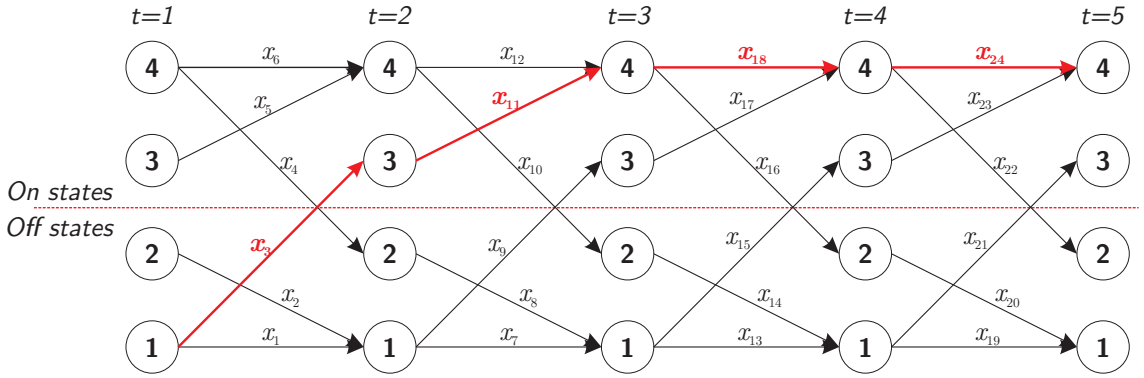


Figure 3.6: Example optimal solution of a generator

In Figure 3.6, the optimal operating trajectory is the path highlighted in red. The feasible path are the set of activated arcs $N^1 = \{x_1 = x_{g,t} \in \mathcal{E}g : \bar{x}g, t = 1\}$; in the example, $N^1 = \{x_3, x_{11}, x_{18}, x_{24}\}$. The remaining arcs define the inactive set $N^0 = \{x_0 = x_{g,t} \in \mathcal{E}g : \bar{x}g, t = 0\}$. In this solution, two partitions can be defined, corresponding to the active and inactive arc sets (N^1 and N^0). From the solution of the generator subproblem, an additional constraint can be introduced to enforce that all variables within a partition share the same value. That is, if one arc of a partition is selected, then all arcs in that partition must be selected. This constraint is written as:

$$\mathrm{H}x = 0.$$

The matrix H represents a coefficient matrix that enforces equality between variables belonging to the same partition. For example, given the partition N^1 from the network solution in Figure 3.6, this constraint guaranties that $x_3 = x_{11}$, $x_{11} = x_{18}$ and $x_{18} = x_{24}$.
Then, since the generator subproblem corresponds to a relaxed version of the original problem, it is possible to take the original formulation and integrate the constraints associated with the partitions. By adding partition constraints that restrict the feasible region, the original formulation is transformed into a tighter restricted problem. Accordingly, a master problem can be defined whose objective value provides a valid upper bound for the CHP problem when the binary variables are relaxed to their continuous counterparts. Thus, the problem described by (3.31)-(3.40), referred to as **Master Problem** (MP) at iteration (k), is established. This problem consists of the original formulation with the addition of partition-related constraints and the relaxation of the binary variable x.

$$\mathbf{MP}^{(k)}: \quad \min_{x,y} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\sum_{t\in\mathcal{T}}\big(c_e x_{e,t} + f_e y_{e,t}\big), \tag{3.31}$$

$$\textbf{s.t.} \quad \sum_{g\in\mathcal{G}}\sum_{e\in\mathcal{E}_g}\big(q_e x_{e,t} + s_e y_{e,t}\big) = d_t \qquad \forall t \in \mathcal{T}, \tag{3.32}$$

$$\sum_{e\in\mathcal{E}_g} y_{e,t} - \sum_{e\in\mathcal{E}_g} y_{e,t+1} \le r^{\text{up}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{3.33}$$

$$\sum_{e\in\mathcal{E}_g} y_{e,t} - \sum_{e\in\mathcal{E}_g} y_{e,t-1} \ge r^{\text{down}} \qquad \forall g \in \mathcal{G}, t \in \mathcal{T}, \tag{3.34}$$

$$y_{e,t} \le r^{su} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{off\text{-}on}, t \in \mathcal{T}, \tag{3.35}$$

$$\sum_{j\in\mathcal{E}_g^{on}} y_{j,t-1} \le r^{sd} x_{e,t} + \sum_{j\in\mathcal{E}_g^{on}} x_{j,t} \qquad \forall g \in \mathcal{G}, e \in \mathcal{E}_g^{on\text{-}off}, t \in \mathcal{T}, \tag{3.36}$$

$$\sum_{e\in\delta^+(i)} x_{e,t} - \sum_{e\in\delta^-(i)} x_{e,t} = b_i \qquad \forall i \in \mathcal{V}_g,\ \forall g \in \mathcal{G}, \tag{3.37}$$

$$0 \le y_{e,t} \le x_{e,t} \qquad \forall e \in \mathcal{E}_g,\ \forall g \in \mathcal{G}, \tag{3.38}$$

$$\mathrm{H}^{(k)} x_{g,t} = 0 \qquad \forall e, t \in \mathcal{P}_g, \forall g \in \mathcal{G}, \tag{3.39}$$

$$0 \le x_{e,t} \le 1 \qquad \forall e \in \mathcal{E}_g,\ \forall g \in \mathcal{G}, \forall t \in \mathcal{T}. \tag{3.40}$$

Since the BZ algorithm operates iteratively, the sets of active and inactive arcs obtained from the subproblems may change over iterations, requiring the partitions used in constraint (3.39) to be updated accordingly. Starting from an initial subproblem solution, the **MP** problem is solved, where the dual variable of the power balance constraint (3.32) provides an updated Lagrangian dual value (market-clearing price) used in the objective function of the subproblem. This enables new feasible operating paths to be identified for each generator.
The partition updates are performed by iteratively merging the new sets of active and inactive arcs with those fixed in earlier iterations. Consider iteration $(k-1)$, in which the partitions associated with generator g are illustrated in Figure 3.6, and the result of its subproblem at iteration (k), shown in Figure 3.7, which determines the corresponding active and inactive arcs.

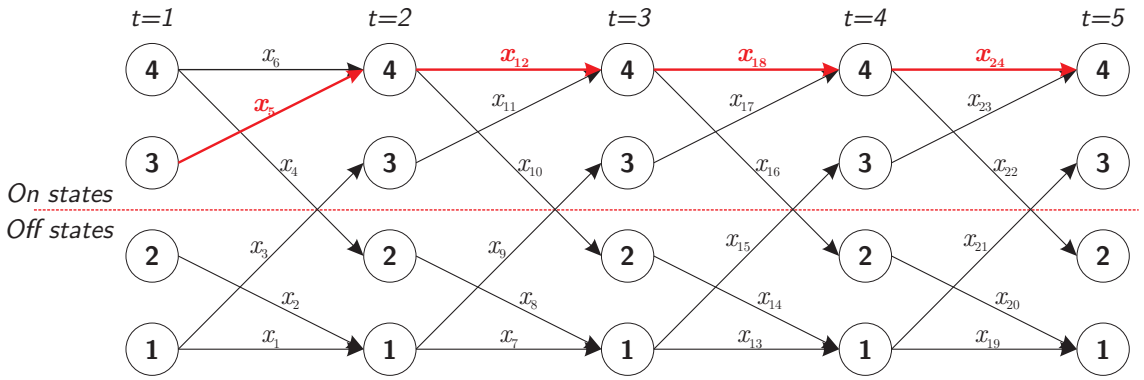


Figure 3.7: Optimal solution at iteration (k) for the example generator

The set of active and inactive arcs in iteration (k) are $N^1_{(k)} = \{x_5, x_{12}, x_{18}, x_{24}\}$ and $N^0_{(k)} = \{x_0 = x_{g,t} \in \mathcal{E}g : \bar{x}_{g,t} = 0\}$. To update the partitions at iteration (k), it is necessary to intersect each set from iteration $(k-1)$ with the results of the subproblem.
To update the iteration, we first define the partitions in iteration (k-1) from Figure 3.6, which are $I^1_{(k-1)} = \{x_3, x_{11}, x_{18}, x_{24}\}$ and $I^2_{(k-1)} = \{x_0 = x_{g,t} \in \mathcal{E}g : \bar{x}_{g,t} = 0\}$. Then, by intersecting each partition of iteration (k-1) with the results of iteration (k), the new partition sets are obtained: $I^1 = I^1_{(k-1)} \cap N^1_{(k)} = \{x_{18}, x_{24}\}$, $I^2 = I^1_{(k-1)} \cap N^0_{(k)} = \{x_3, x_{11}\}$, $I^3 = I^2_{(k-1)} \cap N^1_{(k)} = \{x_5, x_{12}\}$ and $I^4 = \{\mathcal{E} - (I^1 \cup I^2 \cup I^3)\}$, this means that I^4 are the set of partitions that do not have intersections with the other sets. Then, for each subsequent iteration, the partition update process is repeated.

