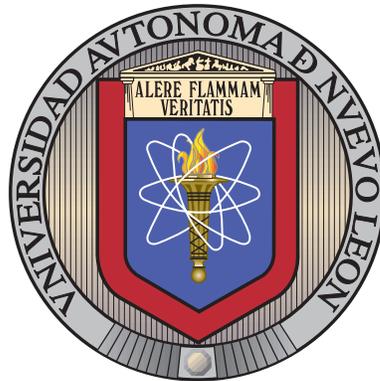


UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



EFFICIENT METHODS FOR SOLVING POWER SYSTEM  
OPERATION SCHEDULING CHALLENGES: THE THERMAL  
UNIT COMMITMENT PROBLEM WITH STAIRCASE COST  
AND THE VERY SHORT-TERM LOAD FORECASTING  
PROBLEM

POR

URIEL IRAM LEZAMA LOPE

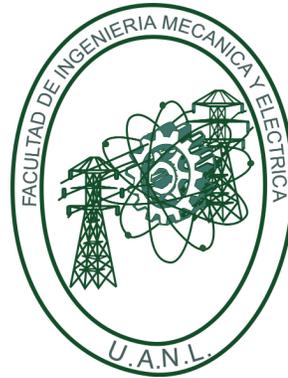
COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE  
DOCTORADO EN INGENIERÍA DE SISTEMAS

AGOSTO 2023

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



RO EFFICIENT METHODS FOR SOLVING POWER SYSTEM  
OPERATION SCHEDULING CHALLENGES: THE THERMAL  
UNIT COMMITMENT PROBLEM WITH STAIRCASE COST  
AND THE VERY SHORT-TERM LOAD FORECASTING  
PROBLEM

POR

URIEL IRAM LEZAMA LOPE

COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE  
DOCTORADO EN INGENIERÍA DE SISTEMAS

AGOSTO 2023

**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN**  
**Facultad de Ingeniería Mecánica y Eléctrica**  
**Posgrado**

Los miembros del Comité de Evaluación de Tesis recomendamos que la Tesis "Efficient Methods for Solving Power System Operation Scheduling Challenges: The Thermal Unit Commitment Problem with Staircase Cost and the Very Short-term Load Forecasting Problem", realizada por el estudiante Uriel Iram Lezama Lope, con número de matrícula 2032424, sea aceptada para su defensa como requisito parcial para obtener el grado de Doctorado en Ingeniería de Sistemas.

**El Comité de Evaluación de Tesis**

Dr. Roger Zirahuén Ríos Mercado  
Director

Dra. Ada Margarita Álvarez Socarrás  
Revisor

Dr. José Luis Ceciliano Meza  
Revisor

Dra. Iris Abril Martínez Salazar  
Revisor

Dra. María Angélica Salazar Aguilar  
Revisor

Dr. Omar Jorge Ibarra Rojas  
Revisor

Vo.Bo.

  
Dr. Simón Martínez Martínez  
Subdirector de Estudios de Posgrado

Institución 190001

Programa 557620

Acta Núm. 329

Ciudad Universitaria, a 30 de agosto 2023.

*A mi amada esposa Diana,  
a mi querida madre Ana Marta,  
y a mi adorado hijo Fernando.*

# CONTENTS

---

<b>Acknowledgments</b>	<b>ix</b>
<b>Abstract</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>5</b>
2.1 Power systems components . . . . .	5
2.2 The Unit Commitment Problem . . . . .	6
2.2.1 Generators constraints . . . . .	7
2.2.2 System constraints . . . . .	9
2.3 The Load Forecasting . . . . .	9
<b>3 Hypothesis and objectives</b>	<b>12</b>
3.1 Hypothesis . . . . .	12
3.2 Research objectives . . . . .	12
<b>4 Literature review</b>	<b>14</b>
4.1 Literature survey on UCP models . . . . .	14

---

4.2	Matheuristics works in UCP . . . . .	16
4.3	Load forecasting works . . . . .	22
4.4	Summary of thesis scientific contributions . . . . .	25
<b>5</b>	<b>A tight and compact thermal unit commitment model</b>	<b>27</b>
5.1	The thermal T&C UCP proposed model . . . . .	29
5.1.1	Notation . . . . .	29
5.1.2	Mathematical formulation . . . . .	32
5.1.3	Model overview . . . . .	35
5.2	Validation of the accuracy of the proposed model . . . . .	37
5.3	Results and discussion . . . . .	38
5.4	Conclusions . . . . .	38
<b>6</b>	<b>Matheuristic approach to solve the thermal UCP</b>	<b>41</b>
6.1	Matheuristics methods . . . . .	43
6.1.1	Construction phase . . . . .	43
6.1.2	Improving phase . . . . .	45
6.1.3	Local branching . . . . .	45
6.1.4	Kernel kearch . . . . .	52
6.1.5	Summary of proposed matheuristics . . . . .	55
6.2	Experimental work . . . . .	56
6.2.1	Description of test instances . . . . .	57
6.2.2	Test 1: Assessment of constructive methods . . . . .	59

---

6.2.3	Test 2: Assessment of improvement methods under a running time limit of 4000 seconds . . . . .	65
6.2.4	Test 3: Assessment of improvement methods under a running time limit of 7200 seconds . . . . .	69
6.3	Discussion . . . . .	71
6.4	Conclusions . . . . .	81
6.5	Future research . . . . .	83
<b>7</b>	<b>A real-time load forecasting method</b>	<b>84</b>
7.1	Real-time forecasting . . . . .	85
7.2	The Analogue Moving Average method . . . . .	87
7.2.1	AnMA variants . . . . .	91
7.3	Experimental work . . . . .	92
7.3.1	Dataset . . . . .	94
7.3.2	Parameters . . . . .	95
7.3.3	Real-time demand forecasting tests results . . . . .	96
7.4	Discussion . . . . .	103
7.5	Conclusions . . . . .	104
7.6	Future research . . . . .	105
<b>8</b>	<b>Main conclusions</b>	<b>107</b>
8.1	Summary of main findings . . . . .	107
8.2	Future research . . . . .	110

---

<b>A</b>	<b>Description of instances</b>	<b>112</b>
<b>B</b>	<b>Results of comparison among solution algorithms</b>	<b>115</b>
B.1	Results on instances from group x7day_small . . . . .	115
B.2	Results on instances from group x7day_medium . . . . .	119
B.3	Results on instances from group x7day_large . . . . .	125
<b>C</b>	<b>Statistical results</b>	<b>131</b>
<b>D</b>	<b>Load demand forecasting results</b>	<b>141</b>

# ACKNOWLEDGMENTS

---

I want to acknowledge the following institutions for their invaluable support during my doctoral studies: FIME (Department of Mechanical and Electrical Engineering) at the Universidad Autónoma de Nuevo León (UANL) and the Graduate Program in Systems Engineering (PISIS) of UANL. Their financial and academic assistance was pivotal in completing this research.

I am grateful to CONACYT (Mexican National Council for Science and Technology) for the financial support from a graduate scholarship enabling me to carry out this project.

I am also grateful to Instituto Nacional de Electricidad y Energías Limpias (INEEL) through FICYDET (the Fund for Scientific Research and Technological Development) for their financial support during this research.

I sincerely thank my supervisor, Professor Roger Z. Ríos Mercado, for his wise guidance and unwavering support throughout this academic journey. Without his invaluable assistance, the accomplishment of this project would not have been possible.

I would also like to acknowledge the advice and insights provided by Armando de la Torre Sánchez, Diana Lucía Huerta Muñoz, José Luis Ceciliano Meza, Miguel Ceceñas Falcón, and Víctor Manuel Cázares Lira, which significantly contributed to the development of this work. Your support has been invaluable to me, and I am truly grateful. Thank you sincerely.

Furthermore, I would like to express my heartfelt appreciation to the reviewers for their valuable comments, questions, and insights, which have enhanced the quality of this thesis.

Finally, I would like to extend my heartfelt thanks to my fellow university colleagues and friends Alan, Alberto, Erick, Johana, Mayra, Pablo, and Citlali whose loyalty and encouraging words have made this journey enjoyable.

# ABSTRACT

---

This dissertation addresses two crucial power system operational scheduling aspects: the thermal unit commitment problem and real-time load demand forecasting. These two are important challenges that need the development of models and efficient algorithms.

To tackle the generator scheduling problem, we propose and develop five matheuristic methods, including four variations of local branching (LB) and one using kernel search (KS). Additionally, we have introduced a novel constructive approach named HARDUC, which effectively generates high-quality initial solutions for the matheuristic methods.

To assess the effectiveness of the proposed matheuristic algorithms, tests were conducted on instances simulating scenarios where an analyst must deliver a generation schedule within one or two hours. A comparison with CPLEX, an off-the-shelf solver, under two scenarios (i) using the solver from scratch and (ii) using the solver with an initial feasible solution from the heuristic, revealed interesting insights. For small instances, the off-the-shelf solver outperformed the matheuristic methods. However, in medium-sized instances, the solver sometimes struggled to find a feasible solution. When solutions were found, there were no significant differences in performance between the solver and the methods. However, the proposed methods excelled and achieved remarkable results for large instances where the solver left many instances unsolved.

Furthermore, we tested the proposed constructive method by comparing it with the best UCP constructive method from the literature. The results, supported by statistical tests, indicate the superiority of our proposed method. Our implementation of the KS algorithm outperformed both the solver and the LB method in terms of relative optimality gap, especially in challenging instances. This discovery is significant as it reveals tremendous

potential in utilizing KS for solving the UCP, paving the way for further research.

As expected, as complexity increased, matheuristic methods outperformed the solver, delivering quicker, more effective solutions. Matheuristics have a proven track record and are incorporated into commercial solvers for efficient solutions in mixed-integer linear programming problems. In our research, we customized matheuristics for the thermal UCP by identifying dominant variables and addressing implementation challenges of the KS method, improving its efficacy.

In addition, we introduced a novel method called the Analogue with Moving Average (AnMA) approach in very short-term load demand forecasting. AnMA exploits the seasonal characteristics of load demand time series by selecting the most correlated days. Its adaptability to real-time data positions it as an ideal choice for accommodating new demand patterns, rectifying biases, and enhancing accuracy. AnMA was compared against other methods recognized for their efficiency and precision in the literature. The results showcased AnMA's superiority, outperforming naive algorithms and exponential smoothing methods in accuracy, computational speed, and cost-effectiveness. Additionally, AnMA achieved comparable accuracy to ARIMA models while requiring significantly fewer computational resources and less time.

Our research addresses critical challenges in power system operational scheduling, highlighting the effectiveness of tailored methods that align with problem-specific characteristics. These findings hold practical significance for the electricity industry, as our approach leverages a profound understanding of problem characteristics to improve operational scheduling. The success of our methods could potentially guide the development of future hybrid heuristic approaches combining mixed-integer programming or matheuristics alongside analogy-based forecasting techniques, offering substantial practical advancements in the electricity sector.



---

**Roger Z. Ríos Mercado, Ph.D.**

**Director**

# INTRODUCTION

---

Electricity is critical in modern society, as numerous activities rely on it. However, power systems have limited storage capacity, which means that electricity must be generated simultaneously with its consumption. Therefore, reliable and precise scheduling of electrical generation capacity becomes necessary to ensure the electricity demand's fulfillment while upholding the electrical system's stability and safety. A mismatch between the available electricity generation capacity and the demand can result in blackouts or other electrical failures, significantly impacting public service and economic productivity. Hence, it is of utmost importance to allocate the appropriate amount of electrical generation capacity for each hour to meet the demand and maintain the stability and safety of the electrical system.

Independent System Operators (ISO) are dedicated agencies entrusted with the vital responsibility of ensuring the safe and dependable operation of the electric power system. Among their primary tasks is the management of electricity flow and the crucial role of balancing supply and demand. ISOs tackle the complex challenge of solving the unit commitment problem (UCP) to achieve effective scheduling in day-ahead markets. Efficient and optimal scheduling yields notable economic benefits.

The UCP is a family of challenging and complex mathematical optimization problems in power systems that play a crucial role in power system optimization, particularly in operation scheduling. This task entails making decisions regarding generator turn-on and turn-off and determining the optimal power generation level to cost-effectively meet

consumer demand while satisfying various operational and technical constraints. The utilization of UCP models is widespread in different countries and markets for power production planning. Each country or market customizes the UCP to align with specific energy policies, incorporating various constraints and objectives.

Its highly non-linear and non-convex nature characterizes the UCP. Due to its complexity, the UCP requires significant computational resources to solve. As a result, several strategies have been proposed to address the issue of computational demands. One such strategy is increasing computational power using high-performance or parallel computing techniques. Another strategy is implementing decomposition methods, which break down the problem into smaller, more manageable sub-problems, allowing for easier solution methods. Additionally, researchers have worked to improve the mathematical formulation of the UCP to make it more tractable.

Despite significant advancements in the field of UCPs, the time required to solve the UCP remains a critical limitation on runtime. To address this limitation, researchers continue to develop and refine algorithms and techniques capable of solving the UCP more efficiently. The study of UCP is a dynamic and rapidly evolving area of research, as new challenges and complexities arise with integrating renewable energy sources and implementing new energy policies.

This research aims to improve the efficiency and profitability of the energy market by reducing the execution time of the UCP. A family of hybrid-heuristic methods, known as matheuristics, have been tested to address the problems at hand.

Matheuristics are a class of optimization algorithms that integrate mathematical programming and metaheuristic techniques to tackle complex problems. By blending traditional mathematical programming approaches, such as integer programming, with the strengths of metaheuristics, “matheuristics” can solve large-scale optimization problems more efficiently and quickly. As a result, they often provide high-quality solutions that are close to optimal. Matheuristics allows for an intelligent search of the solution space guided by the problem’s structure and knowledge, incorporating the advantages of metaheuristics, such as local search, diversification, or tabu search.

Another critical aspect this work addresses is load demand forecasting, a fundamen-

tal element in electricity planning and operation. Load demand refers to the quantity of electricity consumers require at any given time and is subject to fluctuations influenced by human behavior. Inadequate consideration of load demand can have significant cost implications. Hence, ensuring the accuracy of load demand forecasts is of utmost importance. Given that load demand is not constant, precise forecasting becomes crucial to guarantee a reliable and cost-effective electricity supply. Accurate forecasting facilitates efficient, timely scheduling and dispatching of electricity generators, minimizing the need for costly last-minute electricity purchases from the spot market.

Load demand forecasting is important to prevent the under-generation and over-generation of electricity, leading to higher costs until blackouts or other disruptions in the electricity supply. ISO uses advanced forecasting techniques that consider historical data, weather patterns, and other factors to predict load demand accurately and maintain a reliable and cost-effective electricity supply.

This research proposes a statistical learning-based method called Analogue (An) with error correction by moving averages (MA) to address demand forecasting in the short and very short term. The method uses a neighborhood search to find similar days, applies a regression model, and finally makes a real-time error correction with a MA model. This approach has the potential to be utilized in the Mexican real-time market.

Real-time load forecasting and the UCP are closely related to power system operations. Real-time load forecasting is used to update the inputs of the UCP, allowing power system operators to adjust the commitment and dispatch of generators in response to changes in demand. By incorporating real-time load forecasts into the UCP, power system operators can reduce the cost of electricity, improve stability and reliability, and better manage the supply and demand of electricity.

This dissertation is organized as follows. In Chapter 2, We provide an overview of key concepts and methodologies used to address the problems. This chapter includes fundamental ideas of electrical power systems, unit commitment, time series load forecasting, and commonly used solution methods.

Chapter 3 presents the hypotheses and objectives of this research.

In Chapter 4, a compilation of the most recent works related to thermal unit commitment formulations for the UCP models is discussed. Moreover, the main matheuristic methods used to solve the UCPs are also presented and compiled. Finally, works related to real-time load forecasting are discussed, emphasizing those most similar to our proposed forecasting method.

In Chapter 5, a new and improved model for the thermal unit commitment problem is presented. This model is designed based on the latest research and includes all the essential elements of the thermal UCP in a more efficient way. The objective is to provide a robust benchmark for testing the matheuristic methods proposed in our work.

In Chapter 6, five matheuristic methods using local branching and kernel search techniques are introduced and developed. Four of these methods are based on local branching, while another focuses on kernel search. In addition, this chapter includes an empirical assessment of the proposed methods, comparing their performance against an off-the-shelf state-of-the-art general-purpose method. The evaluation involves solving a tight and compact model with a few variables and constraints.

Chapter 7 presents a new demand forecasting method built with statistical and machine learning tools to calculate the load forecast for a real-time market.

Finally, Chapter 8 presents the conclusions of this work, including a discussion of the relevance and applicability of the methods and possible extensions. New perspectives for future research lines are outlined.

# BACKGROUND

---

In this chapter, a concise overview of the essential ideas, terminology, and methodologies used throughout this dissertation will be provided. The primary components of electrical power systems will be introduced, and their interrelationships will be defined. Additionally, the fundamental concepts of the UCP, its key constraints, and the context of time series forecasting in electrical systems, will be outlined. The solution methods used for both the UCP and real-time forecasting will also be briefly explained.

## 2.1 POWER SYSTEMS COMPONENTS

The power system is a complex network of interconnected electrical devices that provide energy to consumers through generation, transmission, and distribution. The main components of the power system include generators, buses/nodes, electrical loads, transmission lines, and tie-lines/flow-gates [78].

Generators are electrical machines that transform mechanical energy from various primary energy sources into electrical power supplied to the system. The different types of generators include thermal, hydro, and renewable generators such as wind and solar farms and geothermal plants.

Buses/nodes are points of connection between generators, lines, and loads located in power substations. Electrical loads are the elements that consume power energy in the system and are connected to the buses. They include retail consumers such as domestic,

commercial, and small industries and wholesale consumers such as factories, arc furnace, and refineries.

Transmission lines are part of the basic infrastructure comprising towers and specialized cables, which transmit energy from the generators to the loads. Tie-lines/flow-gates are groups of transmission lines that have capacity limits.

## 2.2 THE UNIT COMMITMENT PROBLEM

The UCP is a family of mathematical optimization problems that revolve around finding the optimal power production schedule for each generating unit. The aim is to meet the projected loads cost-effectively [7, 44]. The primary objective is to minimize the total cost of power generation, encompassing factors such as fuel expenses and start-up costs, all while fulfilling the technical and operational constraints associated with the power system.

The UCP involves determining which generators to turn-on and turn-off during a given period and how much power they should produce to meet the system demand while taking into account various constraints such as minimum/maximum power output levels, ramp rates, variable start-up costs, and minimum up/down times for each generator.

Various optimization techniques have been proposed for solving different versions of the UCP, including lagrangian relaxation, priority list, and more advanced methods such as mixed-integer linear programming, dynamic programming, and metaheuristic algorithms. In their respective surveys, Montero et al. [62] comprehensively cover various methodologies, techniques, and resolution strategies for addressing the Unit Commitment Problem (UCP). Meanwhile, Abdou and Tkiouat [1] provide insights into the historical evolution of the UCP and its solution methods, highlighting the progress made in addressing its challenges.

### 2.2.1 GENERATORS CONSTRAINTS

Several technical features of generators pose constraints on the optimal scheduling of power generation. These features include:

- **Generation limits:** These are the maximum and minimum levels of power energy a generator can produce while operating in a stable fashion. These limits are imposed by the generator's design and reflect its technical capabilities. The generator must be scheduled in a way that stays within these limits while meeting the expected demand for electricity.
- **Minimum up/down times:** These are the minimum periods of time that a generator must be kept on or off before it can be turned off or on again. This constraint ensures generators do not experience excessive wear and tear from frequent starting and stopping. If a generator is started up, it must remain on for at least the minimum uptime, and if it is turned off, it must remain off for at least the minimum downtime.
- **Ramps:** Ramps refer to the ability of generators to increase or decrease their power output between two consecutive periods. This is constrained by the physical limitations of the generator and its supporting infrastructure. Generators must be scheduled in a way that stays within their ramping capabilities while meeting the expected demand for electricity.
- **Variable start-up costs:** The start-up costs of a generator depend on the number of intervals it has been disconnected. Before a thermal generator can be started, it must reach its working temperature and pressure, and the fuel costs of warming up the generator are considered initial costs. If the generator has been disconnected for a short time, it requires less fuel to reach the required temperature and pressure levels. However, if it has been disconnected for a long time, it requires more fuel to start properly, thus increasing the start-up cost.

In the UCP, modeling the costs of power production by generators is a key aspect that needs to be considered [91]. There are various ways to model the costs of power production, including the following:

- **Linear:** A linear cost model assumes that the cost of producing a unit of electricity is regardless of the level of power produced. This is the simplest cost model and can be useful for certain types of generators, such as renewable energy sources with minimal variable costs.
- **Stepwise:** A stepwise cost model assumes that the cost of producing a unit of electricity varies at specific intervals of power output. This model is more complex than the constant cost model and can accurately represent the true costs of power production.
- **Quadratic:** A quadratic cost model assumes that the cost of producing a unit of electricity increases quadratically as the level of power output increases. This model is more complex than the stepwise cost model and can provide an even more accurate representation of the true power production costs, especially for conventional thermal generators.
- **Staircase:** A staircase cost model is similar to a stepwise cost model but assumes that the cost of power production is constant within each interval of power output. This model can be useful for representing power production costs for certain types of generators, such as hydroelectric generators, with discrete steps in their output levels. The staircase cost model is widely utilized in electricity markets.

The choice of cost model depends on the specific features of the generators being modeled and the desired level of accuracy in the corresponding UCP. In general, more complex cost models can provide a more accurate representation of the true costs of power production but also require more computational resources and can make the optimization problem more difficult to solve.

Overall, the technical constraints related to the features of generators, such as generation limits, minimum up/down times, ramps, and variable start-up costs, present challenges for the UCP in scheduling power generation to meet the expected demand for electricity. However, considering these constraints is crucial for optimizing the use of available resources, minimizing costs, and ensuring the stability and reliability of the power system.

### 2.2.2 SYSTEM CONSTRAINTS

The main constraint in power system operation and planning is to balance the electricity demand, i.e., the load, and the electricity supply, i.e., the generation. This balance is essential to keep the system's frequency within safe limits, ensuring the reliability and stability of the power grid. To achieve this balance, the electricity demand must be continuously monitored and predicted, while the generating units must be scheduled and dispatched to meet the forecasted demand.

One of the challenges in power system operation is that electricity cannot be stored in large quantities, meaning that the generation and demand must be matched in real-time. This constraint requires the system operator to ensure that only the adequate amount of generating power is dispatched for each hour to avoid under or over-generation, which could cause frequency deviations and, in extreme cases, blackouts.

Several mathematical models and optimization techniques are used to achieve the load-generation balance, which considers multiple factors such as the availability and technical characteristics of the generating units, transmission capacity, and demand forecast. The resulting schedule should be feasible and optimized for economic and reliability criteria. The load-generation balance is crucial for the efficient and reliable operation of the power system, and it remains a key research topic in power systems engineering.

## 2.3 THE LOAD FORECASTING

Load demand forecasting is important in planning and operating the electrical power system. Load forecasting is the process of predicting future electrical power demand based on historical data, weather patterns, and other factors. The accuracy of load forecasting is critical for ensuring the reliable and efficient operation of the power system.

In the long term, load forecasting is used for capacity planning and investment decisions. This involves forecasting demand for several years (10 to 20 years) into the future to determine the necessary expansion or upgrades to the power system infrastructure as new power generation facilities, transmission lines, and other related infrastructure. This

helps ensure the power system can meet the expected demand growth while maintaining reliability and stability. The long-term period is focused on identifying long-term trends and patterns in electricity consumption and demand and developing scenarios for how demand may evolve over the longer term.

In the short term, load forecasting supports the daily and weekly scheduling of power generation resources, such as power plants and transmission lines. This involves predicting the expected electricity demand for each hour and day of the week. The forecasts are typically updated daily or weekly, using historical data, weather patterns, and other relevant information.

Short-term load forecasts are used to optimize the use of available resources and reduce the cost of electricity production. By predicting the expected demand, power system operators can schedule the generation resources to minimize the overall production cost while meeting the expected demand. This helps to reduce the cost of electricity for consumers and improve the overall efficiency of the power system.

In the very short-term, load forecasting is used for real-time power system management. This task involves forecasting the electricity consumption for the upcoming minutes or a few hours. Short-term load forecasts are regularly updated every minute through real-time data from the power system.

The real-time load forecasts are used to adjust the output of power plants and switch transmission lines to ensure that electricity supply and demand are balanced. This helps to maintain the stability and reliability of the power system and prevent power outages or other disruptions in service.

The timeliness of load forecasting is essential to ensure that the power system can react promptly to variations in demand. Very short-term load demand forecasts are especially important for managing the power system during high demand or unexpected events, such as equipment failures or extreme weather conditions.

Accurate real-time load forecasting is crucial for managing the supply and demand of electricity in a cost-effective way. Real-time market prices, which are typically based on the marginal cost of producing the next unit of electricity, can be reduced using accurate

---

load forecasts. Inaccurate forecasts can have significant implications for the power system, leading to excess capacity or insufficient capacity to meet demand. Overestimation of demand can result in wasted resources and higher costs, while underestimation can lead to power outages and disruptions in service. Accurate load forecasting enables power system operators to use cheaper and more efficient electricity generators, reducing the marginal cost of producing electricity and ultimately benefiting consumers and the economy. Additionally, accurate load forecasting helps to avoid over-generation and under-generation, both of which can have negative economic consequences.

Overall, load forecasting is an essential tool for effectively and efficiently managing the power system. By predicting the expected demand, power system operators can optimize the use of available resources, reduce costs, and maintain the stability and reliability of the power system.

# HYPOTHESIS AND OBJECTIVES

---

## 3.1 HYPOTHESIS

By utilizing custom methods that leverage the mathematical structure and intrinsic features of scheduling problems in power system operations, significant improvements can be achieved in reducing runtime and enhancing accuracy during schedule calculation. These advancements can be effectively accomplished by implementing strategies such as hybrid heuristic methods based on mixed-integer programming (matheuristics) to address the complexities of the thermal unit commitment problem and integrating classical statistical approaches with machine learning techniques for accurate, very short-term load forecasting.

## 3.2 RESEARCH OBJECTIVES

The main objective of this dissertation is to enhance the efficiency of operation schedules in power systems by reducing runtime calculations and increasing the accuracy of the solutions. This objective can be achieved by employing customized methods that leverage the mathematical structure and intrinsic features of scheduling problems. Specifically, the research uses hybrid heuristic methods based on mixed-integer programming (matheuristics) to address a thermal unit commitment problem. Additionally, the study integrates classical statistical approaches with machine learning techniques to solve the problem of

very short-term load forecasting.