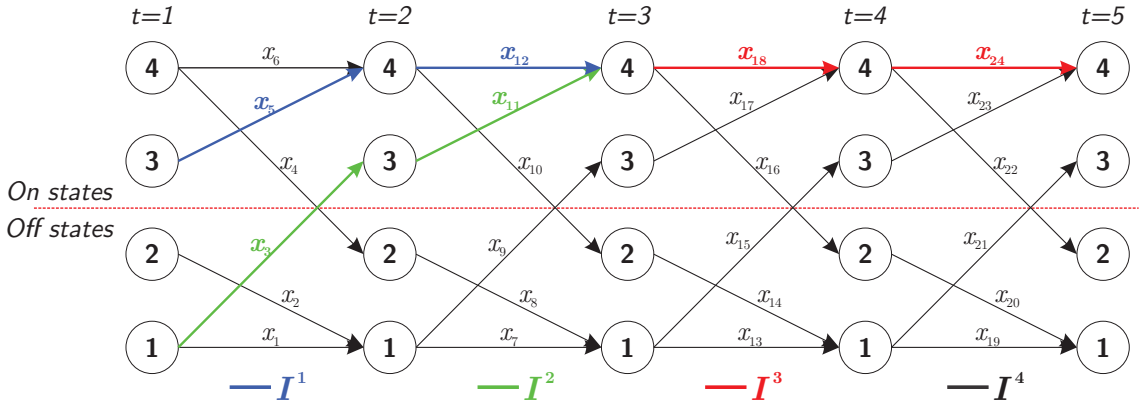


Figure 3.8: Partitions at iteration (k) for a generator with two on-time and off-time periods

3.4.1 Algorithm initialization

To improve the convergence speed of the algorithm, initialization techniques can be applied with the main purpose of rapidly generating partitions, enabling the algorithm to begin with a more restricted feasible solution space.
The initialization of the BZ algorithm is carried out by applying a relaxation of the CHP problem that is solved independently for each time period. As a consequence of this temporal decomposition, the resulting model neglects arc divergence constraints and ramping constraints. This formulation is presented in (3.41)–(3.44) and is referred to as RCHP (Relaxed Convex Hull Pricing).

$$\textbf{RCHP:} \quad \min_{x,y} \quad \sum_{g\in\mathcal{G}} \sum_{e\in\mathcal{E}_g} \left(c_e x_e + f_e y_e\right), \tag{3.41}$$

$$\textbf{s.t.} \quad \sum_{g\in\mathcal{G}} \sum_{e\in\mathcal{E}_g} \left(q_e x_e + s_e y_e\right) = d_t \tag{3.42}$$

$$0 \le y_e \le x_e \qquad \forall\, e \in \mathcal{E}_g,\ \forall\, g \in \mathcal{G}, \tag{3.43}$$

$$0 \le x_e \le 1 \qquad \forall\, e \in \mathcal{E}_g,\ \forall\, g \in \mathcal{G}. \tag{3.44}$$

The methodology consists in solving the problem for a representative sample of demand levels in the instance. From these independent solutions, the price from the dual variable of the power balance constraint is obtained. Subsequently, a linear regression is fitted using the corresponding (demand, dual) pairs, allowing the estimation of market-clearing prices for demand values not included in the sample. A schematic representation of the selected samples and the fitted regression line is provided in Figure 3.9.
Then, using the price curve obtained from the interpolation process, the subproblems associated with each generating unit can be solved. These subproblems determine, for each generator, which arcs are active and which are inactive. As the sampling process is repeated, the partitions of the generators are gradually refined and expanded, without requiring

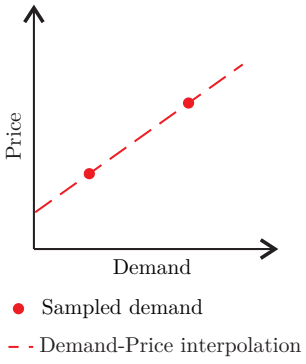


Figure 3.9: Scheme of price-demand interpolation.

the complete CHP problem to be solved. Consequently, the algorithm can be initialized with more tightly bounded solution spaces for each generator, as previously discussed. It is worth noting that, when no initialization strategy is employed, the BZ algorithm begins with two initial partitions per generator, corresponding to the sets of active and inactive arcs obtained from the subproblem solution.

The pseudocode of the Bienstock–Zuckerberg–based algorithm is shown in Algorithm 3. In the pseudocode, the optimality criterion is defined through the expression $\frac{UB-LB}{UB}$, which specifies the algorithm termination condition when the relative optimality gap is less than or equal to a defined tolerance ϵ.

Algorithm 3 Bienstock-Zuckerberg algorithm

stopping criteria (ϵ), initial price $\lambda^{(k)} = \lambda^0$ and initial partitions
$k \leftarrow 0$
while *optimality gap*$> \epsilon$ **do**
  $LB^{(k)} \leftarrow 0$, *optimality gap*$= 0$
  for g in $\mathcal{G}$ **do**
    Solve SP – with $\lambda^{(k)}$ price
    From optimal solution $\bar{x}_g$ get active and inactive arcs:
    $\mathrm{N}^1 = \{i \in \mathrm{N} : \bar{x}_g = 1\}, \mathrm{N}^0 = \{i \in \mathrm{N} : \bar{x}_g = 0\}$
    $LB^{(k)} += SBFO$
    Update Partitions $\mathcal{N}^{(k)}$
  Solve MP (3.31)-(3.40) with partitions $\mathcal{N}^{(k)}$
  Get price update $\lambda^{(k)}$ from dual value of constraint (3.32)
  $UB \leftarrow$ (3.31)
  optimality gap $= \frac{UB-LB}{UB}$

Chapter 4

Computational experiments

The present chapter presents the computational experiments, beginning with a description of the real-world case studies based on the California, FERC, and Belgium systems used to validate the proposal. Then, the impact of operational ramp constraints on market-clearing prices and uplift payments is analyzed, demonstrating that the extended network formulation leads to higher energy prices due to reduced system flexibility. Furthermore, the study quantifies the critical role of the MIP presolve stage, which is shown to reduce computational time by an average of 65%. Lastly, the performance of the proposed Bienstock-Zuckerberg algorithm is benchmarked against the Level Method and Dantzig-Wolfe decomposition, confirming its superior stability and convergence speed.

4.1 Data and case studies

To evaluate the performance of the proposed algorithm and the various analyzes associated with it, Table 4.1 presents a summary of the systems and instances examined in this study.

Table 4.1: Simulated instances data and problem variables numbers

System	Generators	Time horizon (h)	Number of instances
California	610	48	20
FERC	934	48	14
RTS-GMLC	73	48	12
Belgium	68	48	8

The algorithm and associated optimization models are implemented in Julia v1.12.4 using the JuMP v1.26.0 optimization package. The computational experiments were carried out on a computer equipped with an AMD Ryzen 9 9500HX @ 3.30 GHz and 16 GB of RAM @ 3200 MT/s. This hardware configuration was kept fixed throughout all experiments to ensure the comparability of the reported computational times between instances and methodologies.

4.2 Computational experiments

With the aim of comprehensively evaluating the proposed methodology, four computational experiments were designed to address different aspects of the algorithm and the model. First, a comparative analysis of the performance of different MIP solvers is carried out, with the purpose of selecting the one that offers the best trade-off between computational efficiency and numerical stability within the iterative scheme of the BZ algorithm, where the cumulative cost per iteration amplifies the impact of the solver choice.
Second, the impact of the proposed initialization strategy based on the RCHP model is evaluated, comparing it against the conventional UC-LP alternative, analyzing metrics such as the number of iterations required, total convergence time, and the quality of the initial bounds.

Third, the effect of incorporating operational ramp constraints on the market-clearing price is analyzed, comparing the prices obtained with and without such constraints, as well as their impact on uplift payments. This analysis allows quantifying the estimation error introduced when these constraints are omitted from the model.
Finally, the performance of the proposed BZ-based algorithm is benchmarked against state-of-the-art methods, namely the Level Method and the Dantzig–Wolfe decomposition, evaluating convergence times and robustness across instances on the four test systems considered.

4.3 Computational performance evaluation

This section presents the computational experiments conducted to evaluate key aspects of the proposed algorithm implementation, with a particular focus on its computational efficiency and the factors that influence its overall performance. Specifically, the experiments address four main aspects: the selection of the most suitable MIP solver for the generator-level subproblems, the impact of the initialization strategy on convergence speed and iteration count, the effect of the MIP solver presolve phase on computational times, and the influence of incorporating operational ramp constraints on the resulting market-clearing prices and uplift payments. Each of these aspects is analyzed independently, allowing a systematic identification of the design decisions that most significantly affect the practical performance of the BZ-based algorithm in systems and instances of varying complexity.

4.3.1 Performance profiles

Performance profiles, introduced by Dolan and Moré [47], provide a methodology to comparing the relative performance of a set of solution data over a collection of test problems. For each solver $s \in \mathcal{S}$ and the problem instance $p \in \mathcal{P}$, the performance ratio is defined as:

$$r_{p,s} = \frac{t_{p,s}}{\min_{s' \in \mathcal{S}} t_{p,s'}}. \tag{4.1}$$

From (4.1), $t_{p,s}$ denotes a performance metric of interest, typically the computation time of the wall clock. The performance profile of solver s is then the empirical cumulative distribution of these ratios as shown in (4.2):

$$\rho_s(\tau) = \frac{1}{n_p} \left| \{ p \in \mathcal{P} : r_{p,s} \leq \tau \} \right|, \quad \tau \in [1, \tau_{\max}]. \tag{4.2}$$

The value $\rho_s(1)$ indicates the fraction of instances for which the solver s was the fastest (efficiency), while $\rho_s(\tau_{\max})$ indicates the fraction of instances successfully solved within the time limit, regardless of speed (robustness). Solvers that fail to produce a solution within the time budget are assigned $r_{p,s} = \tau_{\max}$.

4.3.2 Performance analysis of MIP solvers on subproblems

The inclusion of operational ramp constraints into the network-flow formulation breaks the TU property of the generator subproblems, as discussed in Section 3.1.6. As a consequence, these subproblems can no longer be solved as standard LP problems through shortest-path algorithms, but instead become MILP problems that must be solved to optimality using a MIP solver. Since the BZ algorithm requires the repeated solution of these subproblems at every iteration, the choice of solver has a direct and compounding effect on the total computational time, making this selection a critical design decision in the practical implementation of the proposed methodology. To address this, three widely used MIP solvers are evaluated within the BZ algorithm framework: CPLEX, Gurobi, and HiGHS. The evaluation considers not only computational efficiency in terms of runtime, but also numerical stability in consistently finding the optimal solution across all systems and instances tested. All experiments are conducted using the initialization strategy described in section 3.4.1.
The results are presented through performance profiles for the CA and RTS-GMLC systems in Figure 4.1 and Figure 4.2, respectively, which provide a statistically robust comparison of the solver behavior between instances of varying complexity. In both systems, Gurobi emerges as

the best-performing solver in terms of computational time, achieving the fastest execution in approximately 75 % of the evaluated instances, while CPLEX ranks as the fastest solver in the remaining 25 %. In both cases, the open source solver HiGHS exhibits a notably less efficient computational performance relative to the two commercial solvers, presenting consistently higher runtimes across all instances.

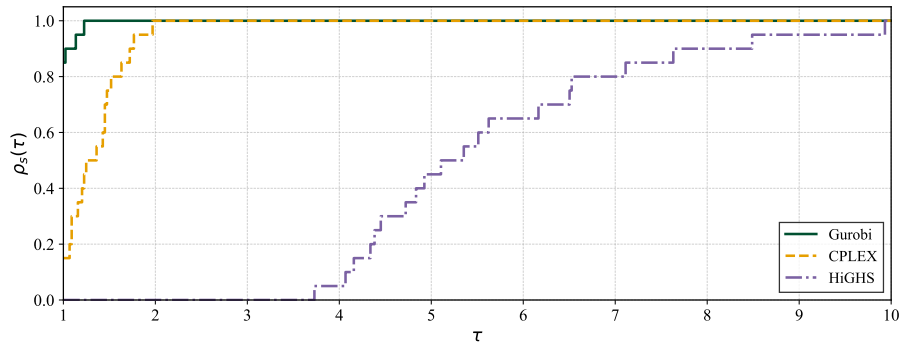


Figure 4.1: Performance profiles of MIP solvers for CA system.

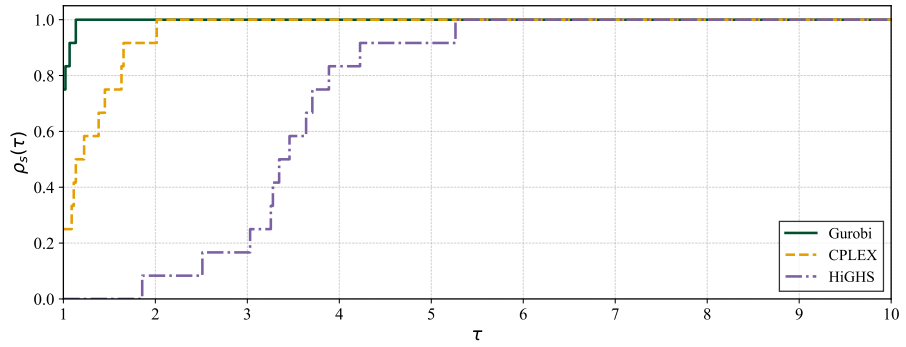


Figure 4.2: Performance profiles of MIP solvers for RTS-GMLC system.

Beyond raw computational speed, a critical requirement in the practical implementation of optimization algorithms is robustness, understood as the ability to deliver consistent and stable performance across instances of varying complexity and system configurations. A solver that achieves fast average runtimes but exhibits high variability between instances may compromise the reliability of the overall methodology, particularly in real market applications where predictable convergence behavior is essential. To assess this dimension, Table 4.2a and Table 4.2b report a set of relevant performance metrics for each solver and system, including the median, minimum, and maximum computational time per instance, as well as the mean and standard deviation (σ) across all instances. These metrics provide a more complete picture of the behavior of the solver than average runtime alone, allowing the identification of solvers that combine computational efficiency with low performance variability. The standard deviation, in particular, serves as a direct indicator of robustness since a lower value reflects a more stable and predictable convergence behavior across the evaluated instances.

Table 4.2: Comparison of MIP solvers metrics for CA and RTS-GMLC systems.

(a) Computational metrics for CA system.

Metrics (s)	CPLEX	Gurobi	HiGHS
Min	36.45	**23.65**	152.59
Max	63.23	**58.75**	265.97
Median	47.46	**35.3**	193.33
Mean	47.41	**37.52**	197.95
σ	**8.35**	9.91	32.40

(b) Computational metrics for RTS-GMLC system.

Metrics (s)	CPLEX	Gurobi	HiGHS
Min	10.59	**8.42**	24.94
Max	32.15	**19.79**	65.93
Median	16.90	**13.54**	47.04
Mean	17.48	**13.99**	45.98
σ	5.20	**3.63**	11.21

4.3.3 Impact of inicialization process

This section evaluates the performance of the algorithm initialization methodology proposed in Section 3.4.1 and examines its effects on the efficiency and numerical stability of the BZ algorithm. The proposed strategy is compared with an alternative and widely adopted initialization procedure that first solves a relaxed UC formulation as a LP by relaxing the binary variables. The energy price obtained from this relaxed UC solution is then used as the initial vector of Lagrange multipliers. Using these prices as input, the profit-maximization subproblems for each generating unit are subsequently solved to generate the initial information required by the BZ algorithm as shown in Figure 4.3. This procedure initializes the scheme with two partitions, yielding an initial approximation of the partitioning structure that the algorithm progressively refines. The comparison between both strategies aims to quantify differences in metrics such as the number of required iterations, total computational time, and the quality of the initial bounds (e.g., primal/dual bounds), thereby justifying the initialization procedure adopted for the remainder of the experiments.

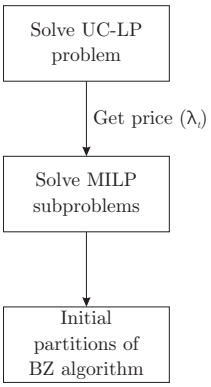


Figure 4.3: UC-LP initialization scheme.

To empirically evaluate the initialization techniques considered, we employed the test systems described above. In particular, Figure 4.4 presents the computational times obtained by running the BZ algorithm on the CA system, directly comparing both initialization methods. In general terms, the results demonstrate that the proposed initialization strategy delivers a more refined set of initial partitions to the BZ algorithm compared to the UC-LP alternative. Since the partition structure directly determines the feasible solution space explored by the master problem at each iteration, a richer and more informative initialization translates into a tighter restricted feasible region from the outset, reducing the number of corrective partition refinements required in subsequent iterations. As a result, the algorithm can reach the prescribed optimality gap tolerance in fewer iterations as shown in Figure 4.5 and with lower overall computational time, as consistently observed across the evaluated instances. This confirms that investing computational effort in the initialization phase yields a net benefit in total convergence time, particularly in large-scale systems where each BZ iteration involves the solution of a significant number of MILP subproblems.

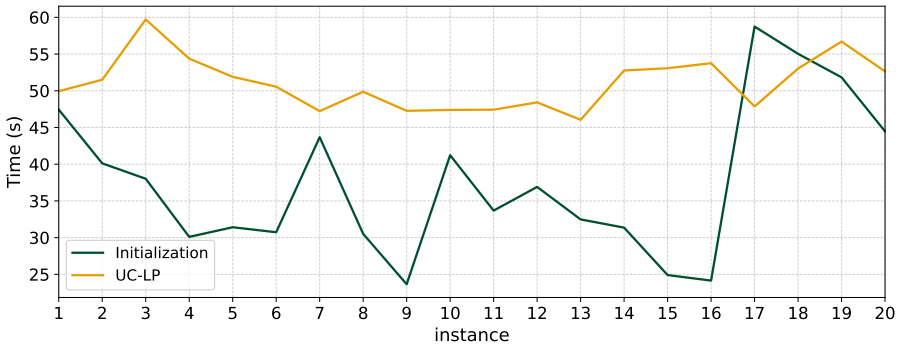


Figure 4.4: Computational times for proposed initialization and UC-LP.