In order to achieve this primary objective, a set of specific objectives has been established, outlined as follows.

1. To comprehensively review related works on the thermal unit commitment problem, focusing on tight and compact modeling approaches [64] and using hybrid heuristic methods based on mixed-integer programming for its solution. Additionally, thoroughly review the relevant literature on very short-term load forecasting, emphasizing strategies that employ classical statistical methods and machine learning tools, specifically those based on pattern analysis.
2. To build a new thermal unit commitment model with tight and compact features [64], incorporating a staircase cost function and selecting constraints from cutting-edge formulations [54] to ensure the model's efficiency as a benchmark.
3. To develop hybrid heuristic methods based on mixed-integer programming or matheuristics, specifically designed to align with the mathematical structure of the problem in order to solve the thermal unit commitment model at hand efficiently.
4. To develop a method explicitly designed to leverage the distinctive characteristics of load demand time series and align with the specific requirements of real-time electricity markets to solve the short-term forecasting problem while concurrently enhancing accuracy and reducing runtime.
5. To evaluate the performance of the proposed methods for solving the thermal unit commitment problem, comparing the results obtained by our methods with those of a commercial solver. Additionally, to assess the performance of our method in solving the very short-term load forecasting problem, we will employ benchmark forecasting models, such as those from the exponential smoothing and Box & Jenkins families [14].

# LITERATURE REVIEW

---

## 4.1 LITERATURE SURVEY ON UCP MODELS

The literature on UCP models is vast [62]. UCP models have been widely used to manage energy production for many decades. Different variations of UCP models addressing particular situations and assumptions have been studied [1]. UCP models can be classified according to generator operating, electrical network, and system constraints. The first set includes technical constraints related to the generators as power limits, ramps, minimum up and down times, and start-up costs. The second set comprises limits in lines and tie-lines. The last one encompasses meeting demand and load-generation balance. Anjos and Conejo [7] outline some examples of these models.

Recent research has established the UCP as a computationally complex challenge. Anjos [6] affirm that the UCP is an NP-hard, large-scale, non-linear, non-convex, combinatorial optimization problem. On the other hand, Bendotti et al. [10] assert that the UCP can be transformed into multiple knapsack problems with linking constraints in time and analyze the complexity of the UCP with respect to the number of units and time periods. The UCP is strongly NP-hard, as proven in this study.

In his doctoral thesis, Morales-España [63] highlights that a significant portion of the UCP literature concentrates on improving the formulation by seeking locally ideal or locally tighter representations for a specific subset of constraints of UCP. However, the thesis concludes that achieving an excessively tight model becomes useless if computational

limitations slow its computability due to the extensive number of variables and constraints involved. Given the insights provided by these findings, recent research has focused on finding tight and compact (T&C) formulations that have shown promising results, particularly in thermal UCP models. Morales-España et al. [64] are the pioneers in this model approach. The thermal UCP model aims to minimize operating costs over a specified period by optimizing the commitment and dispatch of generators that use gas, coal, fuel oil, and diesel as primary energy units. It considers generator capacity, up/down times, ramp rates, and variable start-up costs while meeting electricity demand and reserves.

Recent research on UCP models has concentrated on developing more effective Mixed Integer Linear Programming (MILP) formulations that can represent the various components of the problem more tightly and compactly [65, 67, 89, 90]. These studies have particularly focused on the thermal UCP model. In addition, Guedes et al. [39] have presented a hydraulic UCP model with T&C features.

On one hand, a tighter model reduces or narrows down the best solution space to find the feasible solution and helps methods based on branch and bound (B&B) reach a solution quicker. On the other hand, a more compact model employs fewer constraints and variables and requires fewer computing resources. Knueven et al. [54] have compiled the major modern UCP formulations and proposed different models that balance out T&C features. The new formulations are derived from combining constraints from other models. They also tested the model's performance with instances from electricity markets, showing positive results for tighter models.

These T&C models have shown promising results because they can be extended with additional constraints as Nycander et al. [71] that propose a variant based on power and not in energy. This variant of UCP based on power provides a notable advantage by incorporating a more realistic approach to considering generator ramps. It accurately calculates the generation capacity at the beginning and end of each time period. Additionally, it integrates constraints that account for intra-hour reserve requirements, leading to enhanced practicality within the model.

Finally, UCP models have been solved using the B&B method and its variants [87]. In practice, sophisticated solvers and high computing power are required to solve large-scale

problems at reasonable times. Despite this progress, depending on the size and features of specific instances, sometimes the algorithms cannot reach an optimal solution in the time allowed. In this context, metaheuristic methods help find near-optimal solutions. However, some methods based on Genetic Algorithms [52], Tabu Search [68], and Variable Neighborhoods Search [84] have yet to perform well. Nevertheless, recent research has yielded promising results by integrating metaheuristic methods with mathematical programming to address UCPs. Consequently, the next section comprehensively surveys the most recent matheuristic approaches for solving UCP models.

## 4.2 MATHEURISTICS WORKS IN UCP

Matheuristic or hybrid MIP-based heuristics methods are widely used in optimization for solving complex problems where traditional exact methods may not be efficient. Matheuristics was introduced as a hybrid approach that combines metaheuristics with mathematical programming, i.e., heuristic methods with a mathematical programming component [31, 58, 87].

While heuristic approaches may not guarantee optimal solutions, empirical evidence has demonstrated their ability to outperform exact methods in scenarios when working within a predefined time limit.

The efficiency of B&B algorithms have been significantly improved by implementing heuristic ideas in MILP solvers, as highlighted by a survey conducted by Berthold [12]. Constructive heuristics, such as Relax and Fix (R&F) methods [87] or Feasibility Pump (FP) techniques [13], make use of continuous solutions derived from linear relaxation and rounding. Diving strategies during branching operations of B&B, as outlined by Achterberg et al. [2], are used to improve solution quality further once a feasible solution is found. Matheuristics inspired by local search algorithms are then used to refine the solution quality, such as Local Branching (LB) strategies introduced by Fischetti and Lodi [32], which define neighborhoods with maximum modifications in the incumbent. The Relaxation Induced Neighborhood Search (RINS), proposed by Danna et al. [19], combines relaxation and neighborhood search techniques. RINS fixes shared integer variables

between the incumbent and continuous relaxation solutions, narrowing the search space. This focused exploration within promising regions enhances the efficiency and effectiveness of the optimization process.

The kernel search (KS) matheuristic, by Angelelli et al. [5], is inspired by greedy algorithms. It partitions the solution space into a kernel and multiple buckets and then solves a simplified version of the problem or sub-MILPs constructed with the kernel and each bucket. The final solution is achieved by iteratively adding new variables from the buckets to the kernel until all the buckets are utilized. The solution obtained consists of the variables assigned within the kernel.

Implementing matheuristics can improve solution quality, convergence guarantees, and computational efficiency, leading to better solutions. Next, a survey of the matheuristics methods applied to UCP models is discussed.

It is worth mentioning that there is a significant amount of research on the utilization of matheuristics in UCPs. First, Fayzur et al. [30] present two matheuristic approaches in their study. The first approach combines the original version of LB from Fischetti and Lodi [32] and an iterative linear approximation (ILA) method. The second approach combines particle swarm optimization (PSO) and ILA, incorporating a solver in an iterative process. The UCP problem addressed in the study initially involves quadratic production costs, which are later linearized with ILA and solved using the LB method. The model's constraints encompass power balance, spinning reserve requirements, minimum up and down times, production limits, and ramps [86]. The study successfully solves previously unsolvable instances and achieves faster optimal results for larger instances. The proposed LB-ILA algorithms significantly reduce the mean CPU time for larger instances by 40%-56%. However, these algorithms perform worse for smaller instances due to unnecessary exploration of neighborhoods.

Sabóia and Diniz [77] solve a stochastic thermal network-constraint UPC using the LB approach combined with an iterative approach that considers transmission lines flow limits. They dynamically introduce violated flow limit constraints, embedding them into the nodes of the LB scheme. This iterative procedure involves fixing the optimal integer solution achieved at each node. The DC power flow is then evaluated iteratively, contin-

uously checking for violations of line capacity. The process continues until no further line capacity violations are observed. This model is a new compact MILP formulation with basic constraints for thermal UCP. The method is tested with one instance.

Todosijević et al. [84] propose a hybrid approach that combines variable neighborhood search (VNS) with mathematical programming to address a UCP. In their method, the commitment of generators, i.e., deciding which ones will be turned on and off, is determined using VNS. Additionally, they solve an economic dispatch problem (calculus of each generator's power level) for each period by formulating the dispatch as a linear programming problem. The results obtained by the proposed approach outperformed considerable metaheuristics reported in the literature for solving this version of the UCP. Furthermore, the solutions achieved were very close to the optimal solutions obtained with the CPLEX solver. However, the paper lacks information regarding the CPU times of the solver, making it impossible to compare the computational efficiency of VNS directly. Nevertheless, it is important to note that the mathematical model used in the study considers only constraints of demand, reserves, power limits, minimum up and down times, and hot and cold start-up costs without incorporating ramps, which are essential constraints in real-life UCPs. Furthermore, a particular feature of their UCP is that the production cost of the generators is quadratic.

Dupin and Talbi [25] use a matheuristic method to solve a discrete thermal UCP from programming thermal generators in the real-time electricity French system. First, in a constructive phase, they use R&F strategies to find one initially good solution. In the subsequent improvement phase, a VNS algorithm conducts a local search by iteratively exploring B&B solutions in neighborhoods defined within the solution space confined by the MILP model. In other words, these neighborhoods are defined within the MILP using various heuristics suggested by the authors, such as RINS or LB strategies. The model used by Dupin and Talbi [25] is based on Dupin [24] work incorporating various constraints such as demand, reserves, minimum up and down times [75], and fixed start-up costs. Note that in the discrete thermal UCP, the limits of generation constraints are not applicable. Instead, discrete power levels and transition constraints exist between them. However, the model does not incorporate ramps. The method aims to find the best solutions for the discrete thermal UCP on a 15-minute time limit from its application in a real-time

scenario. The resulting matheuristic method is better than the MILP solved directly by the solver in the allotted time frame.

In follow-up work, Dupin and Talbi [26] improved the discrete thermal UCP formulation with T&C features and extended the model to include minimum-stop ramping constraints for thermal generators. In discrete thermal UCPs, min-stop constraint refers to the minimum duration a generator must operate at a specific power level before transitioning to another power level. The “minimum-stop ramping” represents the rate at which the generator can smoothly change to another power level. To address the problem, the authors propose a set of constructive matheuristics specifically designed to generate advantageous feasible solutions. These solutions can be effective initial solutions for a branch-and-bound (B&B) algorithm. Additionally, they explore the implementation of parallelization strategies. Some constructive matheuristics apply variable fixing strategies, which involve fixing the generator output during a set of periods and relaxing the variables during other sets. Another approach involves fixing a prior solution in one block of variables and applying a MILP to another block. Moreover, another strategy employed is dividing the problem into three periods and solving the initial and final parts independently, fixing the resulting solutions in the entire problem, and then solving the ensuing MILP. Another interesting constructive matheuristic implements the KS ideas proposed by Guastaroba et al. [38], i.e., the segregation of generators based on their marginal costs, eliminating units with higher production costs. Another variant of their heuristic prioritizes utilizing low-cost units as the main production basis, while the remaining units constitute the kernel of the KS algorithm. However, the authors neither utilize the complete KS method nor provide specific details about the method. The proposed method produces precise and high-quality solutions surpassing the solution outperforming the B&B method applied to the MILP model in the allotted time limit. The authors tested the method with more than 600 instances from the French electrical system.

In the investigation undertaken by Santos et al. [79], the authors present a model and method used for planning operations for energy generation in Brazil; their work considers thermal generation and hydro generation and limitations on the electrical network. They used a variant of the FP from Fischetti et al. [33] adapted for binary variable decisions. They also created an objective function adapted with a mathematical term calculating

the Hamming distance. The authors opted for a heuristic method due to the limitations in time to solve the UCP, which should be solved daily for a time horizon of seven days. Considering the dimensions of their problem, utilizing an exact method did not yield a timely resolution within the allotted two-hour timeframe. The authors report using the same version of LB as that of Sabóia and Diniz [77].

In the study conducted by Harjunkoski et al. [41], a R&F method for solving the UCP is devised. The proposed method starts by solving a linear relaxation of the problem. Subsequently, they analyzed the product of the commitment binary variable with each generator's power level. If the product exceeds the minimum feasible generation level, the commitment binary variables are fixed and solved as a MILP problem. This constructive method provides a warm start to the B&B algorithm and speeds-up the UCP solution process.

In the following, we discuss the key distinctions between our work and the related studies, emphasizing the scientific contributions of our research.

Regarding the mathematical modeling of the problem, our UCP is a deterministic thermal model with a staircase cost function. Furthermore, our model incorporates valid inequalities in the cost constraints, making the formulation tighter. This cost function assigns different price levels for each generation level interval, aligning the model more closely with the cost modeling of electricity markets. Other works address similar problems with linear costs [77, 25, 26, 41] and quadratic costs [30, 84]. Another notable difference is that the modeling of generator startup costs is simplified in most of the papers to a fixed startup cost [25, 26, 79, 41] or hot and cold starts [30, 84], or, at best, exponential startup costs depending on the time the generator has been off [77]. In our case, we use variable startup cost constraints [64] with tighter features.

The main difference with the other works is that, although they also solve a thermal UCP, they address different variants. For example, Dupin and Talbi [25, 26] tackle a discrete thermal UCP for a real-time horizon with limited forward scheduling periods. [79] deal with a hydrothermal UCP problem, while Sabóia and Diniz [77] solve a stochastic UCP thermal problem. Although our model does not currently consider the power grid, it can be considered in future work.

Regarding solution methods, we have developed five methods to solve the problem: four versions of LB and one version of the kernel search (KS) method. Both LB and KS versions are unique implementations not previously reported in the literature. On the one hand, our variants of LB implement concepts such as soft-fixing, restricted candidate lists (RCL), and local search on variables identified as dominant in the mathematical model instead of searching in all binary variables. On the other hand, the version of KS that we present has some differences from the original method proposed by Angelelli et al. [5]. First, our method excludes kernel construction since a preliminary constructive stage already provides the kernel. Another important difference is the use of the statistical rule of Sturges to determine the number of buckets in KS. Also, our KS only focuses on the dominant variables of the problem. Unlike the original KS approach, the calculation of reduced costs in our method involves fixing the kernel variables while leaving the remaining variables free, based on the linear relaxation of the problem. In our version of KS, the kernel expansion process continues even after processing the buckets, and if there is remaining time, the kernel is reconfigured using the last solution, and the expansion process is restarted.

Although Fayzur et al. [30] present a hybrid LB method with cost linearization, the version they report is the original one developed by Fischetti and Lodi [32]. Alternatively, Todosijević et al. [84] apply VNS to solve the problem and incorporates linear programming to address a subproblem. Similarly, Dupin and Talbi [25] employ another VNS and some of the neighborhoods they define incorporate the concepts of LB and RINS matheuristics. Moreover, it is worth noting that only the local search definition from the LB method is utilized without involving the branching phase. Additionally, Sabóia and Diniz [77] modify the original LB to include the power network.

In their research [79], the authors use a variant of FP as a constructive strategy and LB as an improvement strategy. Furthermore, the authors state that they utilize the same version of LB as Sabóia and Diniz [77].

The first paper that reports the use of the KS for UCP is by Dupin and Talbi [26], where they present their findings on incorporating the KS concept as a variable fixing criterion into a variable fixing strategy. This strategy introduces methods to select specific variables to fix a priori to obtain a reduced problem that the MILP solver can subsequently solve. However, it is important to emphasize that their work lacks full implementation of

the KS method.

Additionally, Harjunoski et al. [41] propose a constructive method based on R&F, where the initial feasible solution provides a warm start to the B&B algorithm, resulting in an accelerated UCP solution process. Finally, we proposed a new constructive method that aims to compete with the approach of Harjunoski et al. [41] and the solver.

In summary, the differences between the related works and our study are significant. On the mathematical modeling side, they address different variants and formulations of the thermal UCP. On the solution methodologies side, the related papers utilize distinct implementations of LB and KS, which differ from those proposed in our study. It is important to highlight that the works included in this comparative analysis specifically focus on solving the UCP using matheuristics strategies.

In Table 4.1, the comparison between the works that have addressed the UCP problem with matheuristic approaches is presented. The table lists the main constraints of the UCP models addressed in each work and their corresponding solution methods.

### 4.3 LOAD FORECASTING WORKS

Many different approaches are adopted for load demand forecasting [8]. For instance, those based on Machine Learning (ML), such as Artificial Neural Networks (ANN) [40] or Support Vector Machines (SVM), have had good results in accuracy in forecasting demand from one day ahead. However, training these takes considerable time due to its high computational cost. Furthermore, it requires periodic retraining due to changes in demand patterns behavior [50]. For example, the change from winter to summer or the beginning of the SARS-CoV-2 pandemic. [29]. In addition, powerful computer equipment is required for its operation. Among the many works around electricity demand forecasting, we highlight the work of Capuno et al. [16]. They proposed a method for real-time operation using hybrid algebraic prediction and Support Vector Regression (SVR). The first method generates the baseline forecast, whereas the second compensates for deviations caused by drastic changes in temperature and humidity. The proposed method in this dissertation adopts the concept of establishing a baseline forecast and subsequently

Table 4.1: Comparison of works addressing the UCP using matheuristics.

Article	Method	Features UCP	Production cost	Start-up cost	Shut- down cost	Reserve require- ment	Gen lim- its	Ramp up/down	Start- up/shut- down ramps	min up/down times	Network Notes	
Fayzur et al. [30] 2014	LB-ILA, PSO-ILA	thermal	quadratic	Cold and hot start-up	yes	yes	Limits bounded by ramps	yes	no	yes	no	LB's original version and cost linearization is used.
Todosijević et al. [84] 2016	VNS-linear program- ming	thermal	quadratic	Cold and hot start-up	yes	yes	yes	no	no	yes	no	no solver times reported
Sabóia and Diniz [77] 2016	LB	stochastic thermal compact formulation	linear	Exponential start-up cost	yes	no	yes	yes	no	yes	yes	A variant of LB that incor- porates violated flow limits in each node.
Dupin and Talbi [25] 2016	R&F, VNS with MIP neighbor- hoods. LB, RINS.	discrete thermal, real-time	linear	Fixed cost	no	primary and sec- ondary reserves	discretized	no	no	yes	no	Utilizes RINS and LB, among other strategies, to define neighborhoods.
Dupin and Talbi [26] 2018	R&F-B&B	discrete thermal, real-time	linear	Fixed cost	no	primary and sec- ondary reserves	discretized	min-stop ramping	no	yes	no	Multiple construction meth- ods running in parallel. KS used only as a concept in two heuristics
Santos et al. [79] 2020	FP, LB	Hydrothermal	linear	Fixed cost	yes	no	yes	yes	yes	yes	yes	LB's Sabóia and Diniz [77] and a variant of FP
Harjunkoski et al. [41] [17] 2021	R&F	thermal	linear	Fixed cost	yes	yes	yes	yes	no	yes	no	A construct that sets the variables of engagement with a selection rule and solves a B&B with warm-start
<b>This work</b>	R&F, LB, KS.	thermal T&C model	staircase	Variable start-up cost	yes	yes	yes	yes	yes	yes	no	Four variants of LB (applying RCL and soft-fixing) and one variant of KS. A new R&F construction method. Comparison with a T&C formulation.

refining it using real-time data. This approach involves correcting the initial forecast by incorporating more recent and up-to-date information.

Recently, more complex methods using ML and deep learning models for load forecasting have been proposed [85, 76]. However, these require high training times, an exacerbated problem each time the model is retrained due to the arrival of new demand data. In contrast, other forecasting methods that have been widely used are based on linear regressions [69]. These methods are computationally cheap but are less accurate. Environmental factors such as temperature, humidity, rain, and calendar data are among the most common regression variables used to model electricity demand.

Another different model for time series forecasting is proposed by Monache et al. [61]. They proposed the Analogies (An) method initially used in meteorology and climatology. The method assumes that errors in calculating a forecast between similar days can happen again with some probability in the near future. The grouping of days is done by measuring the distance between the days. An analogue space is a selection of those days with matching features and the data set of subsequent errors. These models have been applied with excellent results by Alessandrini et al. [3, 4] to forecast renewable resources in photovoltaic plants and wind farms, respectively. They proposed the An method where a variable is a forecast from a set of its most similar past predictions. Azevedo et al. [9] outline its dynamic time scan forecasting based on analogies as a method fast for large data sets to forecast wind speed time series. The scan procedure uses polynomial regression models as a distance measure to identify similar patterns throughout the time series.

Electricity demand has multi-seasonal characteristics that Gould et al. [37] and Liviera et al. [56] have extensively studied. They have identified multiple seasonalities in the time series. This seasonality defines demand patterns between days and seasons. Dudek [22] studies the seasonal characteristics of the demand and determines some patterns using tools such as Nearest Neighbors (NN). He finds the criteria between days of the same season by minimizing the distances between a window of the latest data against the time series.

Another work related to one day-ahead load forecasting model based on similar patterns searching using NN and combining sample selection with multiple linear regression

is proposed by Dudek [23]. This model refines the regression with reduction variables techniques such as stepwise, Lasso regression, and principal component analysis (PCA). The central idea of this work is to identify, with precision, days with seasonal cycle patterns. Further, days are classified by weekday, Saturday, Sunday, and holidays. Finally, the model was applied for a forecast of the previous day. The ideas presented by the author apply in a similar way to the present dissertation, such as sample selection, regression of samples for determination of a model, and calculation of the forecast using immediate after-sample data.

Ngo et al. [70] propose a method for ultra-short-term load forecasting. This model is based on robust Holt-Winters double seasonal exponential smoothing. Forecasting results for distribution feeder loads showed this method has good prediction accuracy. Exponential smoothing forecasting is widely recognized as a benchmark approach due to its high accuracy and low computational cost.

Unlike the other works that reported used method based-analogies for next-day forecasting in wind power forecasting, solar power forecasting, and of course, climatology, we use the analogies approach in the field of very short-term load forecasting. Also, we distinguish the unique combination of load demand forecasting approaches based on analogs and the integration of moving averages to correct baseline forecasts. To the best of our knowledge has yet to be reported in the literature. The comparison of similarity metrics to select days with a high correlation and using regression models within a unified framework, this work presents a novel framework that results in a significant contribution to the field load forecasting.

#### 4.4 SUMMARY OF THESIS SCIENTIFIC CONTRIBUTIONS

The first significant contribution focuses on solving a variation of a unit commitment problem with the development of matheuristic methods based on local branching and kernel search; these methods have shown promise in solving efficiently large and complicated generation-schedule problems on time much faster and more near-optimal than the solver alone. Moreover, another contribution is developing a novel constructive method that

obtains feasible solutions of good quality within the allowed time limit for all instances.

From a modeling perspective, a contribution was made with the presentation of a tight and compact model for a variant of the UCP. This model incorporates a staircase feature to represent costs accurately and integrates valid inequalities to tighten the problem. The formulation provides a solid foundation from which it is possible to develop tailor-made UCP models that effectively address the distinctive characteristics of electricity markets, including the Mexican market. Another significant contribution focuses on efficiently calculating real-time load forecasting using a method based on analogies and moving averages. This method envisions the potential to overcome the trade-off between accuracy and speed in current forecasting methods. It demonstrates high computational efficiency and robustness, making it well-suited for operating in a real-time electricity market environment, particularly for predicting load demand over horizons ranging from five minutes to a few hours.

# A TIGHT AND COMPACT THERMAL UNIT COMMITMENT MODEL

---

It has been previously stated that the UCP is a challenging optimization problem in electrical power systems. The major problem in unit commitment scheduling is obtaining high-quality solutions that accurately represent the real-world conditions of the power system while ensuring timely solutions is a significant challenge. Power systems' large and complex nature and the need to consider various variables and constraints in decision-making add to the complexity. Developing efficient algorithms and optimization methods that can balance this trade-off is crucial to achieving high-quality solutions on time for the UCP.

MILP models are widely used in many fields, including power systems, and involve finding the optimal solution to a problem with linear constraints and integer and continuous decision variables. UCP is commonly modeled as MILP, requiring optimizing a linear objective function subject to constraints, where decision variables are usually binary and continuous, representing whether a generator is committed during a particular time interval. Constraints within UCP mean the physical and operational limitations of the power system, such as ramping limits, minimum and maximum generator output levels, and power demand, among others. The primary goal of MILP is to determine the optimal values of decision variables that meet all the constraints while optimizing the objective function.

To solve MILP problems, the branch-and-bound method and its variants are commonly used. The basic idea behind the branch-and-bound method is to explore the search space of possible solutions, dividing it into smaller sub-problems until the optimal solution is found. This method is effective and accurate but can be computationally expensive, particularly for large-scale problems.

Many commercially available optimization packages implement the branch-and-bound method, and they are commonly used to solve the UCP. These off-the-shelf solvers are effective and accurate, although they have limitations in terms of processing time. In addition, they can be sensitive to the mathematical formulation used to represent the problem. Therefore, it is important to carefully formulate the problem to achieve the best possible results with the available optimization tools.

Mathematical models, known as tight and compact (T&C) formulations, are employed to represent the constraints and objectives of the problem to address it effectively.

The tightness of a MILP problem defines the search space that the solver needs to explore to find the optimal (integer) solution; the compactness of a MILP problem refers to its size and defines the searching speed that the solver takes to find the optimal solution.

Developing accurate models for the unit commitment problem is essential, but it can also increase the computational burden of the problem. To address this issue, T&C formulations have been developed to solve the MILP efficiently. Tight formulations incorporate valid inequalities into the problem, creating a strong formulation of the convex hull of the set of integer solutions and narrowing down the search space for the solution. Compact formulations, on the other hand, focus on reducing the number of variables and constraints involved in the problem, making the solution process computationally less expensive. The T&C models, which are both tighter and more compact, reduce the computational burden by requiring fewer constraints, variables, and nonzero elements in the constraint matrix and are much more computationally efficient than other UCP formulations in the literature. Therefore, a successful UCP formulation should strive for both tightness and compactness, as this approach yields solutions more efficiently within a reasonable amount of time. The T&C model, first introduced by [64], demonstrated a significant reduction in computational burden and improved solution qualities compared to previous cutting-edge

formulations, such as those presented in [17] and [72]. The most extensive research on UCP formulations with T&C features is shown in Knueven et al. [54]. The authors compiled a comprehensive list of modern UCP formulations and proposed multiple models to balance T&C features by combining constraints from other models. The new formulations were tested using actual instances and showed favorable results for tighter models.