Table 4.3 provides aggregated statistics, including maximum and minimum runtime, median runtime, mean runtime, and standard deviation, capturing both average performance and dispersion between instances. From these results, it can be observed that the proposed initialization methodology consistently reduces computational times compared to the alternative based on solving the relaxed UC, yielding an average runtime reduction of approximately 27% (around 14 seconds). It is important to note that, while the proposed initialization strategy consistently reduces average computational times, it introduces greater variability in solution times across instances, as reflected by the higher standard deviation reported in Table 4.3 compared to the UC-LP alternative. This behavior is inherent to the nature of the proposed approach, since the initialization is based on

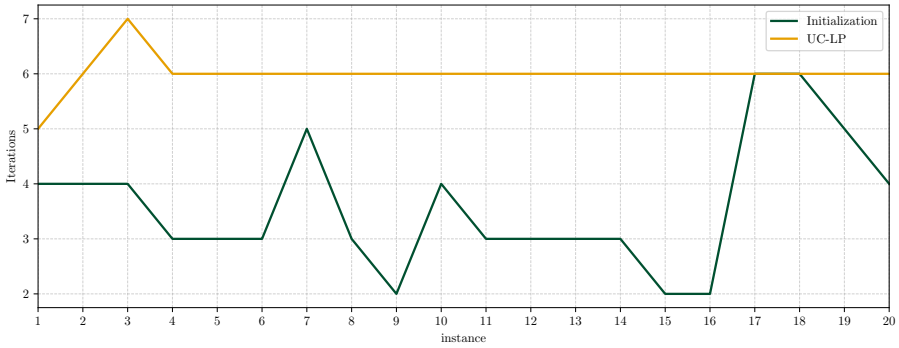


Figure 4.5: BZ Iterations to convergence on CA instances.

sampling a representative set of operating points from the demand profile and subsequently extrapolating the energy price for the remaining hours through linear regression, the quality of the initial partitions set is directly dependent on how well the sampled points capture the underlying price structure of each particular instance. Consequently, instances whose demand profiles are well represented by the sampled points will benefit from a tighter and more informative initialization, leading to faster convergence, while instances where the extrapolation is less accurate may require additional iterations to correct the initial approximation. This trade-off between average runtime reduction and increased variability should be considered when applying the proposed initialization strategy in practice.

Table 4.3: Metrics of the BZ algorithm initialization technique.

Metrics (s)	Initialization	UC-LP
Min	**23.65**	46.05
Max	**58.75**	59.72
Median	**35.29**	51.03
Mean	**37.52**	51.07
σ	9.91	**3.46**

Figure 4.6, 4.7, and 4.8 illustrate the evolution of the optimality gap across BZ algorithm iterations for instances 2, 13, and 20 of the California system, respectively. These instances are selected to represent different demand scenarios and levels of algorithmic difficulty within the system. The trajectories exhibit a consistent behavior across all three cases: when initialized using the UC-LP approach, the algorithm starts from systematically larger initial gap values compared to those obtained with the proposed initialization. This larger initial gap reflects the coarser partition structure generated by the UC-LP strategy, which provides a less accurate approximation of the optimal price vector and therefore requires the algorithm to perform a greater number of partition refinement steps before satisfying the convergence tolerance.

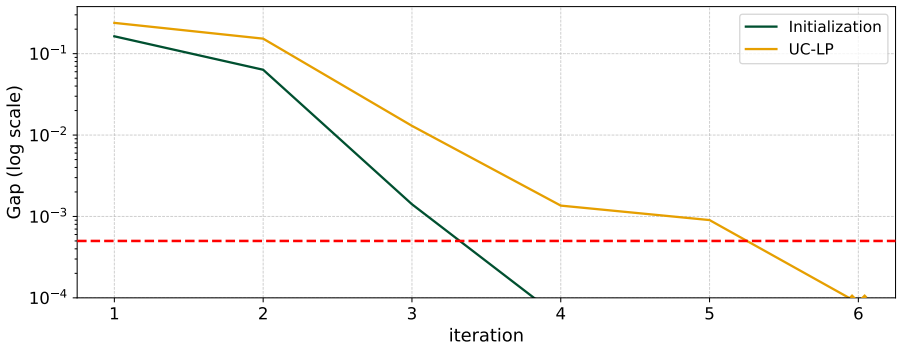


Figure 4.6: Evolution of the optimality gap for instance 2 of CA system.

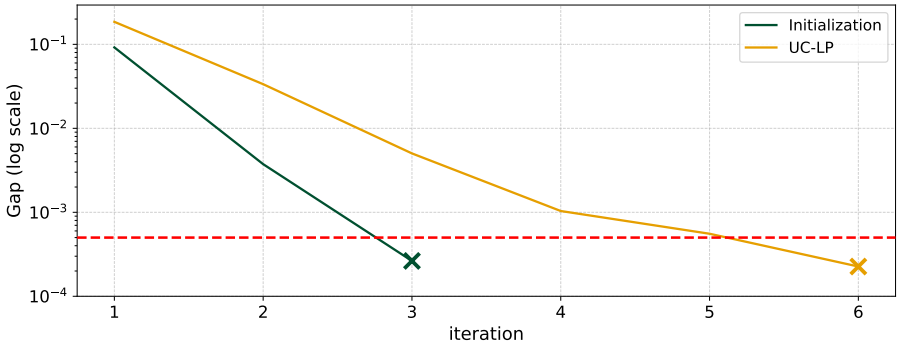


Figure 4.7: Evolution of the optimality gap for instance 13 of CA system.

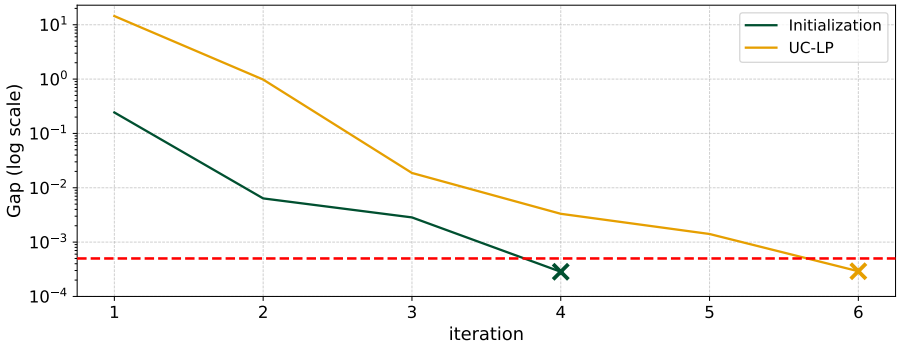


Figure 4.8: Evolution of the optimality gap for instance 20 of CA system.

4.3.4 Impact of MIP presolve

A relevant analysis involves assessing the impact of the MIP solver presolve phase on computational times when addressing the CHP problem using the Bienstock–Zuckerberg–based algorithm. To this end, the proposed methodology is solved using the same initializations and instances, with the only difference being that in one case the MIP solver presolve feature is enabled, while in the other case it is disabled. Figure 4.9 shows the comparison of the computational times obtained for 20 instances of the California system.

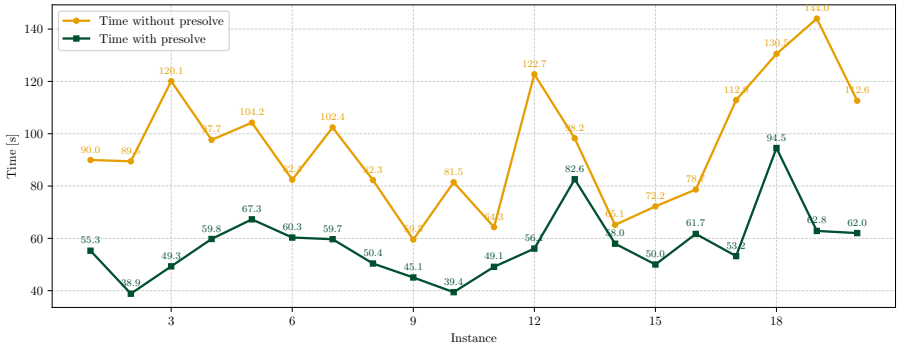


Figure 4.9: Computational time comparison on California instances with and withou MIP presolve

The results show that computational times increase significantly in most instances when the presolve phase is not enabled. This occurs because, without the MIP solver presolve, the problem is formulated with a larger number of variables and constraints, which increases the computational effort required to obtain the optimal solution through MIP solver methods such as branch-and-bound and related techniques.

Table 4.4: Computational times for California instances and impact of the presolve phase.

CA Instance	Time with presolve (s)	Time without presolve (s)	Time increase (%)
1	55.28	89.96	62.74
2	38.85	89.46	130.27
3	49.30	120.09	143.59
4	59.76	97.66	63.42
5	67.27	104.23	54.94
6	60.31	82.43	36.68
7	59.68	102.40	71.58
8	50.38	82.27	63.30
9	45.05	59.53	32.14
10	39.43	81.45	106.57
11	49.13	64.30	30.88
12	56.09	122.73	118.81
13	82.59	98.25	18.96
14	57.97	65.15	12.39
15	50.00	72.23	44.46
16	61.70	78.66	27.49
17	53.22	112.86	112.06
18	94.53	130.52	38.07
19	62.84	144.01	129.17
20	62.02	112.61	81.57
Mean	57.77	95.54	65.38

Table 4.4 provides a detailed presentation of computational times and the percentage increase obtained when the presolve phase is not applied. On average, a 65 % increase is obtained compared to the results with presolve enabled. The maximum increase in computational time corresponds to instance 3, which exhibits a 143 % increase, clearly demonstrating the impact of the variable and constraint reductions performed during the presolve stage when solving MILP problems.

4.4 Approach comparison and benchmarking

The proposed methodology is validated by comparing it with algorithms available in the literature, namely the Level Method and the Dantzig–Wolfe decomposition. The three methodologies are applied to the California, FERC, RTS-GMLC and Belgium systems and assessed in terms of computational time required to reach convergence, with a prescribed relative gap of $\epsilon = 0.05\%$. In this section, market-clearing prices resulting from Convex Hull Pricing are not compared, as the same convex hull is modeled across all methods, leading to identical energy price outcomes.