A T&C formulation of the thermal UCP is introduced in this chapter, incorporating constraints from various authors. The inclusion of this model serves the purpose of providing a robust benchmark for comparison with the matheuristic solution methods proposed in the subsequent chapter. Essentially, this formulation is a reliable reference for assessing the effectiveness of our matheuristic methods outlined in Section 6.

To verify the accuracy of the model, we compared its results with those presented by Morales-España et al. [64]. This was done by utilizing a small instance of their proposed eight generators.

## 5.1 THE THERMAL T&C UCP PROPOSED MODEL

The thermal T&C UCP formulation developed considers various constraints, including power limits, minimum uptime/downtime, ramp capabilities of generators, variable start-up costs, and demand and reserve requirements. Our model also incorporates staircase cost production and its valid inequalities to reduce the solution space. Despite being more sophisticated, our formulation is more compact with fewer restrictions than less advanced models.

### 5.1.1 NOTATION

A summary of sets, indices, parameters, and decision variables is enlisted for quick reference.

*Sets and indices of the power system:*

$\mathcal{G}$  Set of generators ( $g \in \mathcal{G}$ )

$\mathcal{G}^1$	Subset of generators ( $g \in \mathcal{G}$ ) ; (if $UT_g = 1$ )
$\mathcal{G}^{1*}$	Subset of generators ( $g \in \mathcal{G}$ ) ; (if $UT_g = 1$ and $SU_g \neq SD_g$ )
$\mathcal{G}^{>1}$	Subset of generators ( $g \in \mathcal{G}$ ) ; (if $UT_g > 1$ )
$\mathcal{T}$	Set of time periods in the planning horizon; ( $t \in \mathcal{T}$ )

*Sets and indices of cost-related aspects:*

$\mathcal{S}_g$	Set of start-up cost curve segments for a generator ( $g \in \mathcal{G}$ ) from hottest ( $s = 1$ ) to coldest ( $s =  \mathcal{S}_g $ ); ( $s \in \mathcal{S}_g$ ).
$\mathcal{L}_g$	Stairwise production cost intervals for generator ( $g \in \mathcal{G}$ ); ( $l \in \mathcal{L}_g$ )

*Parameters:*

$C_g^R$	Minimum operating cost of generator ( $g \in \mathcal{G}$ ) that works at least at minimum power $\underline{P}_g$ ; in \$
$C_g^l$	Cost coefficient for stairwise segment ( $l \in \mathcal{L}_g$ ) of generator ( $g \in \mathcal{G}$ ) that works at least at minimum power $\underline{P}_g$ ; in \$/MWh
$C_{g,s}^S$	Start-up cost for generator ( $g \in \mathcal{G}$ ) with a set of segment time ( $s \in \mathcal{S}_g$ ); it determines the starting cost by locating a cost in a segment time $s$ in the intervals $[\underline{T}_{g,s}, \bar{T}_{g,s})$ ; in \$/h
$C_g^{SD}$	Shut-down cost for generator ( $g \in \mathcal{G}$ ); in \$
$De_t$	Energy load demand in period ( $t \in \mathcal{T}$ ); in MWh
$p_{g,0}$	Power output of a generation ( $g \in \mathcal{G}$ ) at time 0; in MW
$\bar{P}_g, \underline{P}_g$	Maximum and minimum generation value of generator ( $g \in \mathcal{G}$ ); in MW
$\bar{P}_g^l$	Maximum power available for staircase segment ( $l \in \mathcal{L}_g$ ) of generator ( $g \in \mathcal{G}$ ); in MW
$R_t$	System-wide spinning reserve requirement in period ( $t \in \mathcal{T}$ ); in MW
$RD_g$	Ramp-down rate is the capacity of generator ( $g \in \mathcal{G}$ ) to decrease power between two consecutive periods; in MW/h
$RS_g$	Ramp-up rate of generator ( $g \in \mathcal{G}$ ) to increase power when the generator is starting; in MW/h
$RU_g$	Ramp-up rate of generator ( $g \in \mathcal{G}$ ) to increase power between two consecutive periods; in MW/h

$ \mathcal{S}_g $	Number of segments in the set $\mathcal{S}_g$
$SU_g$	Start-up rate for a generator ( $g \in \mathcal{G}$ ); in MW/h
$SD_g$	Shut-down rate for a generator ( $g \in \mathcal{G}$ ); in MW/h
$T_g^{\text{RU}}$	Time that a generator ( $g \in \mathcal{G}$ ) spends ramping to go from $SU_g$ to $\bar{P}_{g,t}$ ; in h
$T_g^{\text{RD}}$	Time that a generator ( $g \in \mathcal{G}$ ) spends ramping to go from $\bar{P}_{g,t}$ to $SD_g$ ; in h
$TC_g$	Time offline after a generator ( $g \in \mathcal{G}$ ) turned into cold
$\underline{T}_{g,s}$	Start of start-up cost segment ( $s \in \mathcal{S}_g$ ), respectively; in h, (i.e. $\underline{T}_{g,1} = DT_g, \underline{T}_{g,S_g} = TC_g$ )
$UT_g, DT_g$	Minimum up/down time for a generator ( $g \in \mathcal{G}$ ); in h
$u_{g,0}$	Status of generator ( $g \in \mathcal{G}$ ) at time 0
$U_g$	Number of periods generator ( $g \in \mathcal{G}$ ) is required to be online at $t = 1$ ; in h
$D_g$	Number of periods generator ( $g \in \mathcal{G}$ ) is required to be offline at $t = 1$ ; in h

*Binary variables:*

$u_{g,t}$	Equal to 1 if generator ( $g \in \mathcal{G}$ ) is online in period ( $t \in \mathcal{T}$ ), and 0 otherwise
$v_{g,t}$	Equal to 1 if generator ( $g \in \mathcal{G}$ ) starts up at the beginning of period ( $t \in \mathcal{T}$ ), and 0 otherwise
$w_{g,t}$	Equal to 1 if generator ( $g \in \mathcal{G}$ ) is shut-down at the beginning of period ( $t \in \mathcal{T}$ ), and 0 otherwise
$\delta_{g,t,s}$	Equal to 1 if generator ( $g \in \mathcal{G}$ ) have a start-up type $s \in \mathcal{S}$ in period ( $t \in \mathcal{T}$ ), and 0 otherwise

*Real variables:*

$c_{g,t}^{\text{P}}$	Production cost over $\underline{P}$ of generator ( $g \in \mathcal{G}$ ) in period ( $t \in \mathcal{T}$ ); in \$
$c_{g,t}^{\text{SD}}$	Shut-down cost of generator ( $g \in \mathcal{G}$ ) in period ( $t \in \mathcal{T}$ ); in \$
$c_{g,t}^{\text{SU}}$	Start-up cost of generator ( $g \in \mathcal{G}$ ) in period ( $t \in \mathcal{T}$ ); in \$
$p_{g,t}$	Amount of power a generator ( $g \in \mathcal{G}$ ) produces in period ( $t \in \mathcal{T}$ ), in MW
$p'_{g,t}$	Amount of power above minimum $\underline{P}_g$ that a generator ( $g \in \mathcal{G}$ ) produces in period ( $t \in \mathcal{T}$ ), in MW
$\bar{p}_{g,t}$	Maximum power available from generator ( $g \in \mathcal{G}$ ) produces in period ( $t \in \mathcal{T}$ ), in MW

$\bar{p}'_{g,t}$	Maximum power available above minimum from generator ( $g \in \mathcal{G}$ ) produces in period ( $t \in \mathcal{T}$ ), in MW
$p^l_{g,t}$	Power from staircase segment ( $l \in \mathcal{L}_g$ ) from generator ( $g \in \mathcal{G}$ ) produces in period ( $t \in \mathcal{T}$ ), in MW
$r_{g,t}$	Spinning reserves provided by a generator ( $g \in \mathcal{G}$ ) in period ( $t \in \mathcal{T}$ ), in MW

The parameters  $C_{g,l}^v$  and  $C_{g,l}^w$  are calculated as follows [54].

$$C_{g,l}^v = \begin{cases} 0 & \text{if } \bar{P}_g^l \leq SU_g \\ \bar{P}_g^l - SU_g & \text{if } \bar{P}_g^{l-1} < SU_g < \bar{P}_g^l \\ \bar{P}_g^l - \bar{P}_g^{l-1} & \text{if } \bar{P}_g^{l-1} \geq SU_g \end{cases} \quad (5.1)$$

$$C_{g,l}^w = \begin{cases} 0 & \text{if } \bar{P}_g^l \leq SD_g \\ \bar{P}_g^l - SD_g & \text{if } \bar{P}_g^{l-1} < SD_g < \bar{P}_g^l \\ \bar{P}_g^l - \bar{P}_g^{l-1} & \text{if } \bar{P}_g^{l-1} \geq SD_g \end{cases} \quad (5.2)$$

Finally,  $T_g^{\text{RU}}$  and  $T_g^{\text{RD}}$  are also computed as follows [54]:

$$T_g^{\text{RU}} = \left\lfloor \frac{\bar{P}_g - SU_g}{RU_g} \right\rfloor, \quad (5.3)$$

$$T_g^{\text{RD}} = \left\lfloor \frac{\bar{P}_g - SD_g}{RD_g} \right\rfloor. \quad (5.4)$$

### 5.1.2 MATHEMATICAL FORMULATION

Objective function:

$$\min \sum_{t \in \mathcal{T}} \sum_{g \in \mathcal{G}} C_g^{\text{R}} u_{g,t} + c_{g,t}^{\text{P}} + c_{g,t}^{\text{SU}} + c_{g,t}^{\text{SD}}. \quad (5.5)$$

Subject to:

$$\sum_{i=t-UT_g+1}^t v_{g,i} \leq u_{g,t}, \quad g \in \mathcal{G}, t \in \{UT_g, \dots, |\mathcal{T}|\}, \quad (5.6)$$

$$\sum_{i=t-DT_g+1}^t w_{g,i} \leq 1 - u_{g,t}, \quad g \in \mathcal{G}, t \in \{DT_g, \dots, |\mathcal{T}|\}, \quad (5.7)$$

$$\begin{aligned} \min\{U_g, |\mathcal{T}|\} \\ \sum_{i=1} u_{g,i} = \min\{U_g, |\mathcal{T}|\}, \end{aligned} \quad g \in \mathcal{G}, \quad (5.8)$$

$$\begin{aligned} \min\{D_g, |\mathcal{T}|\} \\ \sum_{i=1} u_{g,i} = 0, \end{aligned} \quad g \in \mathcal{G}, \quad (5.9)$$

$$u_{g,t} - u_{g,t-1} = v_{g,t} - w_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.10)$$

$$\begin{aligned} p'_{g,t} + r_{g,t} &\leq (\bar{P}_g - \underline{P}_g) u_{g,t} - (\bar{P}_g - SU_g) v_{g,t} \\ &\quad - [SU_g - SD_g]^+ w_{g,t+1}, \end{aligned} \quad g \in \mathcal{G}^1, t \in \{1, \dots, |\mathcal{T}| - 1\}, \quad (5.11)$$

$$\begin{aligned} p'_{g,t} + r_{g,t} &\leq (\bar{P}_g - \underline{P}_g) u_{g,t} - (\bar{P}_g - SD_g) w_{g,t+1} \\ &\quad - [SD_g - SU_g]^+ v_{g,t} \end{aligned} \quad g \in \mathcal{G}^1, t \in \{1, \dots, |\mathcal{T}| - 1\}, \quad (5.12)$$

$$\begin{aligned} p_{g,t} &\leq \bar{P}_g u_{g,t} - \sum_{i=0}^{T_g^{\text{RU}}} (\bar{P}_g - (SU_g + iRU_g)) v_{g,t-i} \\ &\quad - \sum_{i=0}^{T_g^{\text{RD}}} (\bar{P}_g - (SD_g + iRD_g)) w_{g,t+1+i} \end{aligned} \quad g \in \mathcal{G}, t \in \{T_g^{\text{RU}}, |\mathcal{T}| - T_g^{\text{RD}}\}, \quad (5.13)$$

$$\begin{aligned} p'_{g,t} + r_{g,t} &\leq (\bar{P}_g - \underline{P}_g) u_{g,t} - (\bar{P}_g - SU_g) v_{g,t} \\ &\quad - (\bar{P}_g - SD_g) w_{g,t+1} \end{aligned} \quad g \in \mathcal{G}^{>1}, t \in \{1, \dots, |\mathcal{T}| - 1\}, \quad (5.14)$$

$$p'_{g,t} + r_{g,t} \leq (\bar{P}_g - \underline{P}_g) u_{g,t} - (\bar{P}_g - SU_g) v_{g,t} \quad g \in \mathcal{G}^1, t \in \mathcal{T}, \quad (5.15)$$

$$p'_{g,t} + r_{g,t} \leq (\bar{P}_g - \underline{P}_g) u_{g,t} - (\bar{P}_g - SD_g) w_{g,t+1} \quad g \in \mathcal{G}^1, t \in \{1, \dots, |\mathcal{T}| - 1\}, \quad (5.16)$$

$$\bar{p}'_{g,t} - p'_{g,t-1} \leq (SU_g - \underline{P}_g - RU_g) v_{g,t} + RU_g u_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.17)$$

$$p'_{g,t-1} - p'_{g,t} \leq (SD_g - \underline{P}_g - RD_g) w_{g,t} + RD_g u_{g,t-1}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.18)$$

$$p_{g,t}^l \leq (\bar{P}_g^l - \bar{P}_g^{l-1}) u_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, l \in \mathcal{L}_g, \quad (5.19)$$

$$\sum_{l \in \mathcal{L}_g} p_{g,t}^l = p'_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.20)$$

$$\sum_{l \in \mathcal{L}_g} C_g^l p_{g,t}^l = c_{g,t}^{\text{P}}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.21)$$

$$p'_{g,t} \leq (\bar{P}_g - \underline{P}_g) u_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.22)$$

$$\begin{aligned}
p_{g,t}^l &\leq \left(\bar{P}_g^l - \bar{P}_g^{l-1}\right) u_{g,t} - C_{g,l}^v v_{g,t} - C_{g,l}^w w_{g,t+1}, & g \in \mathcal{G}^{>1}, \\
& & t \in \{1, \dots, |\mathcal{T}| - 1\}, l \in \mathcal{L}_g,
\end{aligned} \tag{5.23}$$

$$\begin{aligned}
p_{g,t}^l &\leq \left(\bar{P}_g^l - \bar{P}_g^{l-1}\right) u_{g,t} - C_{g,l}^v v_{g,t}, & g \in \mathcal{G}^1, t \in \mathcal{T}, l \in \mathcal{L}_g,
\end{aligned} \tag{5.24}$$

$$\begin{aligned}
p_{g,t}^l &\leq \left(\bar{P}_g^l - \bar{P}_g^{l-1}\right) u_{g,t} - C_{g,l}^w w_{g,t+1}, & g \in \mathcal{G}^1, \\
& & t \in \{1, \dots, |\mathcal{T}| - 1\}, l \in \mathcal{L}_g,
\end{aligned} \tag{5.25}$$

$$\begin{aligned}
p_{g,t}^l &\leq \left(\bar{P}_g^l - \bar{P}_g^{l-1}\right) u_{g,t} - C_{g,l}^v v_{g,t} - [C_{g,l}^v - C_{g,l}^w]^+ w_{g,t+1}, & g \in \mathcal{G}^{1*}, \\
& & t \in \{1, \dots, |\mathcal{T}| - 1\}, l \in \mathcal{L}_g,
\end{aligned} \tag{5.26}$$

$$\begin{aligned}
p_{g,t}^l &\leq \left(\bar{P}_g^l - \bar{P}_g^{l-1}\right) u_{g,t} - C_{g,l}^w w_{g,t+1} - [C_{g,l}^w - C_{g,l}^v]^+ v_{g,t}, & g \in \mathcal{G}^{1*}, \\
& & t \in \{1, \dots, |\mathcal{T}| - 1\}, l \in \mathcal{L}_g,
\end{aligned} \tag{5.27}$$

$$\begin{aligned}
\delta_{g,t,s} &\leq \sum_{i=\underline{T}_{g,s}}^{\underline{T}_{g,s+1}-1} w_{g,t-i}, & g \in \mathcal{G}, t \in \mathcal{T}, s \in [1, |\mathcal{S}_g|),
\end{aligned} \tag{5.28}$$

$$\begin{aligned}
v_{g,t} &= \sum_{s=1}^{|\mathcal{S}_g|} \delta_{g,t,s}, & g \in \mathcal{G}, t \in \mathcal{T},
\end{aligned} \tag{5.29}$$

$$\begin{aligned}
c_{g,t}^{\text{SU}} &= \sum_{s=1}^{|\mathcal{S}_g|} C_g^{\text{S}} \delta_{g,t,s}, & g \in \mathcal{G}, t \in \mathcal{T},
\end{aligned} \tag{5.30}$$

$$\begin{aligned}
\delta_{g,t,s} &= 0, & g \in \mathcal{G}, s \in [1, |\mathcal{S}_g|), \\
& & t \in (\underline{T}_{g,s+1} - DT_g^0, \underline{T}_{g,s+1}),
\end{aligned} \tag{5.31}$$

$$\begin{aligned}
c_{g,t}^{\text{SD}} &= C_g^{\text{SD}} w_{g,t}, & g \in \mathcal{G}, t \in \mathcal{T},
\end{aligned} \tag{5.32}$$

$$\begin{aligned}
\sum_{g \in \mathcal{G}} p_{g,t} &= De_t, & t \in \mathcal{T},
\end{aligned} \tag{5.33}$$

$$\begin{aligned}
\sum_{g \in \mathcal{G}} \bar{p}_{g,t} &\geq De_t + R_t, & t \in \mathcal{T},
\end{aligned} \tag{5.34}$$

$$\sum_{g \in \mathcal{G}} r_{g,t} \geq R_t, \quad t \in \mathcal{T}, \quad (5.35)$$

$$p_{g,t} = p'_{g,t} + \underline{P}_g u_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.36)$$

$$\bar{p}_{g,t} = \bar{p}'_{g,t} + \underline{P}_g u_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.37)$$

$$\bar{p}_{g,t} = p'_{g,t} + r_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.38)$$

$$\bar{p}_{g,t} = p_{g,t} + r_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.39)$$

$$p_{g,t} \leq \bar{p}_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.40)$$

$$p'_{g,t} \leq \bar{p}'_{g,t}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.41)$$

$$u_{g,t}, v_{g,t}, w_{g,t}, \delta_g^s \in \{0, 1\}, \quad g \in \mathcal{G}, t \in \mathcal{T}, \quad (5.42)$$

$$p_{g,t}, p'_{g,t}, \bar{p}'_{g,t}, c_{g,t}^p, c_{g,t}^{\text{SU}}, r_{g,t} \geq 0, \quad g \in \mathcal{G}, t \in \mathcal{T}. \quad (5.43)$$

### 5.1.3 MODEL OVERVIEW

The objective function 5.5 seeks to minimize the total cost, which is composed of the energy production cost  $c_{g,t}^p$ , the fixed cost of operating at a minimum production level  $C_g^R$ . The variable startup cost  $c_{g,t}^{\text{SU}}$  and the shutdown cost  $c_{g,t}^{\text{SD}}$  for each generator  $g \in \mathcal{G}$  during a specific time period  $t \in \mathcal{T}$ .

The correspondence of the constraints in the proposed model with those of the original authors has been listed by the model's constraints. In Chapter 2, there are explanations of the physical meanings behind each constraint.

The minimum uptime and minimum downtime constraints as (5.6) and (5.7) from Rajan and Takriti [75], enforced with (5.8) and (5.9). The logic of the generators' start-up, shut-down, and operation is modeled by (5.10) from Garver [34]. The generation limits constraints are modeled as (5.11) and (5.12) from Gentile et al. [36], and (5.13) from Pan and Guan [73]. The start and shut-down ramp limits are modeled as (5.14), (5.15), and (5.16) from Morales-España et al. [64]. The ramp-up and ramp-down limits constraints are modeled as (5.17) and (5.18) from Damci-Kurt et al. [18].

The staircase production cost constraints are modeled as (5.19), (5.20), and (5.21) from Garver [34], (5.22) from Wu [88], and (5.23), (5.24), (5.25), (5.26), and (5.27) from

Knueven et al. [53]. Note that (5.23), (5.24), (5.25), (5.26), and (5.27) are not indispensable in the formulation, but they serve to tighten the variables' staircase production.

The shut-down cost constraints are modeled as (5.32) from Knueven et al. [54].

Constraints (5.28), (5.29), (5.30), and (5.31) from Morales-España et al. [64] equalize the cost of the segment of the variable of start-up function cost  $c_{g,t}^{S_g}$  of the generator  $g$ , where  $C_{g,s}^S$  is the start-up cost in the category  $s$  of generator  $g$  in \$/MWh.

The demand meets, and reserve requirement constraints are modeled as (5.33) and (5.34) from Ostrowski et al. [72], and (5.35) from Morales-España et al. [64]. Constraints (5.36) from Morales-España et al. [64], (5.37), (5.38), (5.39), (5.40), and (5.41) from Knueven et al. [54] establish linear relationships between the power variables  $p_{g,t}, p'_{g,t}, \bar{p}'_{g,t}, \bar{p}_{g,t}$ . These constraints are utilized in the best UCP formulations [54], which helps to constrain the problem's convex hull. The relationship between the variables related to generator power is depicted in Figure 5.1; the rectangle represents the full operating range of the generator, including the observed minimum and maximum limits and the ranges for energy and reserves.

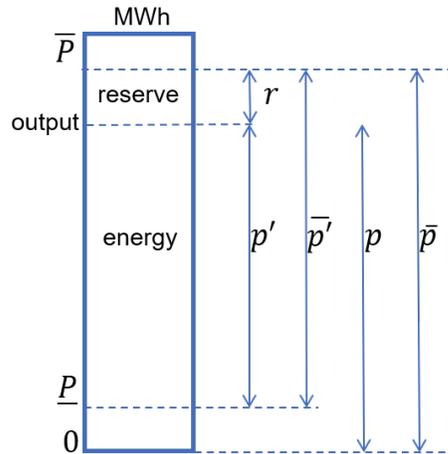


Figure 5.1: The relationship among the power variables in the formulation of thermal UCP.

Lastly, constraints (5.42) and (5.43) define the nature of the variables.

Initial conditions: constraints (5.10),(5.17),(5.18) change when the time is  $t = 0$ . The parameters used in this case to substitute the parameter with  $t - 1$  index are the initial

conditions of the generators such as  $u_{g,0}, p_{g,0}, p'_{g,0}, U_{g,0}, D_{g,0}$ . Also, constraints (5.31) assure the start-up variable  $\delta_{g,t}$  to be zero during the initial periods considering its minimum shut-down time if the generator has been offline [64].

## 5.2 VALIDATION OF THE ACCURACY OF THE PROPOSED MODEL

The proposed thermal T&C UCP model was validated by comparing it with the benchmark formulation developed by Morales-España et al. [64], using the small instances described in their work. Both models contained standard thermal UCP constraints, such as power limits, minimum uptime/downtime, ramp capabilities of generators, variable start-up costs, and demand and reserve requirements. However, our formulation is structured differently and incorporates four valid inequalities and staircase production cost. To ensure a fair comparison, the cost of all staircase steps in our model was set to a single value, consistent with the formulation and data used by Morales-España et al. [64]. Consequently, the eleven steps of the generation cost staircase function in the proposed model had the same value as the costs for these three instances. The consistent results among the eight generators tested in a small benchmark instance have confirmed the model's effectiveness. These results were observed for horizons 1, 3, and 5 days and are detailed in Table 5.1.

Table 5.1: Generators parameters from Morales-España et al. [64]

$g$	Technical Information							Cost coefficients			
	$\bar{P}_g$	$\underline{P}_g$	$UT_g/DT_g$	$RU_g/RD_g$	$U_g/D_g$	$p_{g,0}$	$TC_g$	$C_g^R$	$C_g^{l=1\dagger}$	$C_{g,s=1}^S \ddagger$	$C_{g,s=2}^S \ddagger$
	[MW]	[MW]	[h]	[MW/h]	[h]	[MW]	[h]	[\$/h]	[MW/h]	[\$]	[\$]
1	455	150	8	225	0	455	14	1000	16.19	4500	9000
2	455	150	8	225	0	245	14	970	17.26	5000	10000
3	130	20	5	50	0	0	10	700	16.60	550	1100
4	130	20	5	50	0	0	10	680	16.50	560	1120
5	162	25	6	60	0	0	11	450	19.70	900	1800
6	80	20	3	60	0	0	8	370	22.26	170	340
7	85	25	3	60	0	0	6	480	27.74	260	520
8	55	10	1	135	0	0	2	660	25.92	30	60

<sup>†</sup> The subindex  $s=1$  corresponds to a hot startup, while the subindex  $s=2$  corresponds to a cold startup.

<sup>‡</sup> The energy cost is represented as a linear value with a single numerical value.

The subindex  $l=1$  specifically pertains to a step within a staircase function.

### 5.3 RESULTS AND DISCUSSION

The objective function results obtained by our UCP model match those reported by Morales-España et al. [64] in the three instances. The reported objective value and the one obtained by us are shown below.

- Instance of one day,  $z^{\text{uc-57}} = \$573,630.60$
- Instance of three days,  $z^{\text{uc-58}} = \$1,710,633.60$
- Instance of five days,  $z^{\text{uc-59}} = \$2,8477,708.40$

Table 5.2 provides the committed power generation data in megawatts (MW) to visualize the one-day power generation schedule. The table includes information for eight generators and their respective power output throughout the day.

The results can be repeated using the files `uc_57.json`, `uc_58.json`, `uc_59.json`, located in repository: [https://github.com/urieliram/tc\\_uc/tree/main/instances](https://github.com/urieliram/tc_uc/tree/main/instances).

In addition, the Pyomo code for implementing the optimization model can be found in the repository: [https://github.com/urieliram/tc\\_uc/blob/main/uc\\_Co.py](https://github.com/urieliram/tc_uc/blob/main/uc_Co.py). Please ensure that you have Pyomo version 6.6.1 and the required libraries installed to execute the code successfully.

### 5.4 CONCLUSIONS

This chapter introduced a thermal UCP model with robust features, encompassing the standard thermal model constraints. Through validation tests, the model exhibited outcomes that aligned with the results of a reference author’s research on small instances. This demonstrates the model’s preparedness for the subsequent phase of research, where it will be applied to solve the UCP problem using hybrid MILP or matheuristic algorithms, solving instances considered hard.



In future work, opportunities have been identified to expand this model by incorporating additional constraints such as prohibited zones, sale and purchase offers of reserves with different ramp rates [57], hydraulic generators with non-linear characteristics [79], and variants based on power rather than energy, modeling of combined cycle plants [67], Power-based models rather than energy-based models [71]. Among other variants.

Another potential line of research could involve exploring the modeling of the discretized thermal UCP using constraint programming. As far as we know, no prior studies have investigated constraint programming in this context. This approach can effectively capture operational requirements and limitations, addressing essential constraints such as generator limits, minimum up/down times, ramping constraints, spinning reserves, and demand-supply balancing. A performance comparison can also be conducted with the T&C model presented by [24].

# MATHEURISTIC APPROACH TO SOLVE THE THERMAL UCP

---

Obtaining an optimal schedule solution is a daily challenge faced by electricity market operators; the challenge is even more complicated when time is critical. Less than two hours is typically the expected time to resolve it [79]. Unfortunately, the reality is that wholesale electricity markets operate with near-optimal solutions.

Johnson et al. [48] show that slight variations between near-optimal solutions significantly impact participants' profitability, even though they have a negligible effect on the total cost. Near-optimal UCP solutions with very similar total costs can significantly differ from the marginal costs used to calculate energy and reserve prices in the market. These prices are used to calculate the payment to the generators.

Even with significant computational advances that have made it possible to solve MILP problems with off-the-shelf solvers <sup>1</sup> and reduce the optimality gap in UCP solutions to negligible levels, price volatility problems persist.

The results of a study by Sioshansi et al. [83] show that the size of generators' payment deviations is not necessarily proportional to the size of the optimality gap. Therefore, the authors conclude that even smaller relative optimality gaps do not necessarily mitigate

---

<sup>1</sup>These solvers employ a combination of methods, encompassing preprocessing techniques, cutting plane algorithms, B&B algorithms and their variations, heuristics, solution refinements, and algorithm selection to enhance the optimization process.

economic mismatch and assure that the problem will persist until UCP can be solved in a fully optimal, on time.

In practice, the Independent System Operators (ISO) markets pay “compensation” to generators to recover unpaid start-up and no-load costs due to under-marginal energy prices, which has partially helped to reduce the effects of volatility and surplus differences. For example, a recent study by Eldridge et al. [27] proposed several pricing methods executed after the UCP to aim fair economic retribution to generators. However, although these remedial solutions are available, the problem of price instability will remain until we solve the UCP problem with a zero optimality gap [83]. Despite this, implementing a strategy to decrease the optimality gap could be beneficial.

Strategies to reduce the optimality gap include tightening the UCP problems to models closer to the convex hull commonly by adding valid inequalities. However, Morales-España et al. [64] and Knueven et al. [54] conclude that too many valid inequalities in a model can increase solution time. These authors tackle the trade-off between strengthening models with high computational costs versus increasing computational speed using compact formulations with fewer constraints and variables.

Another strategy is the design of methods that reduce solution times and improve accuracy. In line with this strategy, in this dissertation, we propose several hybrid heuristic methods based on MILP, known as matheuristics [58].

The proposed solution strategy consists of a phase of construction and improvement phase.

- In the constructive phase, a feasible solution is initially obtained using one of two methods. The first method, proposed by Harjunkoski et al. [41], has been referred to as HGPS in this dissertation, which uses the variable-fixing rule. The second method has been developed by ours and is referred to as HARDUC (hard-fixing unit commitment). Both methods are of type R&F.
- The improvement phase aims at finding better solutions from the initial solution. We have designed four adaptations of the local branching (LB) method proposed by Fischetti and Lodi [32] and one particularization of the kernel search (KS) based on

Angelelli et al. [5]. In total, five improvement models are developed.

The methods proposed are tested with three groups of instances constructed from the instances proposed by Kazarlis et al. [52]. The results were evaluated in comparison to those obtained by the CPLEX solver. A time limit was set for all methods. The results reveal that the off-the-shelf solver achieved faster optimization for small instances. However, as the instance sizes increased to medium and large, the proposed methods consistently outperformed the solver, producing superior results within the given time frame. Notably, the proposed methods always found a solution, while the solver failed to solve many instances within the given time frame in the medium and large categories.

## 6.1 MATHEURISTICS METHODS

### 6.1.1 CONSTRUCTION PHASE

In this phase, we propose a construction method called HARDUC (hard-fixing unit commitment) and compare it with the best UCP construction method known so far, presented by Harjunoski et al. [41].

Both methods are of the type R&F. The solution strategy of the R&F methods is to reduce the size of the MILP to a smaller, easier-to-solve one called sub-MILP, fixing those variables likely to keep their values in the optimal solution and letting the solver decide among the others. Finally, the sub-MILP is introduced into the solver, expecting that the solver returns a high-quality solution.

The first constructive method, proposed by Harjunoski et al. [41], will be referred to as HGPS (authors' initials) in this dissertation, which begins by solving a linear relaxation of the UCP, obtaining a solution  $\tilde{x}$ . In this method, the variables  $u_{g,t}$  to be fixed to one in the sub-MILP are those that satisfy the condition  $\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \geq \underline{P}$ ; we will call this condition: Harjunoski's rule. Let  $\tilde{u}_{g,t}$  and  $\tilde{p}_{g,t}$  are the values taken by the variables in the linear relaxation solution. Once the variables are fixed, the sub-MILP is solved, deciding over the rest of the non-fixed variables.

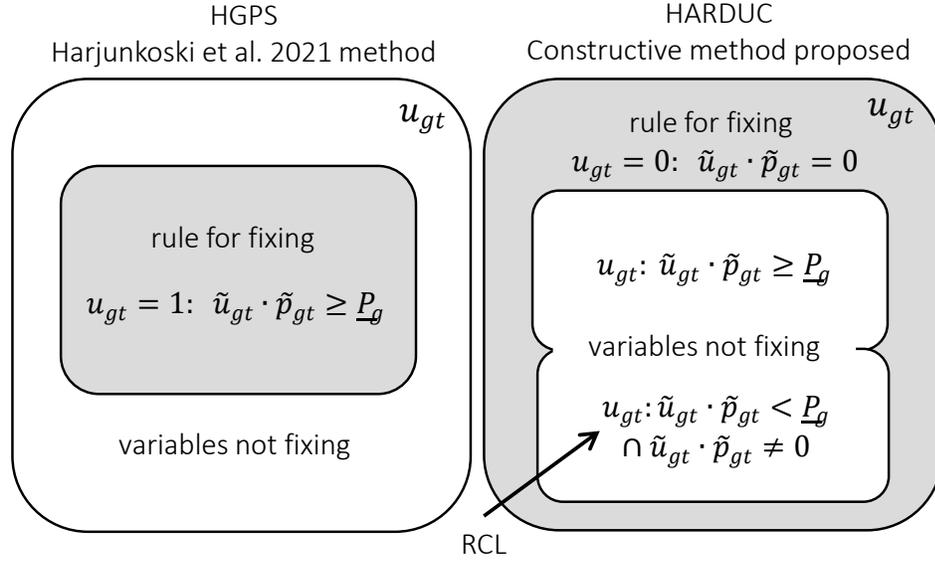


Figure 6.1: Rules for fixing construction methods; shaded area represents the subset of variables to be fixed in the sub-MILP.

HARDUC is the constructive method we propose. The method starts solving a linear relaxation of the UCP. Then, to form the sub-MILP, the variables  $u_{g,t}$  that satisfy the criterion  $\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} = 0$  are fixed to zero, and the decision over the other variables is left to the solver.

In Figure 6.1, the frames illustrate the space of variables  $u_{g,t}$ . On the left, we see the HGPS method, where the shaded region indicates variables set to one, and the light region shows variables yet to be determined by the sub-MILP solver. On the right, we have the HARDUC method, where the light region comprises non-zero variables according to Harjunkoski's rule but with a value greater than  $\underline{P}$ .

Finally, the variables that meet the criteria  $u_{g,t} : \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} < \underline{P}\} \cap \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \neq 0\}$  will be used for the construction of a restricted candidate list (RCL). We assume these variables are more likely to be in the optimal solution than those with zero values. This RCL is essential for other improving methods described in this chapter.

### 6.1.2 IMPROVING PHASE

After obtaining a feasible solution, the search for an improved solution commences, utilizing a set of five matheuristics. These include four versions based on LB and an additional version based on KS. Each matheuristic is fully assessed.

### 6.1.3 LOCAL BRANCHING

This matheuristic was proposed by Fischetti and Lodi [32] and implemented the concept of local search in a MILP problem using a branching strategy. Those feasible solutions within a distance radius of the parameter  $k$  define the neighborhood  $N(\bar{x}, k)$ . The distance from solution  $\bar{x}$  to other solutions is calculated using the hamming distance  $\Delta(x, \bar{x})$ , which counts the number of changes from 0 to 1 and from 1 to 0 between the variables that conform the binary support ( $BS$ ) of the solution  $\bar{x}$  to other variables outside of binary support ( $\overline{BS}$ ). The binary support of a solution  $\bar{x}$  consists of all binary variables that take the value of one in the solution. Exploration in a neighborhood is performed by the solver adding a non-valid inequality called local branching constraint (LBC) to problem  $P$ .

Unlike Fischetti and Lodi [32], which considers all binary variables of a problem to form a  $BS$ , in this dissertation, the  $BS$  of a solution  $\bar{x}$  of UCP is defined as the subset of the commitment variables  $BS = u_{g,t} : u_{g,t} = 1$ . Therefore, the LBC limits the number of moves between the  $BS$  and  $\overline{BS}$  to a number  $k$  as defined by the equation:

$$\Delta(x, \bar{x}) = \sum_{j \in BS} (1 - x_j) + \sum_{j \in \overline{BS}} x_j \leq k \quad (6.1)$$

That is, in our problem, the binary variables  $v_{g,t}$ ,  $w_{g,t}$ , and  $\delta_{g,t}$  contained in the UCP are not part of the  $BS$ ; therefore, they are not used in the local search definition we present. The decision to use only the binary commitment variables  $u_{g,t}$  is because we have observed that these variables  $u_{g,t}$  are found in most of the model constraints, and a change in the value of these variables can activate or deactivate all the operating constraints related to a generator. Using a dominant variable in the local search has advantages, such as a reduction in the solver search time due to reducing the number of variables in the local search. It also compacts the mathematical model by removing constraints related to

a generator. This idea of using a single dominant binary variable was initially reported by Darwiche et al. [21] and is used in his particular version of LB to solve a graph edit distance problem.

Figure 6.2 depicts the basic LB search tree. We start at node 1 with an initial solution  $\bar{x}^1$ . Assuming that the first solution is feasible, we build a left branch to node 2, adding the  $\Delta(x, \bar{x}^1) \geq k$  to the original problem  $P$ . Thus, the new problem  $P^2 \leftarrow P \cup \{\Delta(x, \bar{x}^1) \geq k\}$  is solved. Assuming that we have found the optimum of problem  $P^2$ , we can say that we have already explored the whole neighborhood defined by the  $\Delta(x, \bar{x}^1) \leq k$  finding as best solution  $\bar{x}^2$ . Now, leaving the current neighborhood and exploring other neighborhoods is necessary. To escape from a neighborhood, add to the problem  $P$  a constraint  $\Delta(x, \bar{x}^1) \geq k+1$  in the right branch of node 1. Note that this constraint complements the neighborhood already explored in node 2. This constraint will be kept in the subsequent problems and inherited by all child nodes of node 3; thus, the neighborhood defined by  $\Delta(x, \bar{x}^1) \leq k$  will be excluded from future searches. Now, we create node 4 and its left branch that explores a new neighborhood; the new problem is  $P^3 \leftarrow P \cup \{\Delta(x, \bar{x}^1) \geq k+1\} \cup \{\Delta(x, \bar{x}^2) \leq k\}$  finding the solution  $\bar{x}^3$  as the best in the neighborhood. Now, we will create a new right branch from node 3 with the complement constraint  $\Delta(x, \bar{x}^2) \geq k+1$ . Note that node 3 inherits the inequalities  $\Delta(x, \bar{x}^1) \geq k+1$  and  $\Delta(x, \bar{x}^2) \geq k+1$  from the right-hand side to child node 5. In this way, the method ensures avoiding visiting neighborhoods that have already been exhaustively explored. The process continues at node 5 by creating the left branch with the definition of the new neighborhood  $\Delta(x, \bar{x}^3) \leq k$  and solving the problem  $P^4 \leftarrow P \cup \{\Delta(x, \bar{x}^1) \geq k+1\} \cup \{\Delta(x, \bar{x}^2) \geq k+1\} \cup \{\Delta(x, \bar{x}^3) \leq k\}$ . The procedure is repeated until no better solution than the current one is found or, if necessary, until time runs out.

When the neighborhood exploration time limit is exceeded or when the problem is infeasible, LB uses diversification mechanisms that enhance or reduce the neighborhood's size by modifying  $k$ . Figure 6.2 shows that node 6 has reached its time limit, and no better solution has been found than the current one. As a result, the search continues by generating a new node 7 and reducing the neighborhood to  $\lceil k/2 \rceil$ . Another diversification strategy proposed by Fischetti and Lodi [32] is the addition of tabu constraints  $\Delta(x, \bar{x}^2) \geq 1$  to the problem that forces the solver to avoid the current solution where it is stuck.

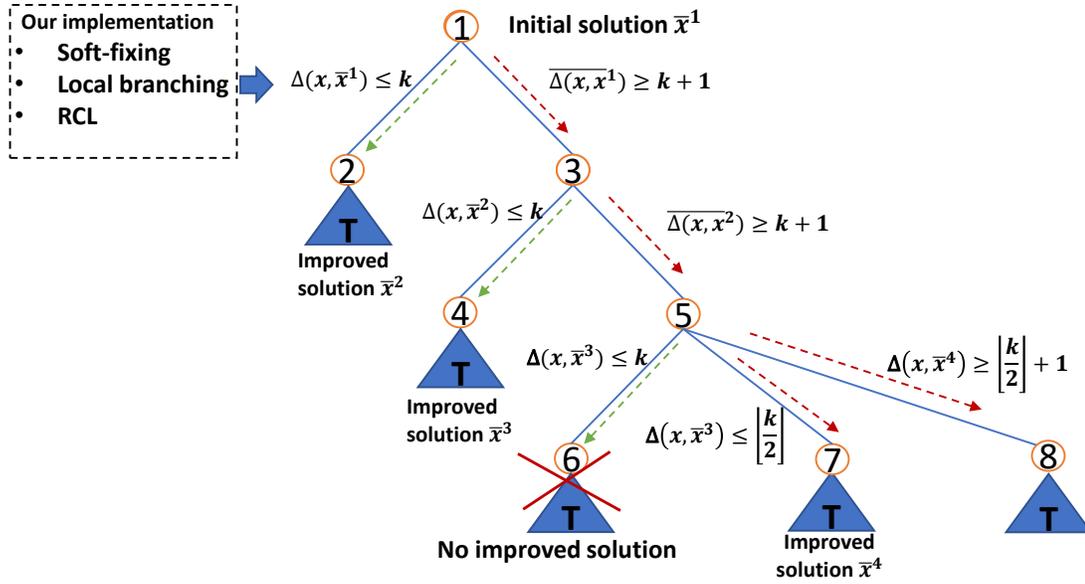


Figure 6.2: Representation of branching in the LB method, based on Fischetti and Lodi [32].

In the case that a maximum solution time is imposed on each node and the time runs out, two cases may arise: a) the first is that the solver finds a feasible solution better than the incumbent; this case does not ensure that the solution found is the best in the neighborhood, and we must continue the exploration with a new incumbent solution; b) if time runs out and no solution better than the incumbent is found, it is necessary to apply a diversification strategy such as those mentioned above of enhancing, reducing, or applying a tabu constraint that forces the solver to leave that solution.

Finally, the constraints representing the movements in LB are enlisted as follows.

$$\text{left-branch :} \quad \Delta(x, \bar{x}) \leq k \quad (6.2)$$

$$\text{right-branch :} \quad \Delta(x, \bar{x}) \geq k + 1 \quad (6.3)$$

$$\text{tabu :} \quad \Delta(x, \bar{x}) \geq 1 \quad (6.4)$$

$$\text{soft-fixing :} \quad \sum_{j \in BS} \bar{x}_j x_j \geq 0.9 \sum_{j \in BS} \bar{x}_j \quad (6.5)$$

### 6.1.3.1 IMPLEMENTATION OF LOCAL BRANCHING FOR THE UCP

**Algorithm 1** Function Locbra**Input:**  $k, t\_total, t\_node, iter\_max, \bar{x}, P$ =a formulation of the instance to solve**Output:** A feasible solution  $x^*$  of value  $z^*$  $rhs \leftarrow \infty; bestUB \leftarrow \infty; cutoff \leftarrow \infty; tl \leftarrow \infty$  $first \leftarrow \text{True}; diversify \leftarrow \text{False}$  $leftbranch \leftarrow \{\phi\}; rightbranch \leftarrow \{\phi\}; tabu \leftarrow \{\phi\}; softfixing \leftarrow \{\phi\}$  $iter \leftarrow 0$ **while** ( $elapsed\_time < t\_total$ ) **or** ( $iter < iter\_max$ ) **do****if** ( $rhs < \infty$ ) **then** $leftbranch \leftarrow \{\Delta(x, \bar{x}) \leq rhs\}$  $softfixing \leftarrow \left\{ \sum_{j \in SB} \bar{x}_j x_j \geq 0.9 \sum_{j \in SB} \bar{x}_j \right\}$ **end if** $tl \leftarrow \min(tl, t\_total - elapsed\_time)$  $P2 \leftarrow P \cup leftbranch \cup rightbranch \cup tabu \cup softfixing$  $\{stat, \tilde{x}, \tilde{z}\} \leftarrow \text{solve}(P2, tl, cutoff, first, \bar{x})$  $tl \leftarrow t\_node$ **if** ( $stat = \text{Optimal}$ ) **then****if** ( $rhs = +\infty$ ) **then****return** ( $x^*$ )**end if****if** ( $\tilde{z} < bestUB$ ) **then** $bestUB \leftarrow \tilde{z}; x^* \leftarrow \tilde{x}$ **end if** $rightbranch \leftarrow rightbranch \cup \{\Delta(x, \bar{x}) \geq rhs + 1\}$  $diversify \leftarrow \text{False}; first \leftarrow \text{False}; \bar{x} \leftarrow \tilde{x}; cutoff \leftarrow \tilde{z}; rhs \leftarrow k$ **end if****if** ( $stat = \text{Infeasible}$ ) **then****if** ( $rhs \geq +\infty$ ) **then****return** (Infeasible problem)**end if** $rightbranch \leftarrow rightbranch \cup \{\Delta(x, \bar{x}) \geq rhs + 1\}$ **if** ( $diversify = \text{True}$ ) **then** $cutoff \leftarrow \infty; tl \leftarrow \infty; first \leftarrow \text{True}; iter \leftarrow iter + 1$ **end if** $rhs \leftarrow rhs + \lceil k/2 \rceil; diversify \leftarrow \text{True}$ **end if****if** ( $stat = \text{Timeout with a feasible solution}$ ) **then****if** ( $rhs < \infty$  and  $first = \text{False}$ ) **then** $tabu \leftarrow \{\Delta(x, \bar{x}) \geq 1\}$ **end if****if** ( $\tilde{z} < bestUB$ ) **then** $bestUB \leftarrow \tilde{z}; x^* \leftarrow \tilde{x}$ **end if** $first \leftarrow \text{False}; diversify \leftarrow \text{False}; \bar{x} \leftarrow \tilde{x}; cutoff \leftarrow \tilde{z}; rhs \leftarrow k$ **end if****if** ( $stat = \text{No solution found}$ ) **then****if** ( $diversify = \text{True}$ ) **then** $tabu \leftarrow \{\Delta(x, \bar{x}) \geq 1\}$  $cutoff \leftarrow \infty; tl \leftarrow \infty; rhs \leftarrow rhs + \lceil k/2 \rceil; first \leftarrow \text{True}; iter \leftarrow iter + 1$ **else** $rhs \leftarrow rhs - \lceil k/2 \rceil$ **end if** $diversify \leftarrow \text{True}$ **end if****end while****return**  $x^*$

In this section, we define the implemented branching algorithm based on the original version of Fischetti and Lodi [32]. The pseudocode can be found in Algorithm 1 called **Locbra**.

The inputs of the algorithm **Locbra** are the MILP ( $P$ ), the neighborhood size ( $k$ ), total time ( $t_{total}$ ), the maximum time per node ( $t_{node}$ ), the maximum number of iterations ( $iter_{max}$ ), and an initial solution ( $\bar{x}$ ). The outputs are  $x^*$  and  $z^*$ , the incumbent solution and its cost.

The function `solve( $P, tl, cutoff, first, \bar{x}$ )` calls the solver, receiving as parameters:  $P$  represents a problem;  $tl$  is the maximum solution time;  $cutoff$  is the upper bound value;  $first$  is a boolean flag to tell the solver to stop the search as soon as the first solution is found;  $\bar{x}$  is an initial solution;

The arrays `leftbranch` and `rightbranch` contain the indices ( $g, t$ ) of the variables  $u_{g,t}$  that constitute the inequalities (6.3) or (6.2) added as constraints to problem  $P$ . Also, we use tabu constraints (6.4) in the method as a diversification strategy; these constraints are incorporated into problem  $P$  to prevent revisiting a previously explored solution  $\bar{x}$  during the search process.

Soft-fixing is a narrowing strategy also reported by Fischetti and Lodi [32] that fixes a considerable number of variables in a solution  $\bar{x}$  without losing the possibility of finding other good feasible solutions. The method adds the soft-fixing constraint (6.5) that enforces keeping at least 90% of the variables in 1 from the first solution  $\bar{x}$  and gives flexibility to the solver to decide which variables to keep. In this dissertation, we combined this soft-fixing strategy with the left-branch constraint (6.2) to modify only 10% of the variables into *BS*. Furthermore, we have relaxed the integrality constraint of the variables  $u_{g,t}$  but keeping their bounds into the range of  $[0,1]$ .

**Locbra** is an iterative algorithm that calls `solve()` several times to find the solution to a problem  $P$ ; depending on the result of `solve()`, its parameters are modified for the next iteration. In the first iteration, the method checks if the initial solution  $\bar{x}$  provided by the constructive method is the global optimum of the problem; the solver stops as soon as it finds the first solution ( $first \leftarrow \text{True}$ ). If the solution of  $\tilde{x}$  is  $stat = \text{Optimal}$ , we have found the optimal global; therefore, the method terminates; otherwise, the iterative

process continues modifying the MILP and the solver parameters according to one of the four solution states: Optimal, Infeasible, Timeout with a feasible solution, and No solution found.

**Optimal:** in this case, we have explored the whole neighborhood and found the best solution, so we have likely fallen into a local optimum. Now we need a diversification strategy to escape from this local optimum. It is also in our best interest that, in subsequent iterations, we not explore this neighborhood again. The strategy consists of adding a right-branch constraint (6.3), which complements the neighborhood that has been explored. Finally, we update the current and incumbent solutions  $\bar{x} \leftarrow x^*$ ,  $\tilde{x} \leftarrow \tilde{x}$ ; as well as *cutoff* to the objective value  $\bar{z}$ , the flags *diversify*  $\leftarrow$  False and *first*  $\leftarrow$  False, and the neighborhood size return to the original size,  $rhs \leftarrow k$ .

**Infeasible:** The cause of the infeasibility of the problem may be due to one or more constraints conflicting in problem  $P2$ , or a solution below the upper limit (cutoff) could not be found. From the second iteration onwards, infeasibility is handled by constructing a tabu constraint (6.4) added to problem  $P$ ; the tabu constraint enforces that the solution  $\bar{x}$  is omitted in the following search. To avoid cutting off the optimal solution in subsequent searches, replacing the last tabu constraint with the new one if it already exists is important. Otherwise, the inequality may be inherited by child nodes. If the *diversify* flag is True, then we will apply a diversification strategy, which consists of increasing the size of the neighborhood to  $rhs \leftarrow rhs + \lceil k/2 \rceil$ . In addition, we will update the *cutoff* to infinity and change the flag *first*  $\leftarrow$  True, which will stop the search process as soon as the solver finds the first feasible solution.

**Timeout with a feasible solution:** In this case, a solution has been found; however, we are not sure that the solution is the best in the neighborhood because the neighborhood has yet to be fully explored; therefore, we must continue to explore the neighborhood. The strategy to continue the search consists of adding to problem  $P$  a left branch (6.2) with the new, improved solution  $\bar{x}$ . If the flag *first* equals False, we add a tabu constraint (6.4) as a diversification strategy; this strategy will help find a solution different from the current one and possibly worse than the previous one. If the found solution  $\tilde{z}$  is better than the incumbent solution *bestUB*, we update the parameters of *bestUB*  $\leftarrow \tilde{z}$ ; and store the incumbent solution  $x^* \leftarrow \tilde{x}$ . Also, reset the neighborhood size value to the original

parameter  $rhs \leftarrow k$  and set the flags of  $first \leftarrow \text{False}$  and  $diversify \leftarrow \text{False}$ . Finally, we update the current solution to  $\bar{x} \leftarrow \tilde{x}$ , and we update the  $cutoff \leftarrow \tilde{z}$ .

**No solution found:** the time is over, and at least one solution was not found; however, the infeasibility of the problem cannot be proven. Therefore, two diversification strategies are employed depending on the *diversify* flag: the first strategy consists of increasing the neighborhood to  $rhs \leftarrow rhs + \lceil k/2 \rceil$  and including a left branch (6.2) plus a tabu constraint to problem  $P$  and continuing the search from the last solution  $\bar{x}$ . The second strategy involves reducing the neighborhood to  $rhs \leftarrow rhs - \lceil k/2 \rceil$  and including a left branch (6.2) to problem  $P$  and continuing the search from the last solution  $\bar{x}$ . If the *diversify* flag is True then  $\infty \leftarrow cutoff$ , and set the flags of  $first \leftarrow \text{True}$  and  $diversify \leftarrow \text{True}$ .

The algorithm terminates when either the time limit stopping criterion or the user-defined maximum number of iterations has been reached.

This chapter outlines four versions of the LB that differ from the original proposed by Fischetti and Lodi [32]. The main differences are described as follows.