4.4.1 California system

Figure 4.10 shows the performance profiles of the three approaches. The performance profile analysis conducted on California system instances reveals a greater computational efficiency of the proposed BZ algorithm over the LM and DW decomposition approaches. In terms of empirical efficiency, the BZ algorithm consistently positions itself as the optimal solution strategy, achieving the shortest resolution time in 100 % of the evaluated instances while maintaining a complete success rate in reaching the prescribed optimality tolerance.
In contrast, alternative decomposition methods exhibit severe computational deterioration and a clear loss of competitiveness relative to BZ. The Level Method fails to achieve the best computational time in any of the evaluated instances and requires significantly larger time budgets to reach convergence, reflecting its sensitivity to the non-smooth structure of the Lagrangian dual function under complex operating conditions. The Dantzig–Wolfe decomposition is the least efficient approach overall, presenting the highest runtimes and the greatest variability between instances, which can be attributed to the degeneracy and tailing-off effects that commonly affect column generation methods in large-scale settings. These results conclusively demonstrate that for the California system, the proposed BZ-based algorithm represents the only robust and computation-

ally tractable alternative capable of guaranteeing convergence to optimality within competitive execution times.

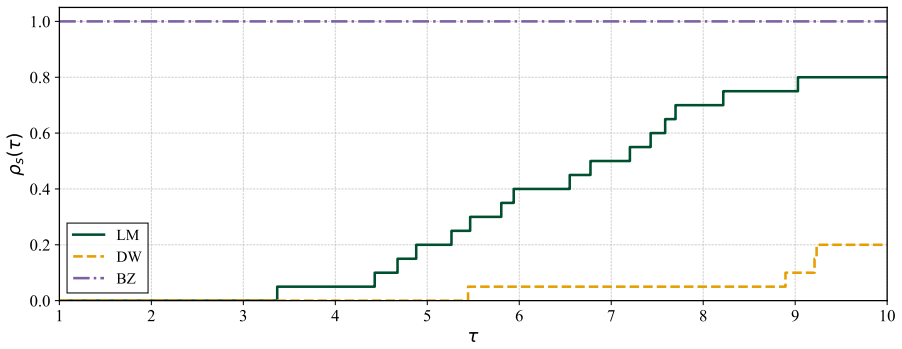


Figure 4.10: Performance profiles by approach for the California system.

The preceding analysis is further supported by the aggregated metrics reported in Table 4.5, which summarizes the average computational time, standard deviation, median, and minimum and maximum runtimes for each approach across all California system instances. The results confirm that the BZ algorithm consistently achieves lower computational times than both alternative methods by a substantial margin. Specifically, the proposed approach is approximately 7 times faster than the Level Method, which reports an average runtime of 232 s, and approximately 13 times faster than the Dantzig–Wolfe decomposition, which reaches an average of 471 s. Beyond the differences in average runtime, the standard deviation reported for the BZ algorithm is markedly lower than those of the competing approaches, indicating a more stable and predictable convergence behavior across instances of varying complexity.

Table 4.5: Approach metrics for the California system.

Metrics (s)	LM	DW	BZ
Min	270.99	509.44	**38.85**
Max	1063.81	1251.54	**94.53**
Median	324.02	745.75	**57.03**
Mean	479.43	813.18	**57.77**
σ	294.31	252.20	**12.80**

4.4.2 FERC system

For the FERC system, the comparison among the three solution approaches is illustrated with performance profiles in Figure 4.11. Performance profile analysis for the FERC system corroborates the computational superiority of the proposed BZ algorithm while simultaneously revealing an inversion in the relative performance of the two decomposition-based approaches (compared to California system). The BZ algorithm achieves the best computational time in more than 85 % of the evaluated instances. Although the LM approach, despite achieving the fastest runtime in approximately 15 % of the instances, exhibits a critical algorithmic deterioration in the remaining cases, failing to reach the prescribed optimality tolerance within the established time limit for a subset of instances. The DW decomposition, while unable to achieve the best computational time in any individual instance, demonstrates a higher degree of robustness than the LM by successfully converging across all evaluated instances, albeit at significantly higher runtimes. These results highlight an important distinction between computational speed and algorithmic reliability. Table 4.6 presents the runtime metrics per method (considering only the first eight instances for the LM). The results further confirm the robustness of the proposed algorithm, as it exhibits the lowest standard deviation among the three approaches. Regarding the convergence time, BZ shows a higher average than the Level Method; nevertheless, for the instances where the Level Method converged, BZ achieves an average time of 161 s.

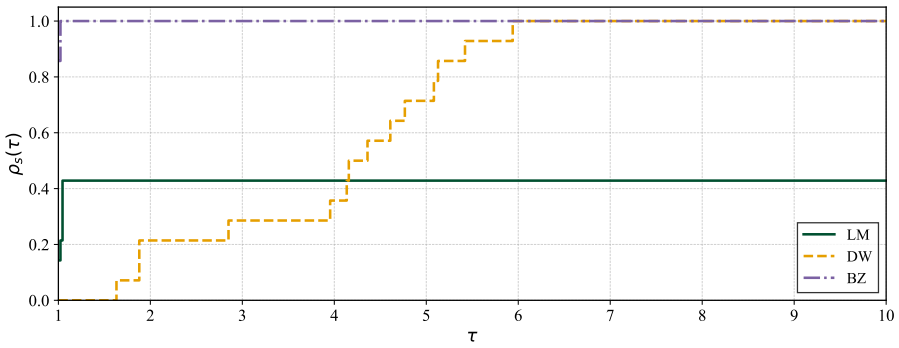


Figure 4.11: Comparison of computational times by approach for the FERC system.

Table 4.6: Approach metrics for the FERC system.

Metrics (s)	LM	DW	BZ
Min	102.72	368.47	**99.94**
Max	249.60	1162.11	**245.91**
Median	**142.27**	621.59	179.33
Mean	**163.70**	664.56	175.43
σ	60.02	234.39	**41.68**

4.4.3 RTS-GMLC system

The computational time results obtained for the different approaches in the RTS-GMLC test system are presented in Figure 4.12. The performance profile analysis for the RTS-GMLC system further reinforces the computational advantages of the proposed BZ-based algorithm. The BZ approach achieves the fastest convergence time in more than 90 % of the evaluated instances. This result is consistent with the reduced scale of the RTS-GMLC system, which comprises 73 generators, allowing the partition refinement mechanism of the BZ algorithm to operate with particular efficiency. The Level Method demonstrates a notable recovery in robustness compared to its behavior on the FERC system, successfully converging to the prescribed optimality tolerance in all instances, while achieving the best computational time in the remaining 10 % of cases. The DW decomposition, on the other hand, exhibits the weakest overall computational performance across all instances, consistently reporting the highest runtimes and confirming its limited efficiency relative to the other two approaches regardless of system size.

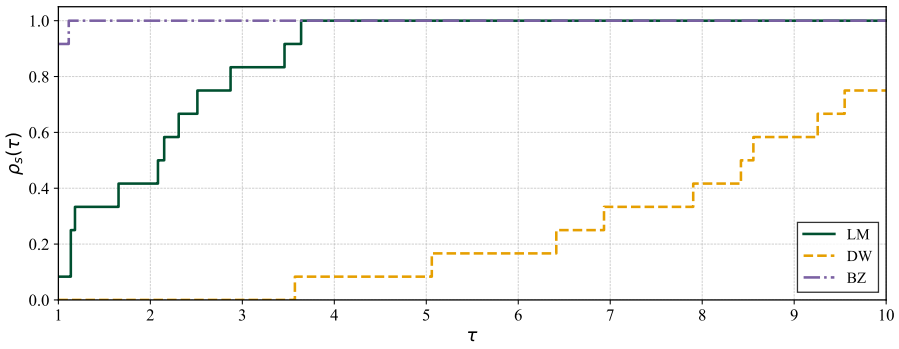


Figure 4.12: Comparison of computational times by approach for the RTS-GMLC system.

Beyond illustrating the relative robustness of BZ in more demanding instances, consistent with the average runtime and standard deviation reported in Table 4.7, these results further emphasize their practical relevance when computational efficiency is a primary objective under consistent convergence tolerances and experimental conditions.

Table 4.7: Approach metrics for the RTS-GMLC system.

Metrics	LM	DW	BZ
Min	25.62	135.59	**13.66**
Max	287.64	296.78	**83.23**
Median	34.20	193.41	**21.97**
Mean	61.82	205.23	**27.50**
σ	70.13	44.61	**17.78**

4.4.4 Belgium system

The performance profile on the Belgium system is shown in Figure 4.13, where the DW decomposition once again shows the weakest performance, whereas the BZ and LM algorithms present similar behavior and comparable computational times. This can be observed from the metrics reported in Table 4.8, where the DW decomposition reports the highest average computational time, the LM and the proposed BZ algorithm report comparable average computational times, representing the closest performance gap observed in all systems evaluated in this study. Notably, the Level Method achieves the lowest individual solution time recorded among all simulated instances in this system. However, LM exhibits a significantly higher standard deviation compared to the BZ algorithm, reflecting greater variability.

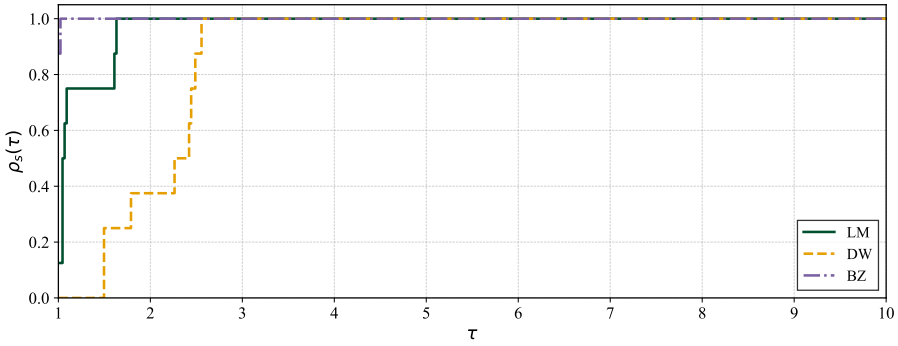


Figure 4.13: Comparison of computational times by approach for the Belgium system.

Table 4.8: Approach metrics for the Belgium system.

Metrics (s)	LM	DW	BZ
Min	**319.74**	603.41	326.14
Max	836.72	1065.6	**526.34**
Median	432.38	873.14	**404.79**
Mean	491.67	852.18	**409.70**
σ	158.69	156.81	**54.13**

4.5 Market implications of ramps in CHP problem

4.5.1 Ramps constraints in market-clearing price

The impact of ramps constraint on market-clearing prices is analyzed in two cases: (i) the network model without operational ramp constraints and (ii) the extended model including operational ramp constraints. For simplicity, the analysis is conducted only on California system instances, as the primary purpose is to demonstrate how the exclusion of these constraints directly impacts the market-clearing price.

The market clearing prices for instances 1, 12, and 20 of the California system are presented in Figure 4.14, Figure 4.15 and Figure 4.16, respectively. Three distinct instances are presented to illustrate energy prices under different system demand scenarios. The results clearly show that incorporating ramp constraints into the network-flow formulation leads to higher energy prices

compared to the formulation without ramping. This occurs because the system no longer has the same degree of freedom to adjust generator dispatch freely, but is instead constrained by the technical limitations of each generator. In particular, even during start-up, the generator output is limited by the start-up ramp constraint.

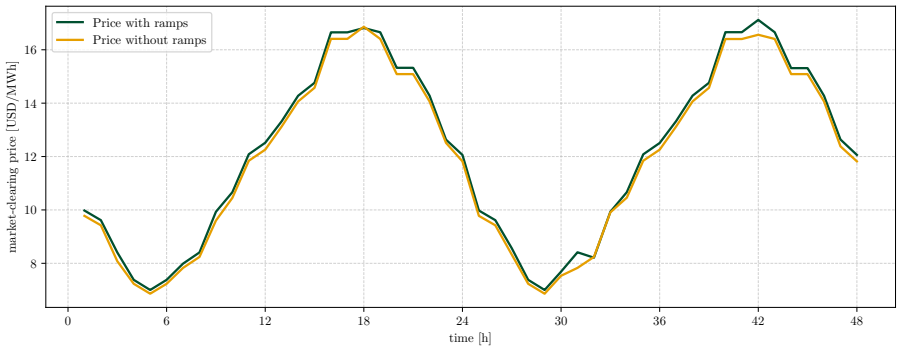


Figure 4.14: Market-clearing price for instance 1 of California system.

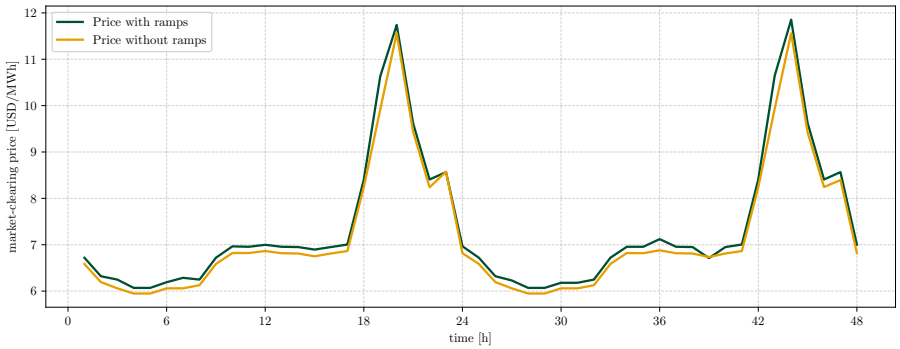


Figure 4.15: Market-clearing price for instance 12 of California system.

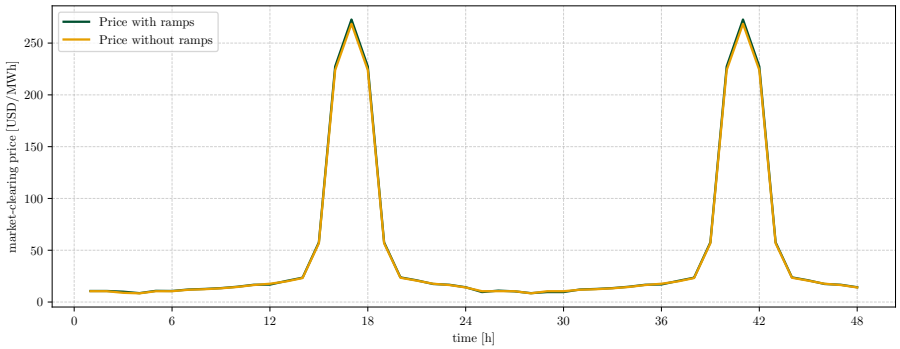


Figure 4.16: Market-clearing price for instance 20 of California system.

4.5.2 Impact of ramp constraints modeling on uplift payments

The uplift payments generated by this price difference presented in the California system are shown in Figure 4.17 in USD and in Figure 4.18 as a percentage of the total generation costs by instance. The results include uplift values for all instances of the California system. It can be observed that the instances that show the greatest uplift differences correspond to those with behavior similar to instance 2 and 3. This is explained by the fact that for such instances, the difference in the market clearing price between formulations with and without ramp constraints is larger, as presented in Figure 4.15. In contrast, instances with behavior similar to that of instances 1 or 20 exhibit smaller

differences between the cases with and without ramp constraints, indicating that in these cases the system is not as constrained by ramping limitations as in the other instances considered.

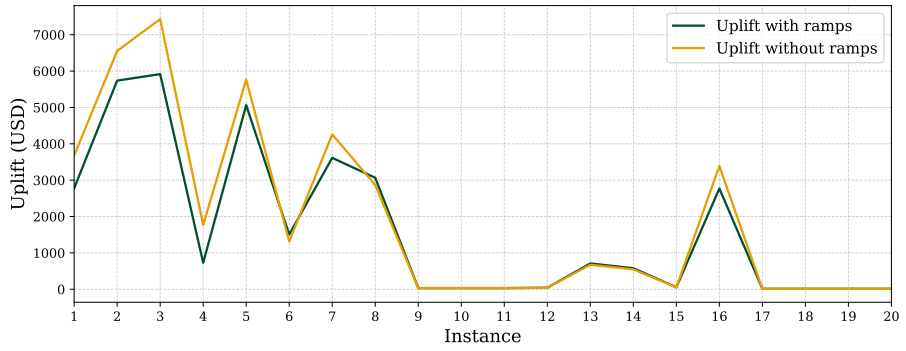


Figure 4.17: Uplift payments in USD for California instances.

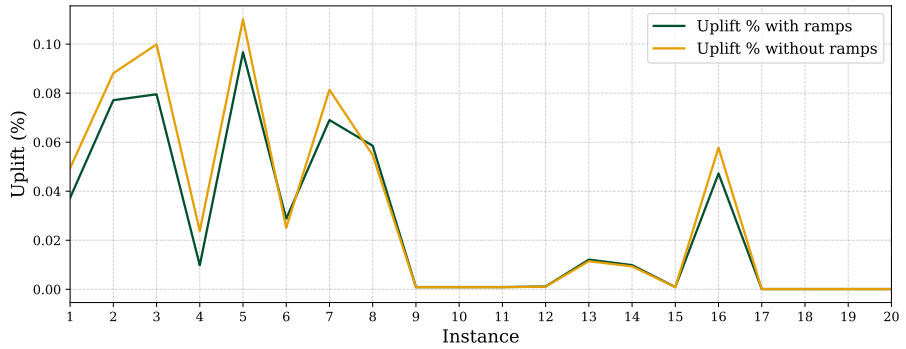


Figure 4.18: Uplift payments in percentage for California instances.

A detailed examination of the uplift payment components obtained under both modeling configurations is presented in Figure 4.19 and Figure 4.20 for the cases with and without operational ramp constraints, respectively.When ramp constraints are explicitly incorporated into the formulation, as shown in Figure 3, both uplift components are simultaneously reduced. The total uplift no longer exceeds 6000 `USD` in any instance, and the relative reduction is observed in both the MWP and LOC contributions. This simultaneous reduction in both components indicates that the CHP prices computed under the extended formulation more accurately reflect the true operational costs and physical limitations of the generating units, resulting in commitment schedules that better align individual generator optimization with the centralized dispatch solution. Consequently, generators face smaller cost recovery deficiencies and reduced opportunity cost deviations under the ramp constrained formulation.

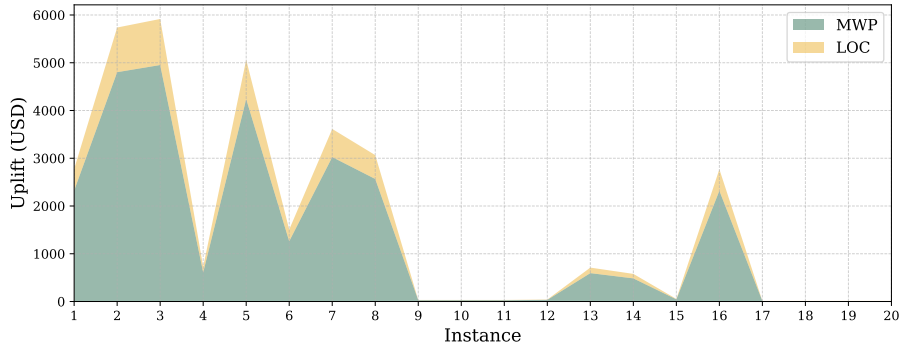


Figure 4.19: Uplift components with ramps.