- LB1: First, this version narrows the local search between the  $BS$  and an RCL, which is defined from the elements satisfying the condition  $u_{g,t} : \{\tilde{u}_{g,t} \cdot p_{g,t} < \underline{P}\} \cap \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \neq 0\}$ ; note that the RCL contains the variables that are not zero but are below the minimum power value of the generator  $\bar{P}_g$ , these variables are discarded by Harjunkoski et al. [41] in his fixing criterion, but we have included them to construct our RCL. Second, the soft-fixing method further narrows the search, forcing at least 90% of the original  $BS$  variables to remain in the solution by applying the constraint  $\sum_{j \in BS} \bar{x}_j x_j \geq 0.9 \sum_{j \in BS} \bar{x}_j$  proposed by Fischetti and Lodi [32]. In addition, soft-fixing relaxes the integrality constraint on the  $BS$  variables but keeps the variables' bounds at  $[0,1]$ . Finally, we take advantage of the fact that the binary variables  $v_{g,t}$  and  $w_{g,t}$  in constraints (5.10) force  $u_{g,t}$  to take binary values, even if  $u_{g,t}$  is defined as continuous. The possibility of relaxing the integrality constraint of  $u_{g,t}$  and obtaining the same solution without being forcibly binary was visualized and reported by Morales-España et al. [64] and Morales-España et al. [66] in their works about the models T&C thermal UCP.

- LB2: This version is similar to LB1 except that it does not have soft-fixing; therefore, constraints (6.5) are not applied.
- LB3: This version is the closest to the original one proposed by Fischetti and Lodi [32]. This version does not do a local search with any RCL and is not narrowed down by soft-fixing. The local search is defined only between the  $BS$  and the variables that do not form the binary support  $\overline{BS}$ .
- LB4: This version is similar to LB1 except that the RCL is formed by the variables with negative values in the reduced costs. The reduced costs are calculated by fixing to 1 the variables that form the  $BS$  of the initial solution  $\bar{x}$  and solving the linear relaxation of the problem.

A key feature of our versions of LB is that they use only the dominant variable  $u_{g,t}$  to define the  $BS$ , unlike the generic LB of Fischetti and Lodi [32], where all binary variables participate in the local search. The adaptations to the original LB method are anticipated to expedite the search process, aiming to achieve a solution of increased accuracy.

#### 6.1.4 KERNEL KEARCH

The matheuristic method KS was initially developed by Angelelli et al. [5] to solve a multidimensional knapsack problem with outstanding results. We envision KS as a highly competitive method for solving the UCP, so it was chosen as another matheuristic besides the four LB versions.

The KS is divided into two phases, initialization, and expansion. During the initialization phase, the binary variables in problem  $P$  that are likely to take a value of one in the optimal solution are determined. This subset of variables is called the kernel (equivalent to the binary support in LB). The selection of the variables that form the kernel is usually those that take the value one in the linear relaxation.

Then, the remaining variables not part of the kernel are sorted according to some economic criterion (often the reduced costs of linear relaxation) and divided into small groups called buckets.

The expansion phase consists of solving the multiple sub-MILPs formed by each bucket concatenated one by one with the kernel, one at a time. This phase is called expansion because, in each iteration, bucket variables that take the value of one are picked and added to the kernel. The constraint  $\sum_{j \in B_i} x_j \geq 1$  must be added to the sub-MILP to enforce that at least one variable of the bucket takes the value one; because of this constraint, the kernel tends to increase in size with each iteration. For each iteration, the remaining bucket variables not part of the sub-MILP must be fixed to zero. The conventional KS terminates when each bucket has been resolved with the kernel or the allowed time is over [5].

#### 6.1.4.1 IMPLEMENTATION OF THE KS METHOD FOR THE UCP

In this chapter, we present a version of the KS that can be seen in Algorithm 2 called **kernelsearch**.

The inputs of algorithm **kernelsearch** are total time ( $t_{total}$ ), a feasible solution ( $\bar{x}$ ) of an instance of UCP to solve ( $P$ ). The outputs are  $x^*$  and  $z^*$ , respectively, the incumbent solution and its cost. The function **kernelsearch** starts the initialization phase by building the kernel from variables with  $u_{g,t} = 1$  in the feasible solution  $\bar{x}$ ; this feasible solution is obtained from the constructive method HARDUC. It is noted that only the dominant variable  $u_{g,t}$  is used for building both the kernel and the buckets. Note that our version of KS does not require kernel building. It is only used as an improvement method, as a constructive method determines the kernel.

The construction of the buckets begins by fixing to 1 the variables constituting the kernel  $u_{g,t} \in K$  and solving a linear relaxation of the problem  $P^{\text{fix}(K)}$ . Then the set  $U$  is constructed with the variables  $u_{g,t}$  that are not inside the kernel  $K$ . The number of buckets is calculated using Sturges's rule [81]:  $nbucks = 1 + 3.322 \ln(|U|)$ .

Then, the  $u_{g,t} \in U$  variables are sorted in descending order (since this is a minimization problem) according to the value of their reduced costs obtained from the linear relaxation problem  $P^{\text{fix}(K)}$ . The newly ordered set is denoted as  $U^{\text{desc}}$ .

The set of buckets of size  $nbucks$ , into which the variables of set  $U^{\text{desc}}$  is divided, is

**Algorithm 2** Function `kernelsearch`**Input:**  $\bar{x}$ =a feasible solution of  $P$ ,  $P$ =a formulation of the instance to solve,  $t_{total}$ **Output:** A feasible solution  $x^*$  of value  $z^*$ **while** ( $elapsed\_time < t_{total}$ ) **do**

{Initialization phase}

 $cutoff \leftarrow z^*$  $K \leftarrow$  kernel formed from the variables that fulfill  $u_{g,t} : u_{g,t} = 1 \in \bar{x}$  $P^K \leftarrow$  **fixing**(1, $K$ , $P$ ) fixing to 1 all the variables into kernel $\{\hat{x}, \hat{z}\} \leftarrow$  **solveLR**( $P^{\text{fix}(K)}$ ) $U \leftarrow$  select the variables that fulfill  $u_{g,t} \notin K$  $nbucks \leftarrow 1 + 3.322 \ln(|U|)$  $U^{\text{desc}} \leftarrow$  sort  $U$  in descending order using its reduced cost values $\{B_i\}_{i=1}^{nbucks} \leftarrow$  **buildbuckets**( $U^{\text{desc}}$ ,  $nbucks$ ) $tl \leftarrow (t_{total} - elapsed\_time)/n$ 

{Expansion phase}

**for**  $i = 1, nbucks$  **do** $P^{K \cup B_i} \leftarrow$  **fixing**(0, $\bar{B}_i$ , $P$ ) fixing to 0 all the variables of the other buckets $P^{K \cup B_i} \leftarrow P^{K \cup B_i} \cup \{\sum_{j \in B_i} x_j \geq 1\}$  A variable from the bucket is forced into entering the kernel. $\{stat, \tilde{x}, \tilde{z}\} \leftarrow$  **solve**( $P^{K \cup B_i}$ ,  $cutoff$ ,  $tl$ )**if**  $stat = \text{feasible}$  **then****if** ( $\tilde{z} < z^*$ ) **then** $cutoff \leftarrow z^* \leftarrow \tilde{z}; x^* \leftarrow \tilde{x}$ **end if** $K \leftarrow$  update the kernel from the buckets variables that fulfill  $u_{g,t} : u_{g,t} = 1 \in \bar{x}$ **end if****end for****end while****return**  $x^*$ **Algorithm 3** Function `buildbuckets`**Input:**  $U^{\text{desc}}$ =list of  $u_{g,t}$  outside the kernel, in descending order,  $nbucks$ =number of buckets**Output:** A set of buckets  $\{B_i\}_{i=1}^{nbucks}$  $\{B_i\} \leftarrow \{\phi\}$  $start \leftarrow 0; end \leftarrow 0$  $n \leftarrow |U^{\text{desc}}|$  $k \leftarrow \lfloor n/nbucks \rfloor$  $remainder \leftarrow n \% nbucks$ **for**  $i = 1, nbucks$  **do** $end \leftarrow start + k$ **if**  $remainder > 0$  **then** $end \leftarrow end + 1$  $remainder \leftarrow remainder - 1$ **end if** $B_i \leftarrow B_i \cup \{U^{\text{desc}}[start : end]\}$  $start \leftarrow end$ **end for****return**  $\{B_i\}_{i=1}^{nbucks}$

denoted as  $B_i$ , where  $i$  is an index ranging from 1 to  $nbucks$ .

Function `buildbuckets`( $U^{\text{desc}}, nbucks$ ), which is located at Algorithm 3, is utilized to partition the set of variables  $u_{g,t} \in U^{\text{desc}}$  into approximately equal-sized subsets known as buckets  $B_i$ . First, the size of each bucket is calculated by dividing the total number of variables in  $U^{\text{desc}}$  by the desired number of buckets. Also, the remaining variables that cannot be equally distributed are identified. Next, an empty list is initialized to hold the buckets  $B_i$ , while two variables, *start* and *end*, are created to monitor each bucket's range of variables. Then, the function iterates over the number of buckets, sets the range of objects for each subset, and assigns any remainder variables to the first bucket. Next, the range of variables for each bucket is appended to the list  $B_i \leftarrow B_i \cup \{U^{\text{desc}}[start : end]\}$ . Lastly, returns all the buckets  $\{B_i\}_{i=1}^{nbucks}$  as the output.

During the expansion phase of the KS, problem  $P^{K \cup B_i}$  is solved by concatenating the kernel firstly with the buckets that contain variables having the most negatively reduced costs. Next, the other buckets' variables  $u_{g,t}$  are fixed to zero in the problem, and the constraint  $\sum_{j \in B_i} x_j \geq 1$  is added to ensure that at least one variable in the bucket  $B_i$  has a value of one. The function `solve`( $P, cutoff, tl$ ) is used to send the problem  $P^{K \cup B_i}$  to the solver, with  $P$  representing the problem, *cutoff* representing the upper bound value, and *tl* representing the maximum solution time. At each iteration, the kernel is updated, adding the buckets variables that result in a value of one in the solution.

After solving all the buckets  $B_i$  with the kernel, if the maximum time has not elapsed, the incumbent solution  $x^*$  is set as the new initial solution  $\bar{x}$ , and the process is restarted until the time is exhausted.

### 6.1.5 SUMMARY OF PROPOSED MATHEURISTICS

To conclude the described methods utilized, Table 6.1 summarizes the characteristics and distinctions among the matheuristics suggested in this dissertation.

Table 6.1: Summary of differences in improvement methods.

LB1	LBC built using dominant variable $u_{g,t}$ . RCL from Harjukovski's rule [41]. Soft-fixing to 90% of binary support.
LB2	LBC built using dominant variable $u_{g,t}$ . RCL from Harjukovski's rule [41].
LB3	LBC built using dominant variable $u_{g,t}$ .
LB4	LBC built using dominant variable $u_{g,t}$ . RCL from reduced costs. Soft-fixing to 90% of binary support.
KS	The first solution is given by the HARDUC constructive method. Kernel and buckets are built only with dominant variables $u_{g,t}$ . The number of buckets is calculated by Sturge's rule [81]. The buckets are built using the reduced costs from linear relaxation (fixing kernel).

## 6.2 EXPERIMENTAL WORK

Three tests were designed to assess the proposed matheuristics. This assessment includes a comparison with the CPLEX solver. These tests are defined as follows:

- Test 1: Assessment of the constructive methods.
- Test 2: Assessment of improvement methods under a running time limit of 4000 seconds.
- Test 3: Assessment of improvement methods under a running time limit of 7200 seconds.

Test 2, which has a time limit of 4000 seconds, approximately one hour, allows the LB methods to perform at least two iterations during their local search, thereby providing a more comprehensive evaluation.

The repository [https://github.com/urieliram/tc\\_uc](https://github.com/urieliram/tc_uc) contains the source code for the methods that have been implemented. All methods are coded in Python version 3.10.0

and Pyomo version 6.6.1. [42, 15]. Plots of results with Mathplotlib [45]. The hardware employed in this test was a 64-bit with 64GB of RAM with a 2.50 GHz Intel(R) i7(R) 11700 CPU, 65W, on a Linux Ubuntu version 20.0 operating system.

### 6.2.1 DESCRIPTION OF TEST INSTANCES

The number of instances to be solved is 83, with 20 obtained from Morales-España et al. [64]; the remaining 63 were constructed from the parameters of the eight generators reported in Table 5.1. Morales-España et al. [64] used the parameters of these eight generators to construct their instances.

The 63 new instances were constructed by gradually increasing the number of generators and combining them randomly. By increasing the number of generators, we also increased the complexity of the problem. This will allow us to assess the performance of the proposed methods and the off-the-shelf solver on instances of different sizes. The instances are divided into small, medium, and large sizes (`x7day_small`, `x7day_medium`, and `x7day_large`) according to the number of generators. There are 20 small instances with 28 to 81 generators, 33 medium instances with 85 to 156 generators, and 30 large instances with 165 to 405 generators.

Each point in Figure 6.3 corresponds to an instance that belongs to a group. The color of each point indicates the group, and the number of generators for each instance is displayed on the vertical axis. The planning horizon of the three groups of instances is 168 periods, equivalent to seven days. The demand profile is shown in Table 6.2. The demand for each instance is obtained by multiplying the demand profile by the sum of the maximum capacity of all generators. A reduction factor of 80% on a weekday is applied to calculate the weekend demand. The spinning reserve requirement of 5% of the power demand has to be met for each hour. The difficulty of these instances is well known, Kazarlis et al. [52] were the first to propose instances with these distinct characteristics; even reaching optimality gap values of less than 0.01% is challenging; for example, Morales-España et al. [64] used this type of instance to test the first thermal T&C UCP formulation. They established the optimality gap of 0.1% and 1.0% for the small and large instances. The

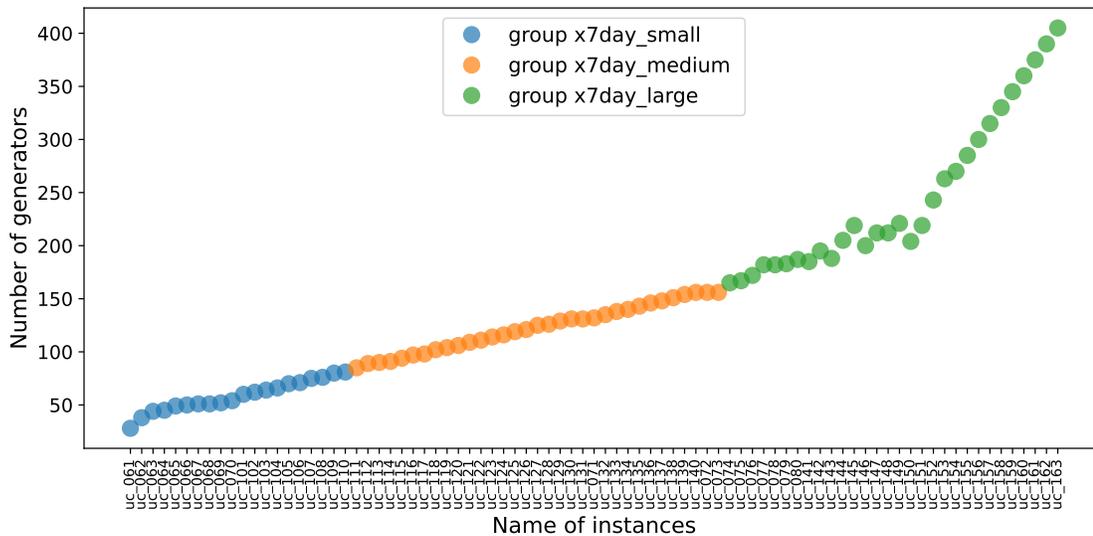


Figure 6.3: Increasing the number of generators in the three groups of instances: x7day small, x7day medium, and x7day large. Each point represents one instance.

Table 6.2: Load demand profile (% of total capacity).

hour	1	2	3	4	5	6	7	8	9	10	11	12
demand	71%	65%	62%	60%	58%	58%	60%	64%	73%	80%	82%	83%
hour	13	14	15	16	17	18	19	20	21	22	23	24
demand	82%	80%	79%	79%	83%	91%	90%	88%	85%	84%	79%	74%

Table 6.3: Groups of instances.

Group	Instances	Generators	Periods	Files uc_
x7day_small	20	28-81	168	061-070, 101-110
x7day_medium	33	85-156	168	071-073, 111-140
x7day_large	30	165-405	168	074-080, 132-163

solution time they report ranges from 1 to 10 hours solving the instances on a personal computer <sup>2</sup>. According to Morales-España et al. [64], the high difficulty of these instances results from combining eight generators to create large instances, leading to high symmetry.

Because these instances are considered more difficult to solve than others in the literature, we used them to test our methods. The summary of the characteristics of the three groups of instances we will use is shown in Table 6.3.

In Appendix A, the parameters of the 83 generators and the number of generators per instance are listed. Also, the instance files in JSON format are available in the repository [https://github.com/urieliram/tc\\_uc/tree/main/instances](https://github.com/urieliram/tc_uc/tree/main/instances) as well as the instance generator in [https://github.com/urieliram/tc\\_uc/blob/main/Instances\\_gen.ipynb](https://github.com/urieliram/tc_uc/blob/main/Instances_gen.ipynb).

### 6.2.2 TEST 1: ASSESSMENT OF CONSTRUCTIVE METHODS

This test aims to compare the number of feasible solutions obtained by the constructive methods HARDUC (the proposed method) and HGPS [41]. Additionally, for comparison purposes, we apply the CPLEX solver by setting a time limit and reporting the best integer feasible solution found within this time frame. We refer to this solution as CBS (CPLEX best solution). The CBS is used to verify whether the results obtained by HARDUC and HGPS are superior to those achieved solely by using the solver. All constructive methods have a maximum solution time of 1200 seconds.

All tests compare the accuracy of the feasible solutions regarding the relative optimality gap. The relative optimality gap is calculated for the three constructive methods HARDUC, HGPS, and CBS using the largest lower bound ( $LB$ ) obtained by the solution

<sup>2</sup>Quad-core Intel-i7 2.4-GHz personal computer with 4 GB of RAM, CPLEX 12.4 under GAMS

Table 6.4: Parameters of the solver used as constructive CBS.

Parameter	Value	Description
emphasis mip	1	Emphasize feasibility over optimality
mip strategy file	3	Node file on disk and compressed
mip tolerances mipgap	1E-5	Relative tolerance between the best integer and the best lower bound

of the CBS solver (if any) using the following equation [47]:

$$gap = \frac{|LB - z_{best}|}{((1E-10) + |z_{best}|)} \quad (6.6)$$

The solver parameters in constructive CBS mode have been tuned according to the values in Table 6.4.

The number of feasible solutions and percentages for each method and each group of instances is shown in Figure 6.4. The figure shows that as the instance difficulty increases, the number of solutions found by other methods decreases, while the proposed constructive method consistently discovers an initial solution within the specified time constraints.

The average and standard deviation of the relative optimality gap with the results of the feasible solutions obtained from the HARDUC, HGPS, and CBS constructive methods are calculated; the summary is shown in Table 6.5.

HARDUC, HGPS, and CBS methods have been run for the 63 instances `x7day_small`, `x7day_medium`, and `x7day_large`. The results indicate that our proposed HARDUC constructive method found all 20 feasible solutions of the `x7day_small` group, 33 feasible solutions of the `x7day_medium` group, and 30 feasible solutions of the `x7day_large` group. Equivalent to 100% of the solutions found in all groups of instances.

HGPS found 20 feasible solutions of 20 instances from the `x7day_small` group, 32 feasible solutions of 33 from the `x7day_medium` group, and 29 feasible solutions of 30 from the `x7day_large` group. Equivalent to finding 100.0%, 96.9%, and 96.6% of the solutions in each group.

Finally, the solver used as a constructive method CBS has found 20 feasible solutions from 20 instances of the group `x7day_small`, 26 feasible solutions from 33 instances of the

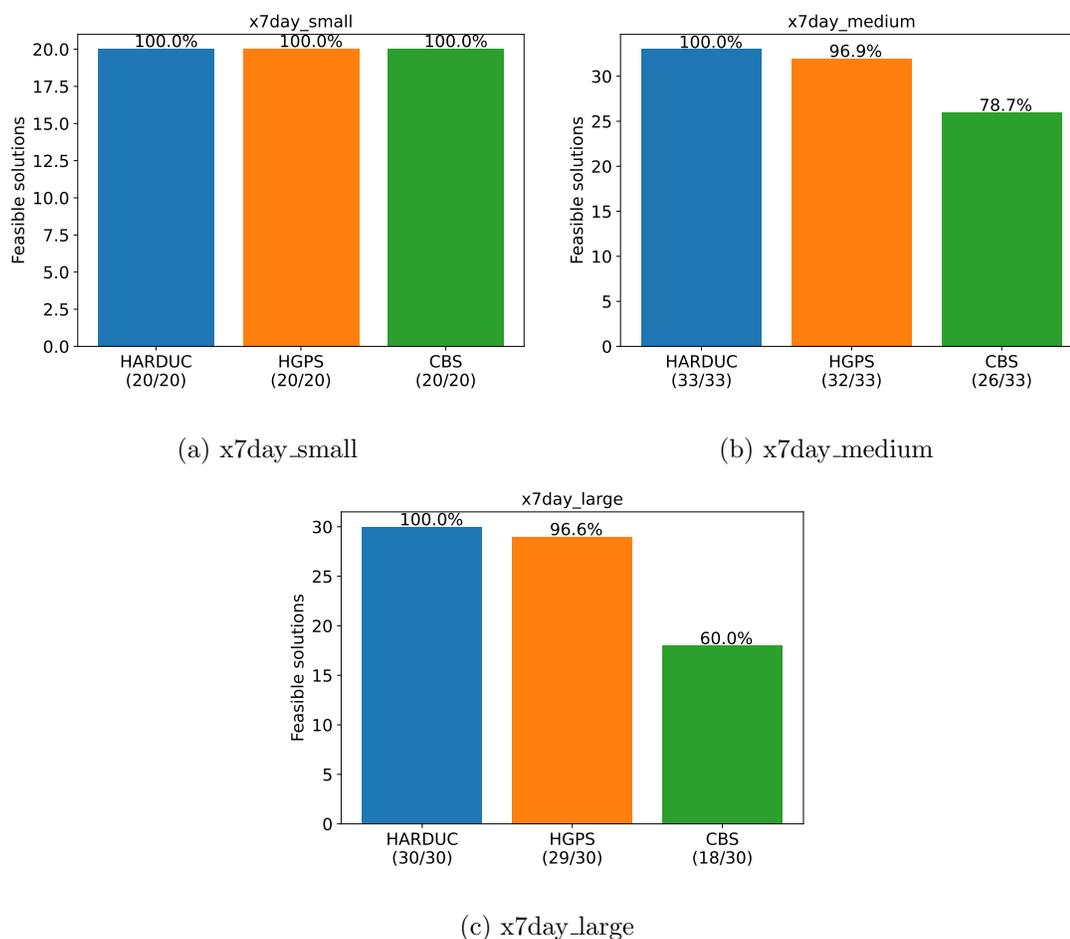


Figure 6.4: Rate of feasible solutions obtained by constructive methods under a running time limit of 1200 seconds.

group x7day\_medium, and 18 feasible solutions from 30 instances of the group x7day\_large. Equivalent to finding 100%, 78.7%, and 63.3% of the solutions in each group.

The distributions of the relative optimality gap results for each method are shown in Figures 6.5, 6.6, and 6.7. Each series represents the results of a method; the relative optimality gap is shown on the horizontal axis. The number of feasible instances found from each group's total instances is reported in parentheses. The figures illustrate that, as the instance difficulty increases, the relative optimality gap of the first solution obtained with the HARDUC method exhibits a distribution with significantly lower variance than other methods. Moreover, it demonstrates superior accuracy, as evidenced by the average relative optimality gap approaching zero. Statistical tests further support these findings.

Table 6.5: Descriptive statistics of constructive methods results.

Method	x7day_small			x7day_medium			x7day_large		
	AROG	ROGSD	NFS	AROG	ROGSD	NFS	AROG	ROGSD	NFS
CBS	0.0075	0.0195	20	0.0005	0.0001	26	0.0071	0.0186	18
HGPS	0.0013	0.0003	20	0.0010	0.0002	32	0.0009	0.0004	29
HARDUC	0.0007	0.0002	20	0.0005	0.0001	33	0.0003	0.0000	30

AROG: average relative optimality gap  
 ROGSD: relative optimality gap standard deviation  
 NFS: number of feasible solutions

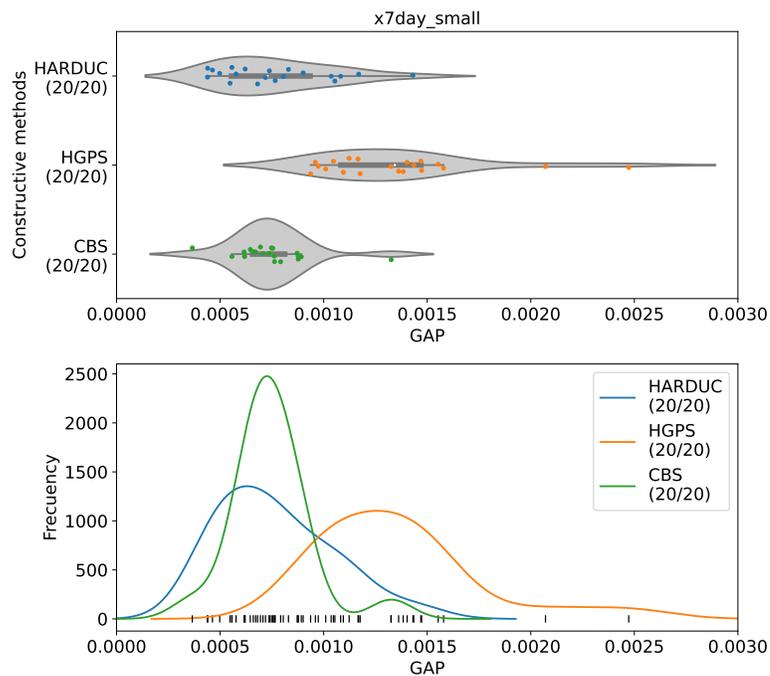


Figure 6.5: Relative optimality gap distributions of constructive methods in the instances group x7day\_small.

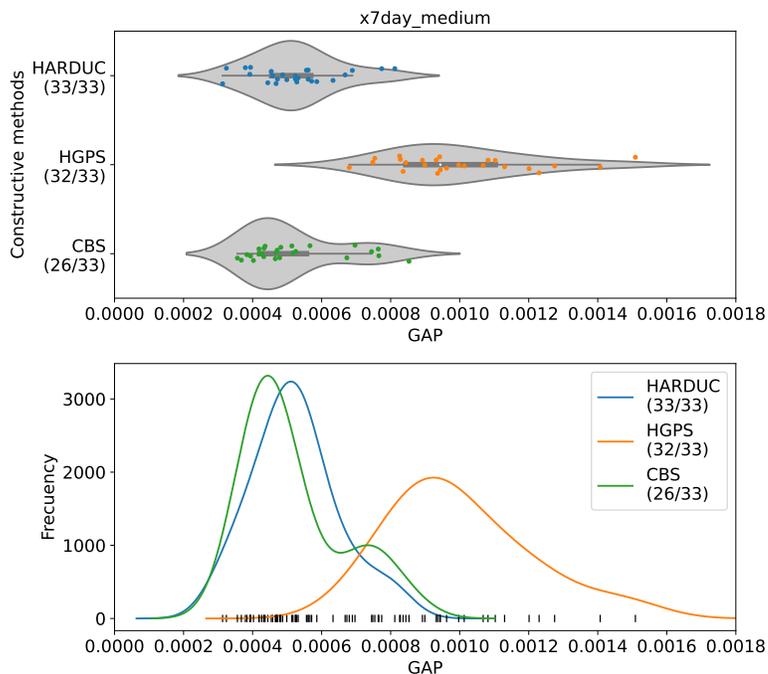


Figure 6.6: Relative optimality gap distributions of constructive methods in the instances group x7day\_medium.

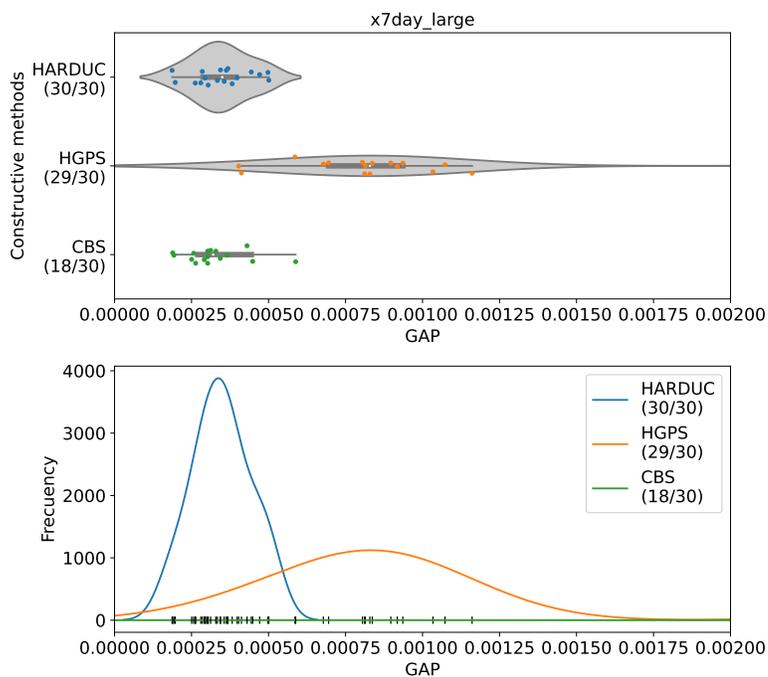


Figure 6.7: Relative optimality gap distributions of constructive methods in the instances group x7day\_large.

The following is an analysis of variance and mean difference analysis applied to the results of the methods regarding the relative optimality gap.

On one hand, the analysis of variance checks whether the average relative optimality gap of the results of all methods is statistically different from each other. On the other hand, a series of mean difference tests applied to each pair of results verify that the average relative optimality gap of one method is significantly higher than the mean obtained by another method. All statistical studies were done only in the instances where the solver found a feasible solution.

The general procedure of analysis of variance and mean difference is briefly described and applied to all tests carried out in this chapter.

- First, we carry out a normality test to decide if we use parametric or non-parametric tests to evaluate the results. The summary of the Shapiro-Wilk tests of `x7day_small`, `x7day_medium`, and `x7day_large` groups of instances can be found in Tables C.1, C.2, C.3, respectively.
- Second, an analysis of variance is performed to determine whether the differences between the results of the group's mean are statistically significant. We used ANOVA (parametric) or Kruskal-Wallis tests (not parametric) depending on the normality test results. The null hypothesis  $H_o$  assumes that the mean for each sample is equal, unlike the alternative hypothesis  $H_a$  which assumes that at least one of the means is different from the rest. Suppose the p-value is less than a significance level established at 0.05. In that case, we reject the null hypothesis and do not reject the alternative hypothesis that assumes that at least one means different from the rest. Otherwise, if the p-value is greater than or equal to 0.05, we fail to reject the null hypothesis  $H_o$ , which assumes that the means are equal. A significance level of 0.05 indicates a 5% risk of stating that a difference in the means exists when there is no actual difference.
- Third, a mean difference test is performed between the results of all methods comparing each pair of the relative optimality gap results of the methods. We used the T-test for two samples (parametric) or Mann-Whitney tests (not parametric)

depending on the normality test results. The null hypothesis  $H_o$  assumes that the sample means are equal, unlike the alternative hypothesis  $H_a$  which assumes a sample mean is greater than the other. For example, suppose the p-value is less than a significance level established at 0.05. In that case, we reject the null hypothesis and do not reject the alternative hypothesis that assumes that the means are different. Otherwise, if the p-value is greater than or equal to 0.05, we fail to reject the null hypothesis  $H_o$ , which assumes that the means are equal. A significance level of 0.05 indicates a 5% risk of stating that a difference in the means exists when there is no actual difference.

The results of the analysis of variance applied to the results of the constructive methods are shown in Table C.4. In the three groups of instances, the null hypothesis  $H_o$  of equality of means was rejected. The alternative hypothesis that at least one of the methods has a mean difference from the others was not rejected. The results of the mean difference analysis between the HARDUC and HGPS methods can be found in Table C.5. The HARDUC method has a significantly lower average relative optimality gap than HGPS.

Because the HARDUC method obtained a feasible initial solution in all test instances and a significantly smaller relative optimality gap than those obtained by HGPS and CBS, we consider the HARDUC constructive method to be the most appropriate to provide the first initial solution to the improvement methods (LB1, LB2, LB3, LB4, and KS) in the following tests.

### 6.2.3 TEST 2: ASSESSMENT OF IMPROVEMENT METHODS UNDER A RUNNING TIME LIMIT OF 4000 SECONDS

This test compares the accuracy of the relative optimality gap results obtained by the matheuristic methods LB1, LB2, LB3, LB4, and KS, in contrast to the best solution obtained by the solver under a running time limit of 4000 sec.

The solver was applied in two modes: Solver Method 1 (SM1) and Solver Method 2 (SM2). SM1 was executed without an initial feasible solution, while SM2 was executed from

Table 6.6: Parameters of the solver without initial feasible solution SM1.

Parameter	Value	Description
emphasis mip	1	Emphasize feasibility over optimality
mip strategy file	3	Node file on disk and compressed
mip tolerances mipgap	1E-5	Relative tolerance between the best integer and the best lower bound

an initial feasible solution provided by our constructive HARDUC. The solver parameters without an initial feasible solution SM1 have been tuned according to the values in Table 6.6. All improvement methods, including LB1, LB2, LB3, LB4, LB5, KS, and the solver SM2 using an initial solution, are initiated from the same solution. The objective is to assess whether the enhancements achieved in reducing the relative optimality gap through matheuristics differ from those solely by employing the solver. This would demonstrate that matheuristics results in a more significant improvement in the initial solution than the solver, even when starting from the same initial solution.

The relative optimality gap set as the optimality criterion is  $1 \times 10^{-5}$ . This small value has been chosen with the objective that the methods exhaust the running time and return the best solution reached in the allotted time. At the beginning of this chapter, the importance for electricity markets of approaching a zero optimality gap in the UCP was already stated [80, 48].

Methods LB1, LB2, LB3, LB4, and KS are given a total running time limit of 4000 seconds. This time covers the search for the first feasible solution and allows at least two iterations of the LB based methods to be solved. Each iteration of LB has a time limit of 1200 seconds. The solver parameters for these improvement methods have been tuned to emphasize the search for feasible solutions over the search for optimal solutions [32]. They can be seen in Table 6.7. The same parameters have been used to tune the solver method SM2.1h using an initial solution.

It should be noted that the symmetry calculation was turned off in the **mip tolerances mipgap=0** presolve. It was observed in preliminary tests that the solver's time taken to identify and remove symmetries could exceed the maximum allowed running time of 7200 seconds without finding a feasible solution.

Table 6.7: Solver parameters used in the HARDUC, HGPS, LB1, LB2, LB3, LB4, KS, and SM2 methods.

Parameter	Value	Description
emphasis mip	1	Emphasize feasibility over optimality
mip strategy file	3	Node file on disk and compressed
mip strategy heuristicfreq	50	Apply the periodic heuristic at this frequency
mip tolerances mipgap	1E-5	Relative tolerance between the best integer and the best lower bound
preprocessing symmetry	0	Turn off symmetry breaking

To ensure that the solver terminates its execution in case of running out of RAM memory, we selected the flag of **mip strategy file=3**, which will alternatively write the solution tree to the hard disk.

Another relevant flag that has been selected is **mip strategy heuristicfreq=50**, which sets the frequency of the heuristic search for feasible solutions in the solver. This value was chosen based on preliminary tests in which it was observed that this value was adequate to accelerate the speed of searching for feasible solutions in our instances.

The average and standard deviation results of the relative optimality gap, for the five methods, for the three instance groups `x7day_small`, `x7day_medium`, and `x7day_large`, and with a solution time of 4000 seconds, approx one hour, are shown in Table 6.8. Finally, the results of the solver are also plotted, including SM1 without an initial feasible solution and SM2 with an initial feasible solution. Statistical results were made only with the instances where the MILP solver found a solution.

We show the distributions formed by each of the methods with the results of the relative optimality gap in Figures 6.8, 6.10, and 6.12, where each series represents a method, the horizontal axis represents the relative optimality gap. The figures illustrate that, as the instance difficulty increases, the relative optimality gap of the results obtained with the KS and LB methods exhibits a distribution with significantly lower variance than the solver (SM1, SM2). Moreover, the KS and LB methods demonstrate superior accuracy of results with an average relative optimality gap closer to zero than the solver (SM1, SM2). We carry out statistical tests by analyzing variance and comparing means following the procedure described in 6.2.2 to determine whether the improvements are significant. The

Table 6.8: Descriptive statistics of the relative optimality gap of improving methods under a running time limit of 4000 seconds.

Method	x7day_small			x7day_medium			x7day_large		
	AROG	ROGSD	NFS	AROG	ROGSD	NFS	AROG	ROGSD	NFS
LB1_1h	6.65E-4	2.20E-4	20	3.74E-4	9.08E-5	26	2.71E-4	6.42E-5	18
LB2_1h	6.56E-4	2.18E-4	20	3.66E-4	9.49E-5	26	2.81E-4	7.17E-5	18
LB3_1h	6.98E-4	2.44E-4	20	3.82E-4	9.44E-5	26	2.91E-4	7.36E-5	18
LB4_1h	6.60E-4	2.30E-4	20	3.76E-4	9.19E-5	26	2.90E-4	6.64E-5	18
KS_1h	6.85E-4	2.63E-4	20	3.10E-4	7.48E-5	26	2.25E-4	3.95E-5	18
SM1_1h	5.79E-4	1.59E-4	20	4.73E-4	1.18E-4	26	3.07E-4	8.77E-5	18
SM2_1h	6.28E-4	1.49E-4	20	3.74E-4	9.08E-5	26	3.33E-4	7.79E-5	18

AROG: average relative optimality gap  
ROGSD: relative optimality gap standard deviation  
NFS: number of feasible solutions

summary of the results of the analysis of variance for the three groups of instances can be found in Table C.6.

The analysis of variance results rejects the null hypothesis of equality of the means and does not reject the alternative hypothesis that at least one of the means differs from that of the rest of the methods.

The results of the mean comparison study among all the methods, for 4000 seconds of execution and for the three groups of instances are presented in Tables C.8 and C.10 and C.12. These results are widely discussed in Section 6.3.

Appendix B shows the figures representing the convergence in the solution of the instances of the three groups x7day\_small, x7day\_medium, and x7day\_large, on the horizontal axis is shown the time in seconds, and on the vertical axis the value of the objective function. Each series represents one of the methods used. The behavior of the solver is also plotted, represented as the SM1 label.

### 6.2.4 TEST 3: ASSESSMENT OF IMPROVEMENT METHODS UNDER A RUNNING TIME LIMIT OF 7200 SECONDS

The 4000 seconds test is designed for scenarios where the decision-maker has limited time to deliver a solution, such as scheduling the daily day-ahead market. However, in scenarios where the decision-maker has additional time, such as when scheduling a weekly program calculated every seven days, it is necessary to establish the effectiveness of our matheuristic methods compared to relying solely on the solver. This test determines if these methods can deliver a high-quality solution within the given time frame. Additionally, this test assesses whether the solver can match the solution achieved by the matheuristic methods, specifically designed to expedite the search process when provided with a slightly extended time frame of two hours.

This test compares the results regarding the relative optimality gap obtained by matheuristic methods LB1, LB2, LB3, LB4, and KS in contrast to the best solution obtained by the solver under a running time limit of 7200 seconds.

The parameters of the solver without an initial feasible solution SM1 have been set according to the values shown in Table 6.6. As in Test 2, we use the solver starting from a feasible solution, SM2; the first solution is provided by our HARDUC constructive method. The relative optimality gap set as the optimality criterion in all methods is  $1 \times 10^{-5}$  and is calculated using Equation (6.6).

The running time limit for all methods is 7200 seconds. This time covers the search for the first feasible solution and the improvement of the first solution. Each iteration of LB has a time limit of 1200 seconds. The solver parameters for these improvement methods have been tuned to emphasize the search for feasible solutions over the search for optimal solutions. They can be seen in Table 6.9. The same parameters have been used to tune the solver to an initial feasible solution, SM2.

The average and standard deviation results of the relative optimality gap of the five methods for the three instances `x7day_small`, `x7day_medium`, and `x7day_large` are shown in Table 6.10. In addition, the solver results are reported with no initial feasible solution (SM1) and with an initial feasible solution (SM2).

Table 6.9: Solver parameters used in the HARDUC, HGPS, LB1, LB2, LB3, LB4, KS, and SM2 methods.

Parameter	Value	Description
emphasis mip	1	Emphasize feasibility over optimality
mip strategy file	3	Node file on disk and compressed
mip strategy heuristicfreq	50	Apply the periodic heuristic at this frequency
mip tolerances mipgap	1E-5	Relative tolerance between the best integer and the best lower bound
preprocessing symmetry	0	Turn off symmetry breaking

Table 6.10: Descriptive statistics of the relative optimality gap of improving methods under a running time limit of 7200 seconds.

Method	x7day_small			x7day_medium			x7day_large		
	AROG	ROGSD	NFS	AROG	ROGSD	NFS	AROG	ROGSD	NFS
LB1	6.11E-4	1.82E-4	20	3.36E-4	7.25E-5	26	2.47E-4	5.75E-5	18
LB2	6.22E-4	2.00E-4	20	3.39E-4	7.67E-5	26	2.42E-4	5.82E-5	18
LB3	6.67E-4	2.31E-4	20	3.50E-4	6.46E-5	26	2.63E-4	6.44E-5	18
LB4	6.31E-4	2.06E-4	20	3.38E-4	6.91E-5	26	2.54E-4	6.30E-5	18
KS	6.78E-4	2.67E-4	20	2.98E-4	6.73E-5	26	2.08E-4	3.82E-5	18
SM1	5.56E-4	1.59E-4	20	3.13E-4	8.77E-5	26	2.72E-4	7.33E-5	18
SM2	5.81E-4	1.55E-4	20	4.45E-4	1.07E-4	26	3.19E-4	8.34E-5	18

AROG: average relative optimality gap

ROGSD: relative optimality gap standard deviation

NFS: number of feasible solutions

We show the distributions formed by each of the methods with the results of the relative optimality gap in Figures 6.9, 6.11, and 6.13, where each series represents a method, the horizontal axis represents the relative optimality gap. The figures illustrate that, as the instance difficulty increases, the results obtained with the KS method outperform the other methods (including the base on LB methods) in terms of relative optimality gap, showcasing a distribution with significantly lower variance. Moreover, the KS method demonstrates superior accuracy, as indicated by an average relative optimality gap closer to zero compared to the other methods. We carry out statistical testing to determine whether the improvements are significant by analyzing variance and comparing means following the procedure described in 6.2.2. The summary of the results of the analysis of variance for the three groups of instances can be found in Table C.7.

The analysis of variance results rejects the null hypothesis of equality of means and does not reject the alternative hypothesis that at least one of the means is different from that of the rest.

The results of the mean comparison study among all the methods with a maximum allowed running time of 7200 seconds and for the three groups of instances are shown in Tables C.9, C.11 and C.13. These results will be discussed further in the next section.

Figures 6.14a, 6.14a, and 6.14a depict the percentage of solutions found by the solver within the allowed running time of 7200 seconds.

The convergence of the methods can be observed in the figures presented in Appendix B, where the evolution in the solution of each instance of the three groups `x7day_small`, `x7day_medium`, and `x7day_large` is plotted. The horizontal axis of the figures represents time in seconds, and the vertical axis represents the value of the objective function. Each series in the figures represents one of the methods used, and the solver's behavior is represented as SM1.

### 6.3 DISCUSSION

Analyzing the results of the methods, we can discuss some general observations.

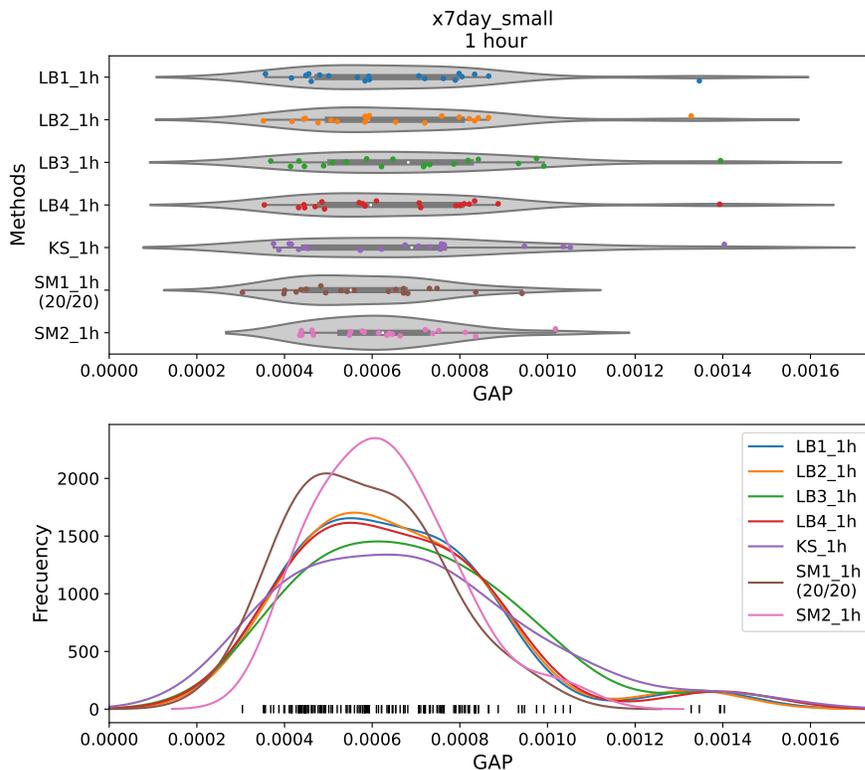


Figure 6.8: Relative optimality gap distribution of improvement methods of group x7day\_small under a running time limit of 4000 seconds.

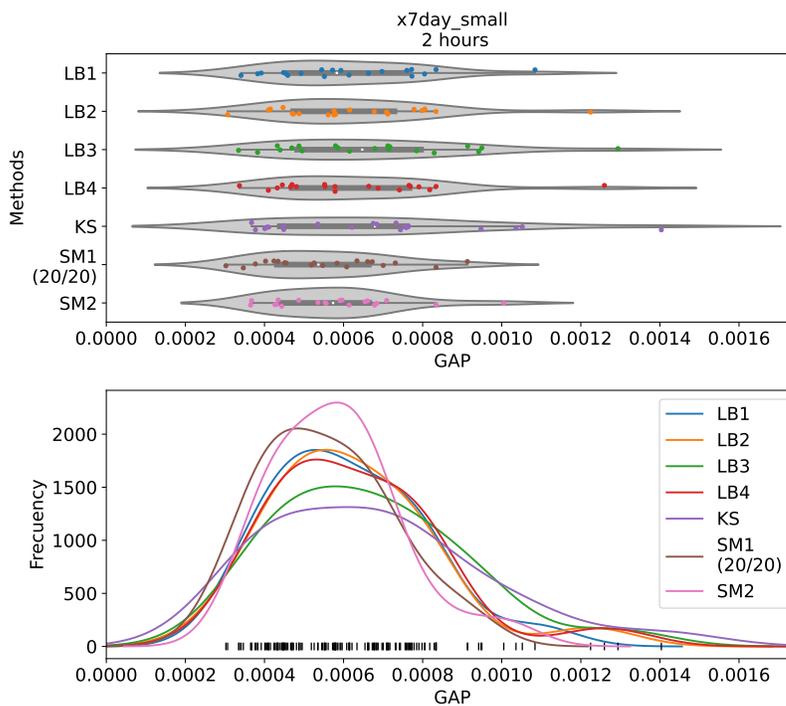


Figure 6.9: Relative optimality gap distribution of improvement methods of group x7day\_small under a running time limit of 7200 seconds.

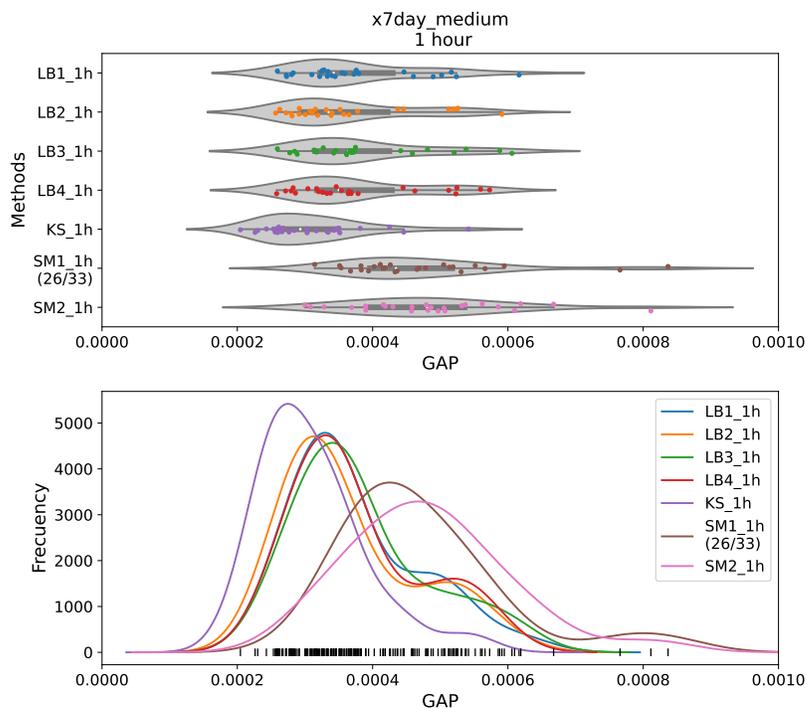


Figure 6.10: Relative optimality gap distribution of improvement methods of group x7day\_large under a running time limit of 4000 seconds.

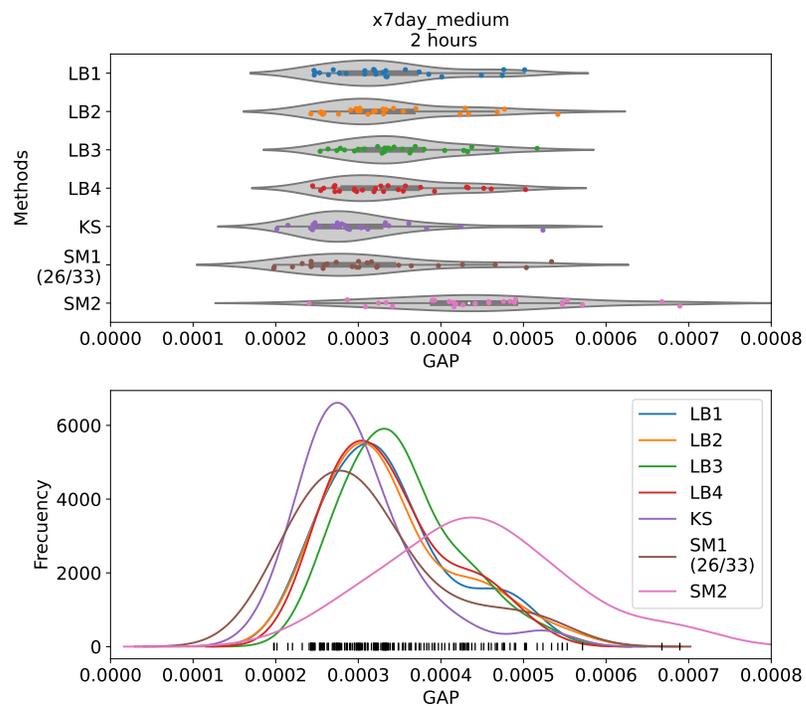


Figure 6.11: Relative optimality gap distribution of improvement methods of group x7day\_medium under a running time limit of 7200 seconds.