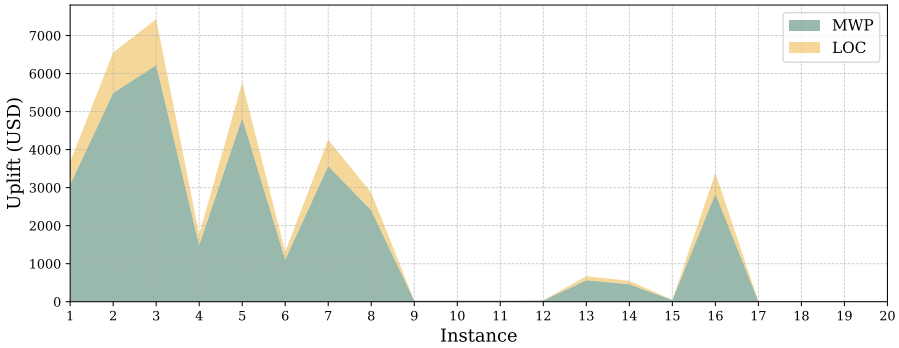


Figure 4.20: Uplift components without ramps.

Chapter 5

Conclusions and future work

The operation of modern power systems is characterized by the presence of intrinsic non-convexities in the cost structure and technical constraints of generating units, such as start-up costs, minimum up- and down-times, and minimum technical generation levels. Under conventional marginal pricing schemes, these non-convexities prevent energy prices from fully recovering operational costs, making uplift payments necessary to ensure generator cost recovery. Convex Hull Pricing addresses this issue by defining prices based on the convex envelope of the system cost function, thereby reducing side payments and improving price transparency. Although CHP reduces market distortions, its computational implementation poses significant challenges due to the difficulty of explicitly characterizing the convex hull of the UC problem.

This thesis proposes and implements a methodology to solve the CHP problem using an extended Network Flow formulation. Since operational characteristics are modeled through the node and arc structure of the network, the convex hull underlying the formulation remains invariant under modeling changes, in contrast to more compact formulations where modifications in the model require the derivation of new valid inequalities to redefine the convex hull. The formulation explicitly includes intertemporal ramping constraints and warm-up periods, which break the TU property and lead to MILP subproblems, but allow for a more accurate representation of generator flexibility. To efficiently solve this large scale model, a Bienstock–Zuckerberg–based algorithm is implemented, relying on iterative partition refinement within a column generation framework, enhanced through MILP presolve techniques and heuristic initialization.

Computational experiments conducted on the California, FERC, RTS-GMLC and Belgium systems confirm the robustness and efficiency of the proposed approach. First, the impact of including ramp constraints in the model is analyzed. The inclusion of operational ramp constraints into the network-flow formulation has shown to significantly affect the resulting market-clearing prices by restricting the dispatch flexibility available to the system operator. When ramp constraints are explicitly modeled, generators are no longer free to adjust their output arbitrarily between consecutive time periods, but are instead bound by their physical ramping capabilities, which reduces the set of feasible dispatch trajectories and tightens the convex hull of the feasible operating region. As a consequence, the system must rely on more expensive generation alternatives to meet demand under these restricted conditions, leading to systematically higher market-clearing prices. In contrast, the formulation without ramps constraints artificially lowers prices, resulting in insufficient revenues to cover generator operating costs (considering the a real operation, that is, including operational ramps).

From a computational perspective, the Bienstock–Zuckerberg–based algorithm consistently outperforms conventional decomposition techniques such as the Level Method and Dantzig–Wolfe decomposition. In particular, the proposed methodology achieves shorter convergence times and, more importantly, significantly lower performance variability across instances of differing complexity, confirming its suitability for real market applications. Additionally, the critical role of the MIP solver presolve phase is quantified, showing that disabling presolve increases computational times by an average of 65%, highlighting the importance of reducing the search space in large-scale combinatorial problems through the elimination of redundant variables and constraints.

Nevertheless, the proposed methodology is not without limitations, because the current formulation focuses on a detailed network-flow representation of generators and assumes a simplified transmission network (single node model), ignoring grid congestion effects that are essential for lo-

cational pricing. Furthermore, the growth of the network graph with the time horizon may impact memory usage in long-term simulations, although this issue is partially mitigated by the partition refinement strategy of the algorithm, which rapidly restricts the feasible solution space at each iteration.

5.1 Future work

The work developed in this thesis opens several promising research directions that extend from the contributions presented.

One relevant line of future research consists of incorporating the proposed network-flow formulation and the Convex Hull Pricing problem into a hydrothermal coordination framework. This extension is particularly relevant for power systems with significant hydroelectric participation, such as the Chilean electricity system, where projected water value plays a central role in the strategic planning of reservoir management and generation scheduling. In such systems, the decision of when and how much water to use for power generation involves complex intertemporal trade-offs, since water stored today represents a valuable resource that can be used to displace more expensive thermal generation in future periods. By extending the proposed CHP formulation to explicitly incorporate the hydraulic constraints of cascaded reservoir systems (including water balance equations, storage limits, and turbine operating ranges) it would be possible to derive more accurate and economically consistent water values within the convex hull pricing framework.

A second relevant extension of the proposed methodology consists in incorporating stochasticity into the CHP problem formulation by explicitly considering uncertainty in both demand and renewable generation through a scenario-based representation. Under a stochastic formulation, multiple demand and generation scenarios would be simultaneously considered within the optimization problem, requiring the market-clearing prices to be determined in a manner that is consistent across all scenarios. As a direct consequence, the number of variables and constraints in the resulting problem would grow proportionally with the number of scenarios considered. This increased complexity would provide a natural and more demanding benchmarking setting in which to compare the proposed BZ algorithm with the LM and DW decomposition, allowing a more rigorous assessment of the computational advantages of the proposed approach under large scale stochastic configurations and diverse operating conditions.

Bibliography

[1] Guoling Peng, Rongzhang Cao, Yantao Zhang, Mengfu Tu, and Li Chang. Analysis of convex hull pricing in electricity markets. In *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, pages 2024–2029, 2021.

[2] Generadoras de Chile. Boletín marzo 2024 generadoras de chile, 2024.

[3] Sauer and William. Uplift in rto and iso markets. *Federal Energy Regulatory Comission, Tech. Rep*, 2014.

[4] William Hogan and Brendan J Ring. On minimum-uplift pricing for electricity markets., 2003.

[5] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.

[6] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988.

[7] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific Optimization and Computation Series. Athena Scientific, Belmont, MA, 2 edition, 1999.

[8] Shiang-Tai Liu and Chiang Kao. Network flow problems with fuzzy arc lengths. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(1):765–769, Feb 2004.

[9] Laurence A. Wolsey. *Integer Programming*. John Wiley & Sons, New York, 1998.

[10] Monique Guignard. Lagrangean relaxation. *Top*, 11(2):151–200, 2003.

[11] Marco E Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations research*, 53(6):1007–1023, 2005.

[12] François Vanderbeck and Laurence A Wolsey. Reformulation and decomposition of integer programs. In *50 years of integer programming 1958-2008: from the early years to the state-of-the-art*, pages 431–502. Springer, 2009.

[13] Hatem MT Ben Amor, Jacques Desrosiers, and Antonio Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009.

[14] Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3):316–329, 1998.

[15] Jacek Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

[16] Jacek Gondzio and Pablo González-Brevis. A new warmstarting strategy for the primal-dual column generation method. *Mathematical Programming*, 152(1):113–146, 2015.

[17] Daniel Bienstock and Mark Zuckerberg. A new lp algorithm for precedence constrained production scheduling. *Optimization Online*, pages 1–33, 2009.

[18] Gonzalo Muñoz, Daniel Espinoza, Marcos Goycoolea, Eduardo Moreno, Maurice Queyranne, and Orlando Rivera. A study of the bienstock-zuckerberg algorithm, applications in mining and resource constrained project scheduling. *arXiv preprint arXiv:1607.01104*, 2016.

[19] Qipeng P. Zheng, Jianhui Wang, and Andrew L. Liu. Stochastic optimization for unit commitment—a review. *IEEE Transactions on Power Systems*, 30(4):1913–1924, 2015.

[20] N.P. Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, 2004.

[21] M. Carrion and J.M. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, 2006.

[22] Gustavo Morales-España, Claudio Gentile, and Andrés Ramos. Tight MIP formulations of the power-based unit commitment problem. *OR Spectrum*, 35(4):929–950, 2013.

[23] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. Gurobi Optimization, LLC, 2023. Retrieved from `https://www.gurobi.com/documentation/`.

[24] IBM ILOG CPLEX Optimization Studio. *CPLEX User's Manual*. IBM, 2023. Retrieved from `https://www.ibm.com/docs/en/icos/latest`.

[25] James Ostrowski, Miguel F. Anjos, and Anthony Vannelli. Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Transactions on Power Systems*, 27(1):39–46, 2012.

[26] Germán Morales-España, Jesus M. Latorre, and Andres Ramos. Tight and compact milp formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 28(4):4897–4908, 2013.