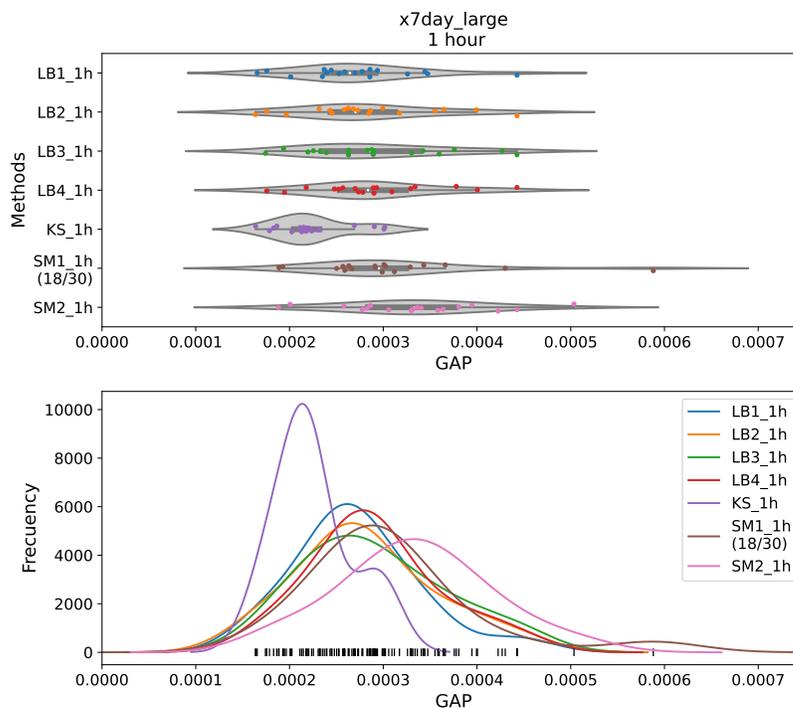


Figure 6.12: Relative optimality gap distribution of improvement methods of group x7day\_large under a running time limit of 4000 seconds.

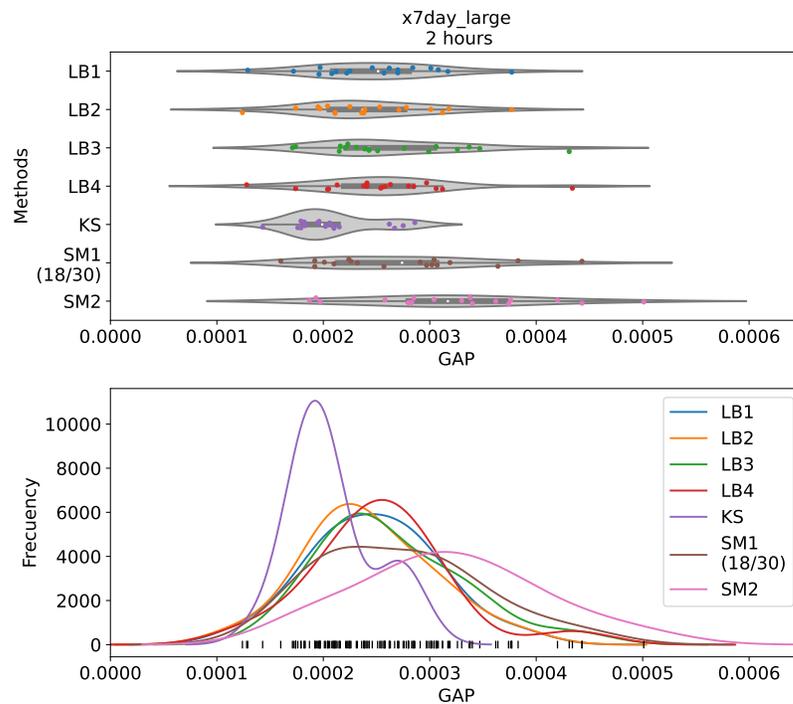


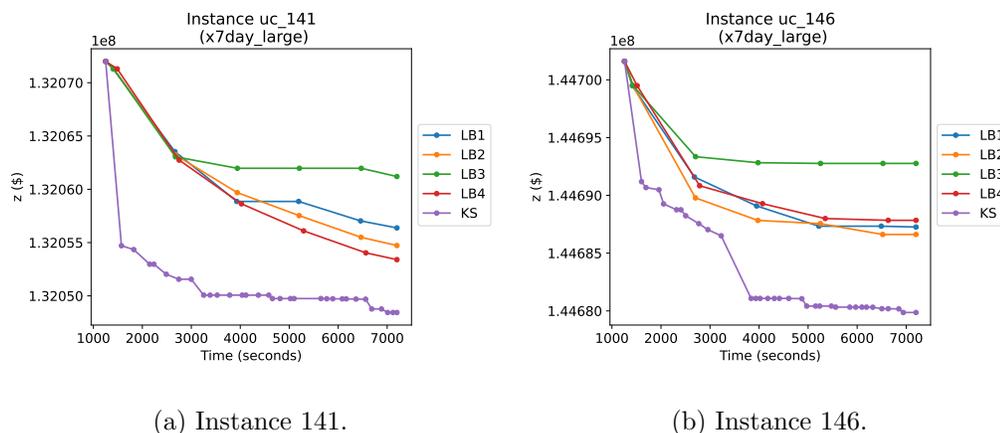
Figure 6.13: Relative optimality gap distribution of improvement methods of group x7day\_large under a running time limit of 7200 seconds.



Figure 6.14: Solution rate of each group of instances using the solver method (SM1) without a warm start solution.

Regarding the constructive methods, the HARDUC method, which found 100% of the feasible solutions in the three groups of instances and had better accuracy, can be considered the best alternative to provide a high-quality initial feasible solution. On the other hand, HGPS and CBS found only some feasible solutions compared to the other methods. Also, HGPS and CBS were less accurate than HARDUC.

The improvement matheuristic methods LB1, LB2, LB3, LB4, and KS found 100% of the solutions of the test instances, unlike the solver, which found only 78.8 % of the medium instances and 63.0% for the large instances, within a maximum allowed running time of 7200 seconds, as observed in Figures 6.14a 6.14b, and 6.14c. Therefore, we can confirm that the proposed methods are more efficient than using the solver without an initial solution.



(a) Instance 141.

(b) Instance 146.

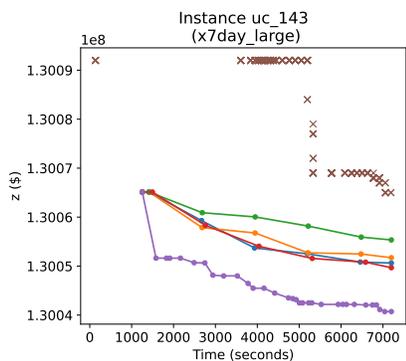
Figure 6.15: Comparison of methods on instances where the solver (SM1) could not solve within the allowed time limit.

Figures 6.15a and 6.15b depict large instances where the solver failed to find a solution within the expected time. It is worth noting that in these instances, the KS algorithm showed a higher rate of solution descent compared to the other methods based on LB. Nevertheless, when the solver did find a solution, its speed of finding solutions was noticeably slower than that of the other matheuristics methods, as illustrated in the Figures 6.16a and 6.16b.

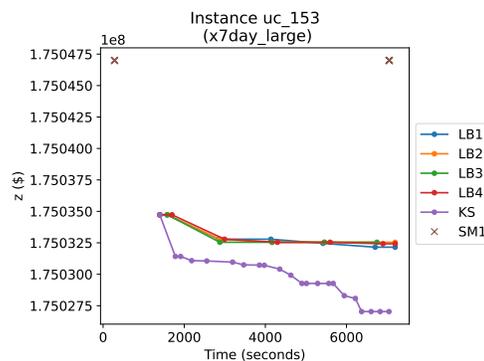
While there were some isolated large instances where the solver produced results that were on par with or even better than those obtained through matheuristics methods, it is important to note that such occurrences were not the dominant pattern. This observation becomes apparent when examining Figures 6.17a and 6.17b.

During the first iterations, KS was able to quickly find a solution with a fast reduction of the optimality gap. However, as the iterations continued, the convergence speed gradually decreased and eventually reached a point of “stagnation”, a common phenomenon in greedy algorithms. This phenomenon is clearly observable in the 6.18a and 6.18b instances.

In some instances, KS remained in the first solution without improvement; the percentage of instances remaining in the first solution is 13.3% of the large instances and 4.3% of the total instances. Figures 6.19a, 6.19b, 6.19c, and 6.19d represent the instances where the KS did not improve the first solution. Despite the fact that the methods based on

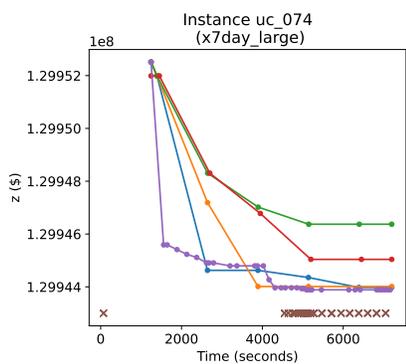


(a) Instance 143.

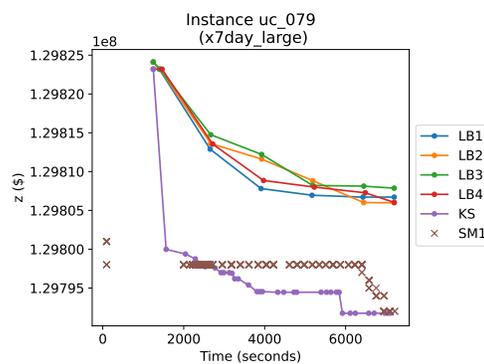


(b) Instance 153.

Figure 6.16: Comparison of methods on instances where the solver (SM1) found a solution but had significantly slower solution-finding speed compared to metaheuristics.

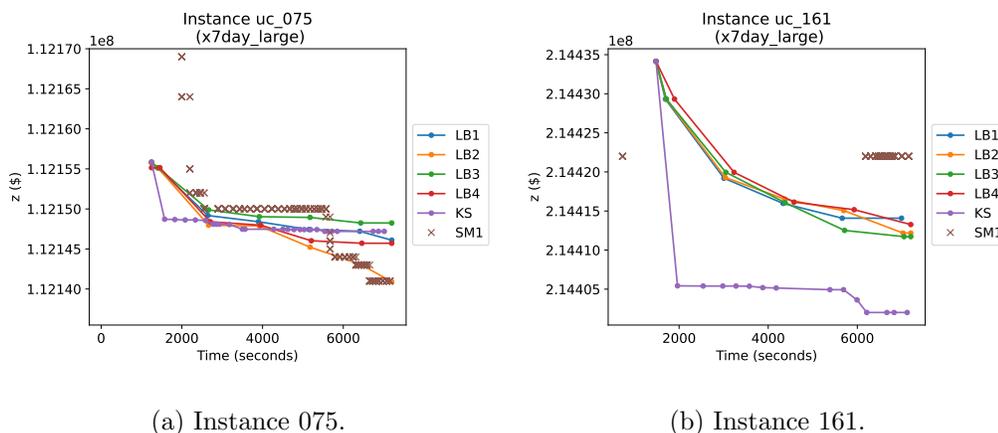


(a) Instance 074.



(b) Instance 079.

Figure 6.17: Comparison of methods on instances where the solver (SM1) produced results that were on par with or better than those obtained through the matheuristics.



(a) Instance 075.

(b) Instance 161.

Figure 6.18: Comparison of methods on instances where the KS exhibits a “stagnation” phenomenon.

LB are slower to converge than the method based on KS, they consistently improved the starting solution.

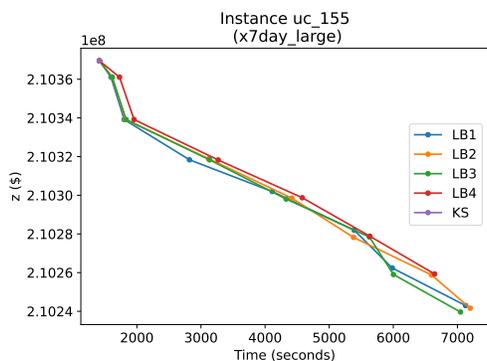
Here, a limited selection of examples has been presented to illustrate the behavior of the methods in specific instances. For a detailed visualization of the evolutionary progress in the convergence toward solutions for each instance, please consult the figures provided in Appendix B.

Observing that the solver could not find a feasible solution in some large instances, we intervened by providing it with a first starting solution obtained by our HARDUC method. However, even with the first solution, it was found that the solver performed less accurately than the proposed LB1, LB2, LB3, LB4, and KS improvement methods.

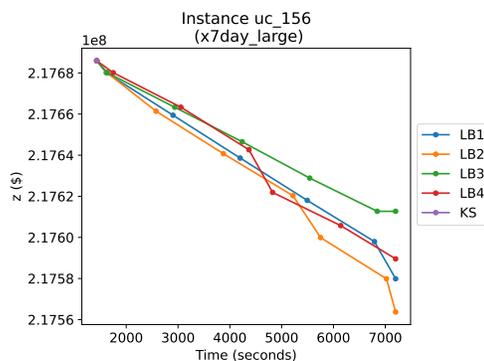
On the statistical results of the methods, we expose the following points.

Regarding the small instances and a maximum allowed running time of 4000 seconds, we can confirm that the solver can be used as an indistinct alternative to these methods since no significant differences were found in the accuracy of the LB1, LB2, LB3, LB4, and KS methods, and the solver.

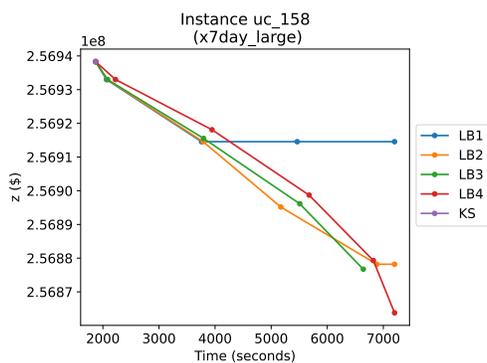
Regarding the small instances and a maximum allowed running time of 7200 seconds, differences were found in favor of the solver, which obtained a significantly smaller relative optimality gap than the LB3 method and the solver with the initial solution SM2.



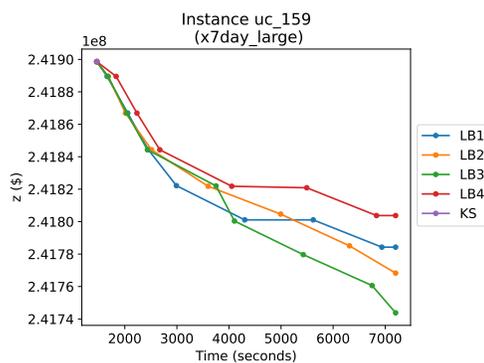
(a) Instance 155.



(b) Instance 156.



(c) Instance 158.



(d) Instance 159.

Figure 6.19: Comparison of methods on instances where the KS remained in the initial solution without improvement.

Regarding the medium instances and a maximum allowed running time of 4000 seconds, the KS method performed better than the other methods, obtaining a significantly smaller relative optimality gap than all the others. Also, the methods LB1, LB2, LB3, and LB4 performed better than the solver with a significantly smaller relative optimality gap than SM2\_1h and SM1\_1h.

Regarding the small instances and a maximum allowed running time of 7200 seconds, the results of the KS method are equal in accuracy by the solver without initial solution SM1, finding no significant differences between the results. We also observed that the LB3 method is again outperformed by the solver without an initial solution SM1, finding significant differences in favor of SM1. All methods were more accurate than the solver with the initial solution SM2.

Regarding large instances and a maximum allowed running time of 4000 seconds, KS performs better than the other methods, finding significant differences in favor of KS. Significant results were also found favoring LB1 over the solver without the initial solution SM1. We note that the solver with initial solution SM2 performed better than the solver without initial solution SM1.

Regarding large instances and a maximum allowed running time of 7200 seconds, KS performs better than all other methods, finding significant differences in favor of KS.

It was observed that all methods utilizing local branching (LB1, LB2, LB3, and LB4) outperformed the solver with an initial solution (SM2). However, it is noteworthy that within a 7200-second time frame, no significant differences were observed between LB1, LB2, LB3, LB4, and the solver without an initial solution (SM1). It is important to remember that these results were obtained based on the successful instances the solver could do. Therefore, it should be noted that the solver did not effectively solve all medium and large instances.

Furthermore, we observed a noticeably diminished solver performance when an initial solution was provided compared to when no initial solution was used.

In the ranking of the methods, we can observe that the KS is the method that obtained a lower average optimality gap than the other methods. Therefore, KS is the

best method for large instances within a maximum allowed running time of 4000 and 7200 seconds.

Among the proposed methods, LB3, which is the version most similar to the original method of Fischetti and Lodi [32], exhibits the poorest performance.

Finally, it should be noted that although the KS method generally achieved better results regarding the relative optimality gap, it did not the initial solution provided by HARDUC in approximately 4% of the instances. In contrast, the methods based on local branching successfully improved the initial solution.

## 6.4 CONCLUSIONS

Five methods were developed to solve a thermal UCP within a matheuristic approach, four based on the local branching method (LB1, LB2, LB3, and LB4) and one based on the kernel search (KS) method. In addition, a constructive method (HARDUC) was developed to provide the first solution to the matheuristic methods.

Among the methods based on local branching, LB3 is the closest version to the original local branching, unlike LB1, LB2, and LB4, which are variants that implement soft-fixing concept and a restricted candidate list (RCL). Additionally, only the variables classified as dominant are considered in the local search. In this case, the commitment variable, which determines the on or off state of the generator, was identified as such.

The methods were tested by solving three groups of instances classified into small, medium, and large according to the number of generators they contained. A time span of 4000 seconds to 7200 seconds was allocated for solving the instances using the LB1, LB2, LB3, LB3, LB4, and KS improvement methods, including the solver. The results of the methods were compared against the best solution achieved by the solver providing it with an initial feasible solution and without an initial feasible solution. The instances were created based on existing instances from the literature that are known to be challenging to solve.

The solver without an initial solution only solved approximately 79% of medium-

sized instances and 63% of large instances. In contrast to our methods, LB1, LB2, LB3, LB4, and KS always had a feasible initial solution provided by our HARDUC method in the constructive phase.

In terms of accuracy, measured in the relative optimality gap. Statistical tests showed no significant differences among the LB1, LB2, LB4, LB3, and KS methods and the solver on small instances.

For the case of testing with medium-sized instances within a maximum allowed running time of 4000 seconds, the KS performed the best. The other methods outperformed the solver with an initial solution and matched the results of the solver without an initial solution. For the tests with a maximum allowed running time of 7200 seconds, the solver performed equal to all methods except for LB3, which performed worse than the solver without an initial solution. LB3 is the method most similar to the original version of local branching.

Regardless of the time limit, all methods outperformed the solver regarding solution quality for the large instance set. Again the KS had the best performance.

Therefore, we can confirm that the proposed methods are more efficient than using only the solver; moreover, KS is the best method for medium and large instances. According to the results, the proposed adaptations (soft-fixing and RCL) to the original version of the local branching helped find better solutions only in medium instances.

While previous studies have suggested that the KS method effectively solves knapsack problems with promising outcomes, the results of our study demonstrate that the KS method may also be useful in solving the UCP problem.

The KS exhibits a behavior similar to a greedy algorithm, with a quick descent followed by a “stagnation” effect. On the other hand, local branching methods offer a constant improvement of the solution and the ability to avoid local optima, albeit with a slower descent rate than KS. A promising direction for future work would be to hybridize KS with LB to leverage the strengths of both methods.

We have learned that matheuristics work as “jumps” within the solution space and help speed up the search for better solutions than if we only used the solver.

We have found that the proposed improvement methods do their job by improving the initial solution. Therefore, we can consider our strategy of constructing and improving solutions using matheuristic methods to be effective.

Finally, in practice, we recommend using these mathematical methods and the solver simultaneously on different computers. This strategy of diversifying solution methods allows us to obtain always-in-time feasible solutions of high quality and choose the best solution.

## 6.5 FUTURE RESEARCH

- Develop a hybrid algorithm that combines the advantages of both the KS and LBC methods. For example, hybridization could leverage the fast convergence of KS in the initial iterations and the ability of LBC to escape from local optimal solutions. Then, when the KS solution becomes stagnant, the LB method can be used to escape the local optimal solution and go out to find a better solution.
- Develop a branch-and-cut method that incorporates the constraints of the local branching method. Callback functions can introduce local branching cuts into the solver's solution tree, saving time when loading the model into the solver. Additionally, by generating and introducing on-the-fly local branching constraints into the tree, matheuristic methods can solve the problem more efficiently, saving time in the process of reading the problem and preprocessing data by the solver.
- Test the proposed method with other UCP models. UCPs in electricity markets have different features depending on technical, operational, and economic factors. For example, some UCPs include disjoint constraints such as prohibited operative zones, nonlinear constraints such as hydraulic generation, elastic demand, and combined cycle plants with interdependent generators. Another important variation in the modeling is the consideration of network flow limits that further complicates the problem; this version is called network constraint UCP. Therefore, there is a significant potential to evaluate the effectiveness of the proposed method in UCP variations.

# A REAL-TIME LOAD FORECASTING METHOD

---

We have thoroughly addressed the UCP issue and considered it the computational core for scheduling the operation of electricity markets in the short-term. However, equally important is the demand forecast, which serves as one of the essential inputs for the UCP.

For example, in Mexico, short-term load forecast (STLF) and very short-term load forecast (VSTLF) are calculated daily for the scheduling processes. On the one hand, STLF calculated seven days in advance in hourly periods are used for the day-ahead market (MDA) and seven-day operational planning (AU-HE). On the other hand, VSTLF calculates two and a half hours ahead in fifteen-minute intervals, are used in the real-time market, specifically in the Real-time Unit Commitment (AU-TR) and Economic Dispatch with Multi-interval Network Constraints (DERS-MI) processes. The main features of the forecasts and their use in the electricity market in Mexico are shown in Table 7.1.

In this dissertation, VSTLF is understood as the forecast with a horizon ranging from a few minutes to a few hours, while real-time forecasting refers to the continuously calculated forecast using periodically refreshed data. The research of this chapter covers both concepts simultaneously.

Although unit commitment models and their solution methods can achieve very accurate results with negligible gaps, suppose the demand for which the operation is scheduled comes from a forecast with a low degree of accuracy. In that case, it can lead to significant

Table 7.1: Electricity demand forecasts in the Mexican market, based on [60].

Forecast	Market model	Deadline for submitting forecasts	Time horizon	Time interval	Update frequency
	Medium-term planning				
Short-term	AU-MDA AU-GC	10:00 a.m. daily	7 days	1 hour	Daily
VSTLF	AU-TR DERS-MI	before every quarter hour	2.5 hours	15 minutes (10 data)	Continuously every 15 minutes

deviations between generator scheduling and power system operation, with unexpected energy and reserve costs. Forecasts above the real demand can lead to higher costs for keeping generators running and extra reserve payments to compensate for deviations. Conversely, forecasts below actual demand may result in a last-minute purchase of power at a higher price. Another risk can be over-generation, which can jeopardize the system's security, which depends heavily on balance between generation and demand [92]. Therefore, having a quality forecasting method with high accuracy helps to maximize the economic benefits of the marking participants while preventing power system security problems.

In this chapter, we propose a demand forecasting method for a real-time electricity market; the method is called Analog Multiple with Moving Averages (AnMA). This method is divided into three phases: neighbor selection, calculation of a baseline forecast, and correction of the baseline forecast. The method is designed as a framework in which each phase can combine both statistical and machine learning tools. As a result, the AnMA method computes very short-term load forecasting (VSTLF) very fast and with a low computational cost. Moreover, it has proven to be sensitive to sudden load changes and excellent real-time bias adjustment.

## 7.1 REAL-TIME FORECASTING

Real-time electricity markets operate continuously 24 hours a day, every day of the year, collecting inputs, solving optimization problems (such as UCP), and delivering results within minutes. In addition, the results are published continuously. Therefore, the speed of the very short-term demand forecast (VSTLF) calculation is as important as its accuracy.

However, accuracy and computation time in currently used forecasting methods seem to be opposite attributes. Algorithms with high levels of accuracy may take many hours of training before they can be used; on the other hand, using simpler algorithms with little training may deliver results with low accuracy. The solution to this trade-off has been to increase the computational capacity of the equipment, involving high infrastructure and processing costs. Some forecasting models typically used are long-memory neural networks with high training times.

A real-time market forecasting (RTM) method demands more than just accuracy and computational speed. Real-time forecasting methods must meet other requirements, such as adaptability, low computational cost, robustness, and reproducibility. These requirements will be discussed in detail as follows.

- **Adaptability:** The method must consider corrections for sudden changes in demand due to external factors such as cold fronts, rain, or even network operations or failures. Therefore, the forecast must update its calculations with the most recent demand information coming in real-time.
- **Low computational cost:** This feature is fundamental when different forecasting methods are run simultaneously, or the forecasting method shares resources with other processes on the same equipment. A computationally cheap forecasting method has the benefit of saving on infrastructure investments and reducing the carbon footprint. They are also ideal when computing resources are limited.
- **Robustness:** The model must have a high convergence, always guaranteeing results for the market. Although some convergence problems are often caused by singular matrices or numerical instability and may be difficult to prevent, they must be handled in the programming in the best possible way, or methods should be implemented to mitigate their effects. It is generally observed that simpler methods are less prone to failure.
- **Low maintenance:** Models should be able to tune parameters automatically with minimal operator intervention. Simple algorithms with few parameters are preferred.
- **Reproducibility:** Forecasting methods should be deterministic with the ability to

obtain the same results even if run multiple times or on different equipment. The input data and the model parameters must be stored to repeat the calculation if required.

Dannecker [20] provides a comprehensive analysis of the forecasting needs for electricity markets, and we concur with certain aspects discussed in the article.

Real-time demand forecasting is not only utilized in market environments; it also benefits various other real-time applications, such as look-ahead applications. These applications aim to predict immediate system behavior and assist in operational decision-making. Additionally, real-time forecasting is employed in the economic dispatch of generation, transient stability analysis, online coordination between different generation sources (including hydraulic, wind, and solar sources), price setting, and interchange scheduling [28].

These methods, which look for patterns in the past similar to present behavior, are known as analogues (An). The idea was initially proposed by [61], who used it to predict errors in weather forecasts. However, the basic idea has been extended and used by several authors [9, 3, 4, 49]. To the best of our knowledge, the analogues method has yet to be used to forecast electricity demand nor has a moving average adjustment been implemented for real-time data.

The proposed method breaks the trade-off between accuracy and high computational cost, presenting a flexible environment that combines simple and complex statistical tools.