[27] Leigh Tesfatsion. Locational marginal pricing: A fundamental reconsideration. *IEEE Open Access Journal of Power and Energy*, 11:104–116, 2024.

[28] California ISO Department of Market Monitoring. 2023 annual report on market issues and performance. Technical report, California Independent System Operator (CAISO), Folsom, CA, July 2024. Section on bid cost recovery: statement on insufficiency of net revenues from LMP to cover start-up and minimum load costs.

[29] Martin Bichler, Johannes Knörr, and Felipe Maldonado. Pricing in nonconvex markets: How to price electricity in the presence of demand response. *Information Systems Research*, 34(2):652–675, 2023.

[30] Richard P O'Neill, Paul M Sotkiewicz, and Michael H Rothkopf. Equilibrium prices in power exchanges with non-convex bids, 2007.

[31] PJM Interconnection. Pjm manual 28: Operating agreement accounting, version 88 – section 5.2.6: Wind lost opportunity cost. Technical Report Manual 28, v88, PJM Interconnection, Valley Forge, PA, October 2022.

[32] George Liberopoulos and Panagiotis Andrianesis. Critical review of pricing schemes in markets with non-convex costs. *Operations Research*, 64(1):17–31, 2016.

[33] Mathieu Van Vyve et al. Linear prices for non-convex electricity markets: models and algorithms. Technical report, CORE Leuven, Belgium, 2011.

[34] Dane A. Schiro, Tongxin Zheng, Feng Zhao, and Eugene Litvinov. Convex hull pricing in electricity markets: Formulation, analysis, and implementation challenges. *IEEE Transactions on Power Systems*, 31(5):4068–4075, 2016.

[35] Yanan Yu, Yongpei Guan, and Yonghong Chen. An extended integral unit commitment formulation and an iterative algorithm for convex hull pricing. *IEEE Transactions on Power Systems*, 35(6):4335–4346, 2020.

[36] Paul R. Gribik, William W. Hogan, and Susan L. Pope. Market-clearing electricity prices and energy uplift. Technical report, Harvard Electricity Policy Group, Harvard Kennedy School of Government, Cambridge, MA, December 2007.

[37] Congcong Wang, Tengshun Peng, Peter B. Luh, Paul Gribik, and Li Zhang. The subgradient simplex cutting plane method for extended locational marginal prices. *IEEE Transactions on Power Systems*, 28(3):2758–2767, 2013.

[38] Nicolas Stevens, Yurii Nesterov, and Yves Smeers. Models and algorithms for pricing electricity in unit commitment. *URL https://ap-rg. eu/wp-content/uploads/2020/06/MasterStevens. pdf*, 2016.

[39] Gui Wang, Uday V. Shanbhag, Tongxin Zheng, Eugene Litvinov, and Sean Meyn. An extreme-point subdifferential method for convex hull pricing in energy and reserve markets—part i: Algorithm structure. *IEEE Transactions on Power Systems*, 28(3):2111–2120, 2013.

[40] Nicolas Stevens and Anthony Papavasiliou. Application of the level method for computing locational convex hull prices. *IEEE Transactions on Power Systems*, 37(5):3958–3968, 2022.

[41] Bowen Hua and Ross Baldick. A convex primal formulation for convex hull pricing. *IEEE Transactions on Power Systems*, 32(5):3814–3823, 2017.

[42] Deepak Rajan, Samer Takriti, et al. Minimum up/down polytopes of the unit commitment problem with start-up costs. *IBM Res. Rep*, 23628:1–14, 2005.

[43] Kai Pan and Yongpei Guan. A polyhedral study of the integrated minimum-up/-down time and ramping polytope. *arXiv preprint arXiv:1604.02184*, 2016.

[44] Cristian Álvarez, Fernando Mancilla-David, Pablo Escalona, and Alejandro Angulo. A bienstock–zuckerberg-based algorithm for solving a network-flow formulation of the convex hull pricing problem. *IEEE Transactions on Power Systems*, 35(3):2108–2119, 2020.

[45] Panagiotis Andrianesis, Dimitris Bertsimas, Michael C. Caramanis, and William W. Hogan. Computation of convex hull prices in electricity markets with non-convexities using dantzig-wolfe decomposition. *IEEE Transactions on Power Systems*, 37(4):2578–2589, 2022.

[46] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[47] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.

Appendix A

First Appendix

A.1 Details of computational times per system

This section presents a detailed summary of the computational times obtained for each of the three solution approaches (Level Method, Dantzig–Wolfe decomposition, and the proposed Bienstock–Zuckerberg) across all systems and simulated instances considered in this study. The results are organized by system and reported in individual tables, allowing a direct instance-by-instance comparison of the convergence times achieved by each methodology under a uniform optimality gap tolerance of 0.05%. These tables complement the performance profile analyses presented in the preceding sections by providing the exact numerical values upon which the statistical metrics and comparative conclusions are based, and serve as a comprehensive reference for assessing the computational efficiency and robustness of each approach across the full set of evaluated configurations.

A.1.1 California system

Table A.1 reports the computational times achieved by each of the three solution approaches for every individual instance of the California system. The results confirm the computational superiority of the proposed BZ-based algorithm, which consistently achieves the lowest solution time across all evaluated instances without exception.

Table A.1: Computational times per instance for the California system.

instance	LM	DW	BZ
1	290.32	509.44	**55.28**
2	287.96	558.92	**38.85**
3	292.78	656.12	**49.30**
4	290.53	530.58	**59.76**
5	454.84	730.72	**67.27**
6	456.65	790.13	**60.31**
7	458.54	760.77	**59.68**
8	454.53	668.68	**50.38**
9	324.05	1251.54	**45.05**
10	323.99	1169.64	**39.43**
11	321.11	1218.25	**49.13**
12	324.39	1172.3	**56.09**
13	277.60	1001.77	**82.59**
14	270.99	982.52	**57.97**
15	272.32	975.62	**50.00**
16	272.70	996.58	**61.70**
17	1039.19	548.71	**53.22**
18	1052.71	512.83	**94.53**
19	1059.56	658.50	**62.84**
20	1063.81	570.06	**62.02**

A.1.2 FERC system

Table A.2 reports the computational times achieved by each solution approach for every individual instance of the FERC system. For the first six instances, the BZ algorithm and the Level Method exhibit broadly comparable performance, reporting average computational times of 160 `s` and 163 `s`, respectively, indicating that both approaches are similarly competitive under the operating conditions. However, a critical divergence emerges from instance seven onward, where the Level Method fails to reach the prescribed optimality gap tolerance of 0.05 `%` within the established time limit across the remaining eight instances. This convergence failure can be attributed to the fundamental characteristics of the Lagrangian dual based approach underlying the Level Method. This behavior demonstrates a significant lack of robustness in the Level Method when confronted with certain system configurations, representing a critical limitation for its practical application in large-scale electricity markets.

Table A.2: Computational times per instance for the FERC system.

instance	LM	DW	BZ
1	140.79	587.79	**135.31**
2	**143.76**	684.12	145.53
3	241.34	437.94	**233.51**
4	249.60	399.40	**245.91**
5	102.72	506.40	**99.94**
6	**104.01**	532.05	105.63
7	-	463.38	**163.03**
8	-	368.47	**197.12**
9	-	1162.11	**195.71**
10	-	996.27	**183.78**
11	-	843.66	**204.44**
12	-	942.31	**205.05**
13	-	655.38	**166.16**
14	-	724.57	**174.88**

A.1.3 RTS-GMLC system

The computational times for the RTS-GMLC system are reported in Table A.3. Similarly to the results observed for previous systems, the BZ algorithm and the LM exhibit broadly comparable performance across the majority of instances. However, a exception arises in instance 11, where the LM approach experiences a significant convergence runtime. While this behavior does not result in a complete convergence failure as observed in the FERC system, it nonetheless highlights the sensitivity of the LM to particular operation conditions. The DW decomposition, on the other hand, exhibits consistently poor computational performance across all instances of this system, reporting runtimes approximately three times higher than those achieved by both the BZ algorithm and the LM approach.

Table A.3: Computational times per instance for the RTS-GMLC system.

instance	LM	DW	BZ
1	46.15	185.35	**20.07**
2	53.76	223.88	**14.84**
3	32.44	164.58	**15.6**
4	25.62	184.05	**21.91**
5	30.27	135.59	**26.85**
6	29.23	250.54	**13.66**
7	31.05	238.33	**27.86**
8	**29.35**	187.86	32.12
9	35.96	152.48	**22.03**
10	51.97	198.96	**20.87**
11	287.64	296.78	**83.23**
12	88.44	244.35	**30.99**

A.1.4 Belgium system

Finally, Table A.4 reports the instance-level results for the Belgium system. The results validate the proposed BZ algorithm, which proves to be competitive with the Level Method and even achieves superior computational times across several instances. Furthermore, the consistently poor performance of the Dantzig–Wolfe decomposition is once again corroborated, as this approach failed to rank as the best-performing method in any system or instance evaluated throughout this study, systematically reporting the highest computational times across all configurations.

Table A.4: Computational times per instance for the Belgium system.

instance	LM	DW	BZ
1	836.72	785.59	**526.34**
2	653.72	603.41	**404.28**
3	398.10	668.44	**374.21**
4	468.23	1065.6	**430.77**
5	**319.74**	772.65	326.14
6	420.00	985.63	**405.30**
7	392.04	975.43	**382.24**
8	444.77	960.68	**428.31**