## 7.2 THE ANALOGUE MOVING AVERAGE METHOD

We propose the Analogue Moving Average (AnMA) method, a time series forecasting method based mainly on statistical components that consist of three phases:

- i) Selection of the neighbors
- ii) Calculation of the baseline forecast
- iii) Correction of the baseline forecast

PHASE I. SELECTION OF THE NEIGHBORS The load demand data set forms the time series  $S = (s_1, s_2, \dots, s_p)$  with  $p$  periods. A sub-sequence is a set of consecutive periods in a time series. The AnMA method starts by defining a sub-sequence of the latest data  $Y = \{y_j : j \in [|\mathcal{S}| - v_1, |\mathcal{S}|]\}$ , where  $v_1 \in \mathbf{N}$  is one of the most significant lags in the time series  $S$  from auto-correlation function (ACF) [14]. Then, the forecast horizon of  $Y'$  with size  $v_2 : v_2 < v_1$  is defined. Note that  $v_1$  should have a value greater than the forecast horizon, represented by  $v_2$ .

From the sub-sequence  $S' \subset S, S' = (s_1, s_2, \dots, s_n)$  where  $s_n = |\mathcal{S}| - (v_1 + v_2)$ , the set  $X$  of  $n$  sub-sequences of size  $v_1$  is extracted. We will refer to this  $X$  as a neighborhood and each one of its sub-sequences as a neighbor.

Afterward, the set of sub-sequences  $X_k : X_k \subset X$  represents the  $k$  neighbors having the highest correlation with  $Y$  based on a distance measure, i.e., the Pearson correlation coefficient. Those  $k$  neighbors are selected using the  $k$ -nearest neighbors ( $k$ -NN) method [59].

Figure 7.1 depicts the time series  $S$ , from which the  $k$  neighbors are extracted, which are the sub-sequences of  $S$  with the highest correlation with the latest data. Moreover, the subsequent data are displayed after the neighbors.

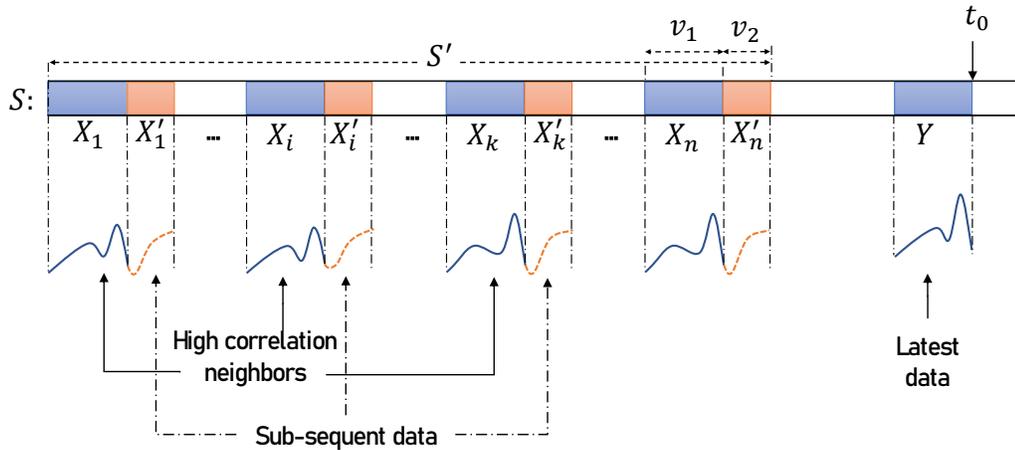


Figure 7.1: Phase of selection of neighbors, An method.

PHASE II. CALCULATION OF THE BASELINE FORECAST In this step, we use a regression model to explain the recent data of  $Y$  as a function of the neighbors  $X_k$ . In the basic version

of the AnMA method, we use the ordinal least squares (OLS) method with backward stepwise elimination. As a result, we obtain an adjusted regression model  $M$ .

Then, we obtained the set of sub-sequences  $X'_k$  that are the subsequent periods of a size  $v_2$  for each  $X_k$ . Finally, we input this set  $X'_k$  as data to model  $M$  to estimate the baseline forecast  $Y'$ .

Figure 7.2 depicts the regression phase, in which a regression model  $M$  is estimated with the latest data of the series  $Y$  as the independent variable and the data of the  $k$  selected neighbors  $X_k$  as the independent variables. The baseline forecast is the output of the  $M$  model with the sub-sequent of the neighbours data as input.

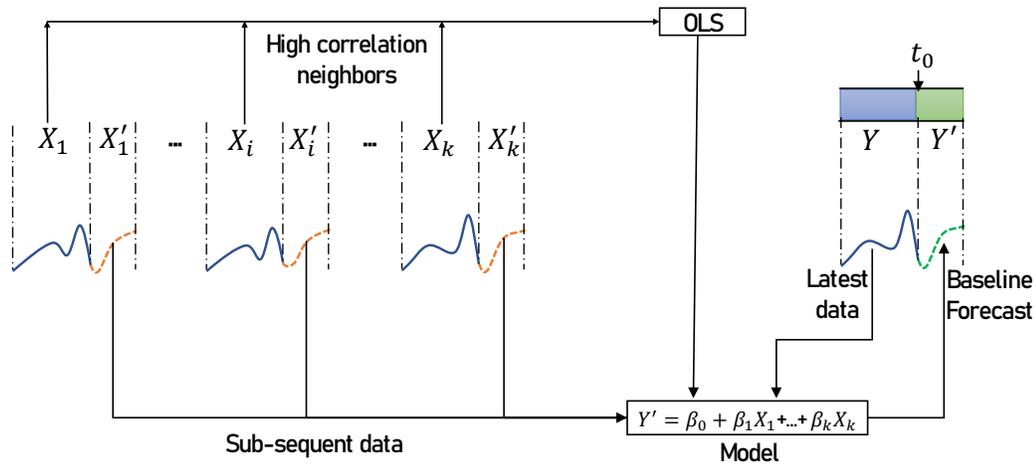


Figure 7.2: Phase of calculation of the baseline forecast.

Below in Figure 7.3 we show a actual example of selecting  $X_k$  neighbors with a high correlation with the latest  $Y$  data on the left side of the vertical line. The subsequent  $X'_k$  series and the  $Y'$  forecast are drawn on the right side of the vertical line. Note that there is no subsequent  $Y$  data on the right side of the figure; instead, the  $Y'$  forecast is drawn.

**PHASE III. CORRECTION OF THE BASELINE FORECAST** Because the An method is based on identifying and repeating patterns that have occurred in the past, it is susceptible to accumulating a bias over several periods before correcting itself. Therefore, the An method's baseline forecast ( $Y'$ ) must be compensated for its use with new data from real-time. This way, it is provided with an adaptive mechanism to correct bias and sudden

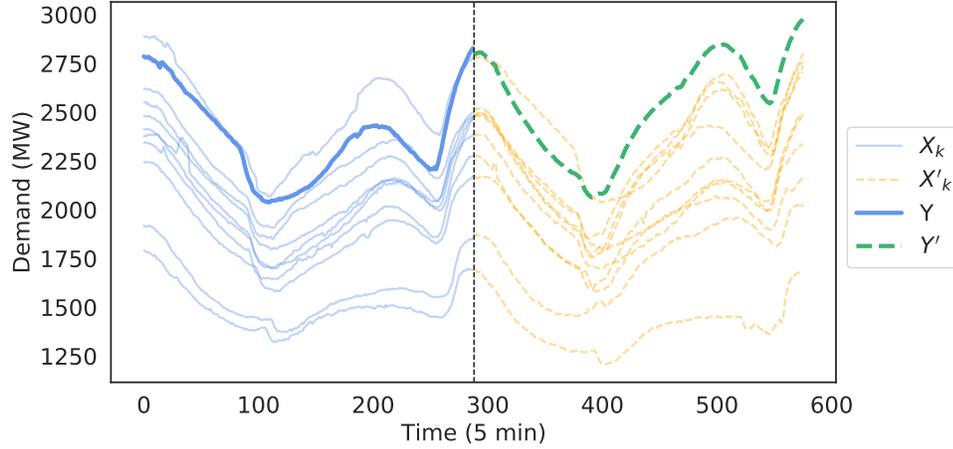


Figure 7.3: Left: neighbors  $X_k$  with highest correlation with the latest data  $Y$ . Right: subsequent data  $X'_k$  and the baseline forecast  $Y'$ .

changes in demand by attenuating the error.

The AnMA method generates a time series of the differences between the forecast ( $Y'$ ) and the actual demand ( $S$ ). This time series of differences, also known as the error series, can be mathematically represented as  $\epsilon = \epsilon_t : t \in [-v, 0]$ . The variable  $t$  in this equation represents the time period, with  $t = 0$  being the current time and  $t = -v$  being the time interval. This means that the error series contains the latest  $v$  errors of the An method. It's important to keep the baseline forecast results, as they will be used to correct future forecasts.

We use a moving average (MA) [14] model to calculate the following forecast errors  $\hat{\epsilon}$  with the stored latest data  $\epsilon$ .

Finally, combining the An and MA methods, we calculate the final forecast  $Y''$ , adding the  $\hat{\epsilon}$  to the result of the baseline forecast  $Y'$  as follows:  $Y'' = Y'(t) + \hat{\epsilon}(t), \forall t = 1, \dots, w$ .

Figure 7.4 depicts the phase of correction. First, we obtain an error series  $\epsilon$  by subtracting the latest actual data from the previous baseline forecast. Second, we compute the bias  $\hat{\epsilon}$  using an MA model. Third, we correct the baseline forecast error by adding the bias  $\hat{\epsilon}$ .

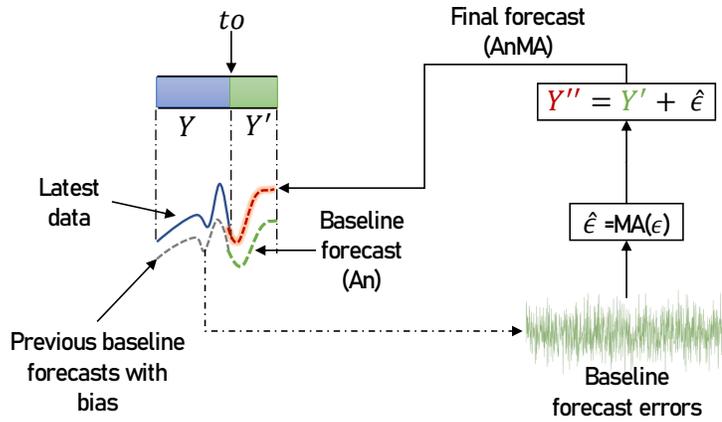


Figure 7.4: Phase of correction, MA method.

To provide a comprehensive overview of the AnMA method, Figure 7.5 merge the three phases of the method: i) Selection of neighbors, ii) Calculation of the baseline forecast, and iii) Correction of the baseline forecast.

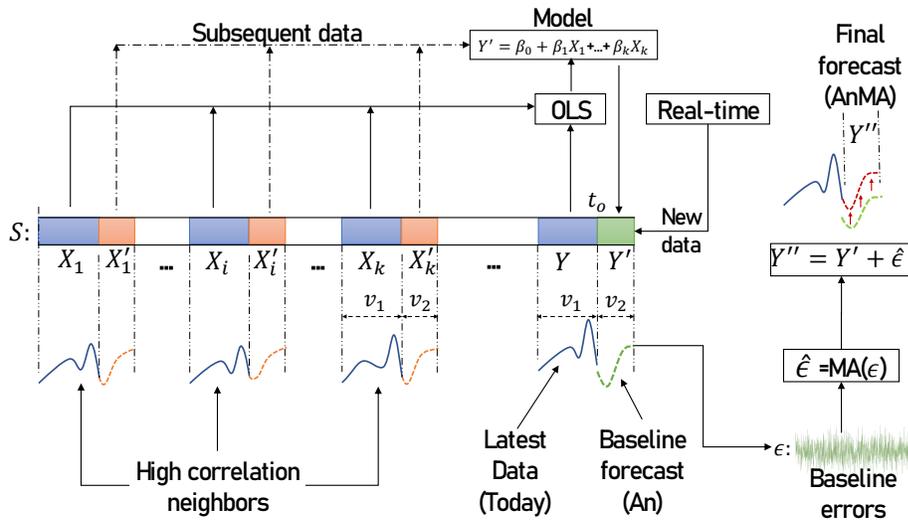


Figure 7.5: AnMA method.

### 7.2.1 ANMA VARIANTS

The basic version of the proposed AnMA method uses statistical tools such as OLS with stepwise as the regression model and the Pearson coefficient as the measure of distance between neighbors. However, other variations of the method can be implemented using other statistical and machine learning tools in the neighbor selection and calculation of

the baseline forecast phases.

The first variant can be performed in the neighbor selection phase by changing the distance measure between neighbors from the Pearson correlation coefficient to the Euclidean distance.

The second variant is performed at the calculation of the baseline forecast phase, applying tools such as principal component regression (PCR) and partial least squares (PLS), which are linear dimension reduction methods that keep a subset of the relevant variables (in our case, the neighbors) and discard the rest in the model. Other regression models are Lasso and Ridge, with small differences, which are shrinkage methods that impose a penalty on the regression coefficients for their size, minimizing the penalized residual sum of squares. Finally, we can use ensemble-based machine learning models such as Random Forest (RF), Bagging, and Boosting. All these machine-learning methods are based on ensembles. RF is a method that combines multiple decision trees to create a solution; Bagging is a method that combines multiple solutions of the same type of prediction to decrease variance. Boosting is a method that combines multiple types of predictions to decrease bias. These methods are broadly discussed by Hastie et al. [43].

### 7.3 EXPERIMENTAL WORK

The tests consist of calculating the demand forecast of a time series, using the proposed AnMA method and its variants against other benchmark methods that compete in accuracy and computational speed.

Two tests will be carried out:

- Calculation of the demand forecast for five minutes (one period).
- Calculation of the demand forecast for two-and-a-half hours (thirty periods).

The AnMA method's default version uses OLS with stepwise as the regression method and the Pearson coefficient as the distance metric during neighbor selection. However, other AnMA variants use different regression methods such as PCR, PLS, Lasso,

Ridge, RF, Boosting, and Bagging. Additionally, two neighbor selection metrics are used in the AnMA approach: Pearson's coefficient (the default) and Euclidean distance (indicated by the suffix `_euc`). The idea behind using these variants is to identify the best combination of regression model and selection metric that produces the most accurate results.

We have selected the additive Holt-Winters (HWA), multiplicative Holt-Winters (HWM), autoregressive moving average (ARMA), and persistent or naive methods as reference benchmarks to test the AnMA approach. These widely-used benchmarks possess important qualities such as adaptability, low computational cost, robustness, and reproducibility, which are essential for real-time forecasting in an electricity market. Our AnMA method will be compared to these benchmarks to determine its accuracy and promptness.

Accuracy metrics will be in terms of mean absolute percentage error (MAPE), and mean absolute error (MAE). We will also measure the CPU time of the models and particularly for AnMA the time of the neighbor selection, regression, and correction baseline; the time is measured in seconds.

As a validation method,  $k$ -fold cross-validation will be used in a particular setup for time series forecasting, same to the purpose of Bergmeir et al. [11]. However, we simulate real-time data by incorporating the latest data point while removing the oldest point using a first-in, last-out approach. This method of simulating real-time data ensures that the algorithm operates with the most up-to-date information available.

Figure 7.6 shows the time series data divided into training (blue) and testing (orange) periods for training and evaluating the method performance.

Also, the tests will analyze how effectively the MA error correction component reduces the error of the baseline forecast generated by An. To assess the MA correction's impact, we omit the MA correction in some variants of AnMA identified only by the prefix **An**. The test aims to demonstrate the significant effect of the baseline forecast correction approach that uses moving averages on the final forecast.

The AnMA method is coded in Python version 3.10.0 The scikit-learn and statsmodels libraries developed by Pedregosa et al. [74], Seabold and Perktold [82] in Python

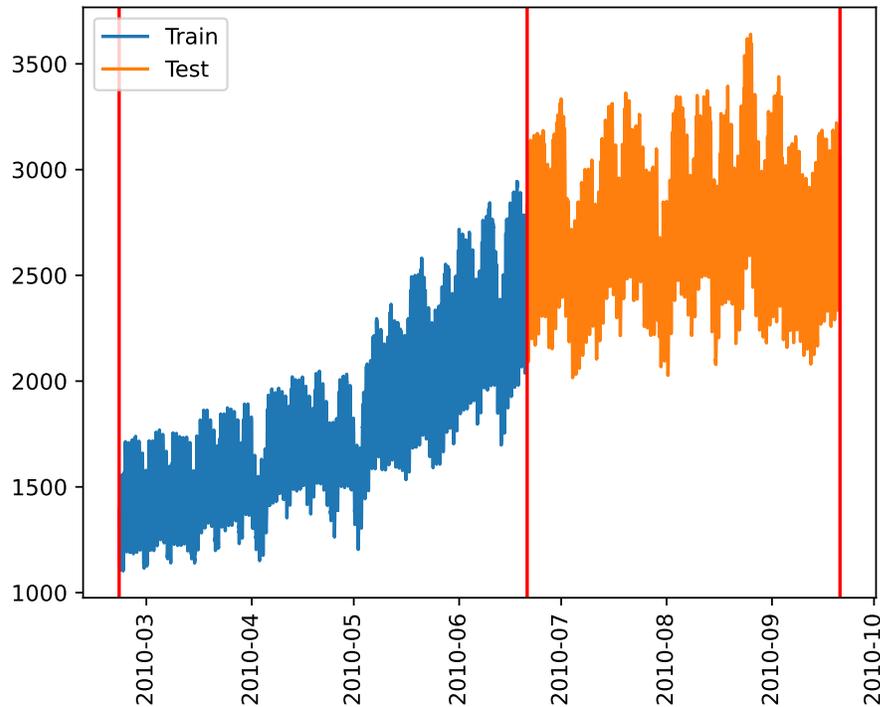


Figure 7.6: Training and testing periods for tests.

language were used. Plots of results with Matplotlib [45]. The source code of the AnMA method and data is available in the <https://github.com/urieliram/analog/blob/main/Analog3.ipynb> repository. The hardware employed was a 64-bit with 64GB of RAM with a 2.50 GHz Intel(R) i7(R) 11700 CPU,65W, on a Linux Ubuntu operating system.

### 7.3.1 DATASET

The tests use a time series of load demand from a typical region of Mexico. The time series plot in Figure 7.7 considers one year of load data with a sampling frequency of five minutes. As can be seen, the series presents seasonal patterns, i.e., the load increases during the warm seasons reaching approximately 2500 MW, and decreases during the cold seasons, with a load of approximately 1000 MW. The data are available in this repository <https://github.com/urieliram/analog/blob/main/data5min.csv>

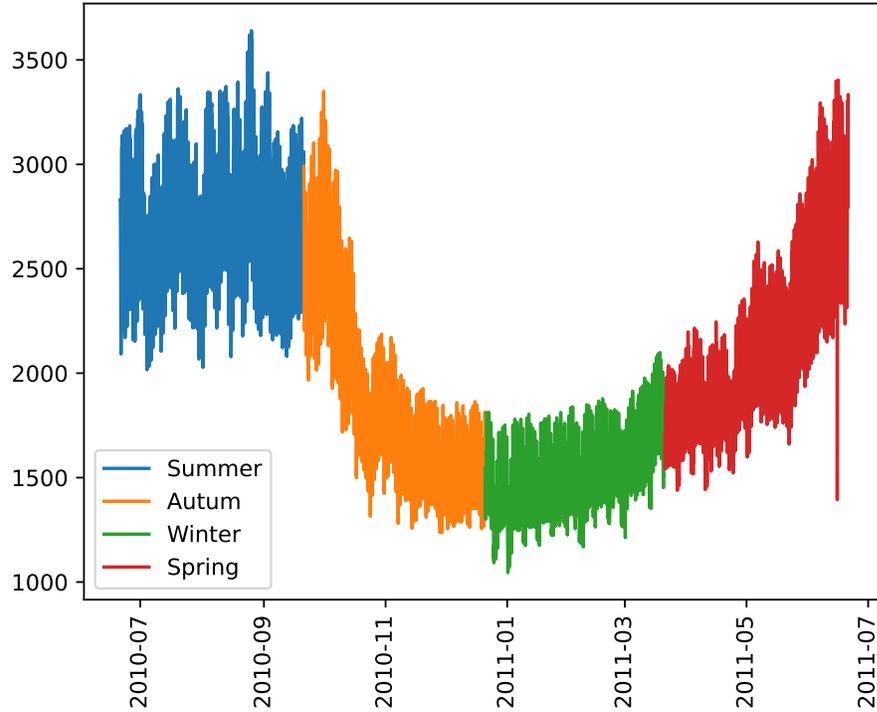


Figure 7.7: Time series of load demand.

### 7.3.2 PARAMETERS

The AnMA approach has few parameters. Firstly, a subsequence length of 288 data points is used for  $v_1$ , equivalent to one day. This subsequence size is important because it determines the length of the subsequence of the neighbors. Additionally, the AnMA searches five neighbors  $k = 5$ , and a lag of 144 periods are used for baseline forecast error correction through MA.

We initially planned to use auto-ARIMA libraries from [46, 35] for day-ahead forecasts since they have been effective in previous studies [51]. However, we found that they were unsuitable for real-time scenarios due to their extensive computation time, which averaged around ten minutes per forecast, exceeding the maximum time limit of ten seconds per forecast. Instead, we opted for a faster AR-MA approach. Additionally, to account for the daily and weekly seasonality of electricity demand [56], we implemented an AR model with a 7-day lag ( $288 \times 7$  periods). We performed error correction by applying an MA model with a half-day lag (144 periods) to the errors of the AR forecast. Finally, the

HWA and HWM models were trained using the historical data from the last month.

### 7.3.3 REAL-TIME DEMAND FORECASTING TESTS RESULTS

The results of the five-minute test suggest that the AnMA variants with the PCR, Lasso, and Ridge regression methods are the most efficient methods in terms of CPU time, with mean values of 1.6022, 1.5335, and 1.6184 seconds, respectively. The mean CPU execution time of AnMA-PCR was 1.6022 seconds, used into 1.5959 seconds for neighbor selection, while 0.0018 seconds were used for regression. It should be noted that the time for baseline correction was negligible. Furthermore, these methods use Pearson's coefficient as the similarity metric, which has proven effective in this context.

The Persistence method had negligible CPU time and usage, indicating that it is high-speed and lightweight. However, the HWM, HWA, and ARMA methods had longer mean CPU times, with values of 5.5842, 3.5475, and 12.6203 seconds, respectively.

Regarding accuracy, the AnMA variants with PCR, Lasso, and Ridge methods had lower MAPE values of 0.2512, 0.2892, and 0.2893, respectively. On the other hand, benchmarks such as HWM, HWA, Persistence, and ARMA obtained higher MAPE values of 0.3604, 0.3637, 0.3310, and 0.1633, respectively.

It is important to mention that in both tests, AnMA and its related methods (HWA, HWM, and persistent) utilized only one processor core, whereas ARMA utilized all eight cores. The tests recorded the minute-by-minute CPU utilization.

All models, except for ARMA, were tested using a single processing core to ensure a fair comparison. Although the Statsmodel [82] library implemented in ARMA is designed to utilize all computer resources to speed up the process, our benchmarks, including our AnMA, still outperformed it in terms of speed.

Furthermore, the AnMA variants with PCR, Lasso, and Ridge methods had lower MAE values of 6.80, 7.83, and 7.83 in MW, respectively. In contrast, benchmarks such as HWM, HWA, Persistence, and ARMA obtained higher MAE values of 9.82, 9.91, 9.03, and 4.45 in MW, respectively.

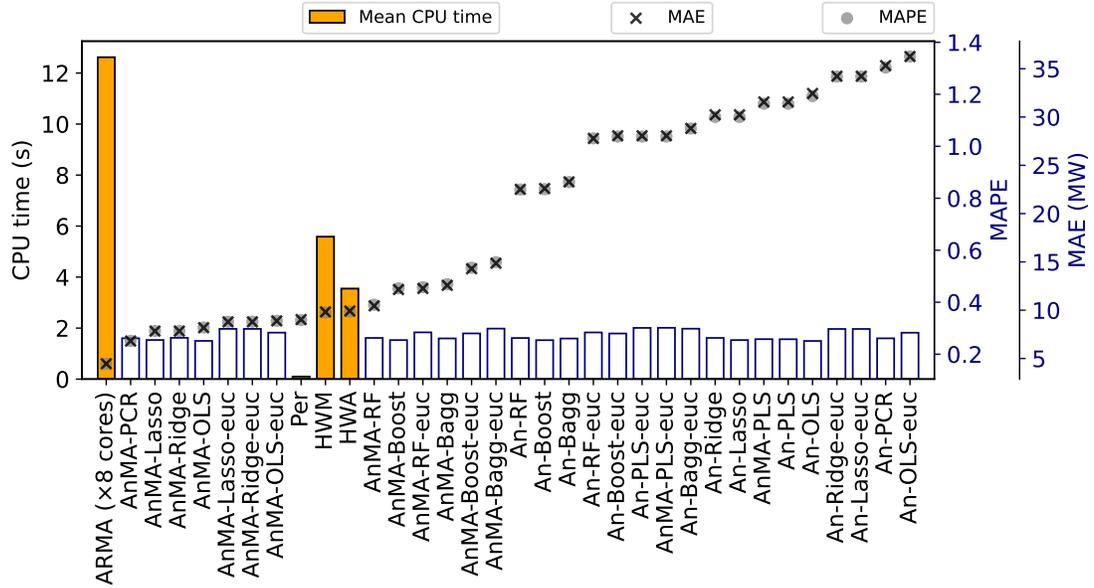


Figure 7.8: Real-time demand forecast summary in five minutes.

These results suggest that the AnMA variants with PCR, Lasso, and Ridge methods are efficient and accurate for forecasting, while the Persistence method is a good choice when speed is of utmost importance; however, Persistence obtained the worst accuracy results compared to the other benchmarks. The benchmarks such as HWM, HWA, and ARMA may be less suitable for this particular application, given their higher CPU times.

Figure 7.8 and Figure 7.9 provide a visual representation of the performance comparison of the forecasting methods. Both figures have a dual-axis plot, where the left axis represents CPU execution time in seconds, and the right axis represents MAPE and MAE values. The blue bars represent the performance of AnMA methods, while the orange bars represent the benchmark methods such as Persistence, HWA, HWM, and ARMA. Additionally, black crosses denote the MAE, and gray dots denote the MAPE values. The figures clearly compare the methods, indicating the fastest and most accurate forecasting methods.

Statistical tests are conducted for both tests to determine the significant differences in the means of the distributions of the absolute errors of AnMA-PCR, HWA, HWM, Persistence, and ARMA. A Friedman test was conducted in the five-minute test, and a significant difference was found  $p$ -value = 0.0000, implying that the null hypothesis of equal means between groups is rejected and the alternative hypothesis of difference

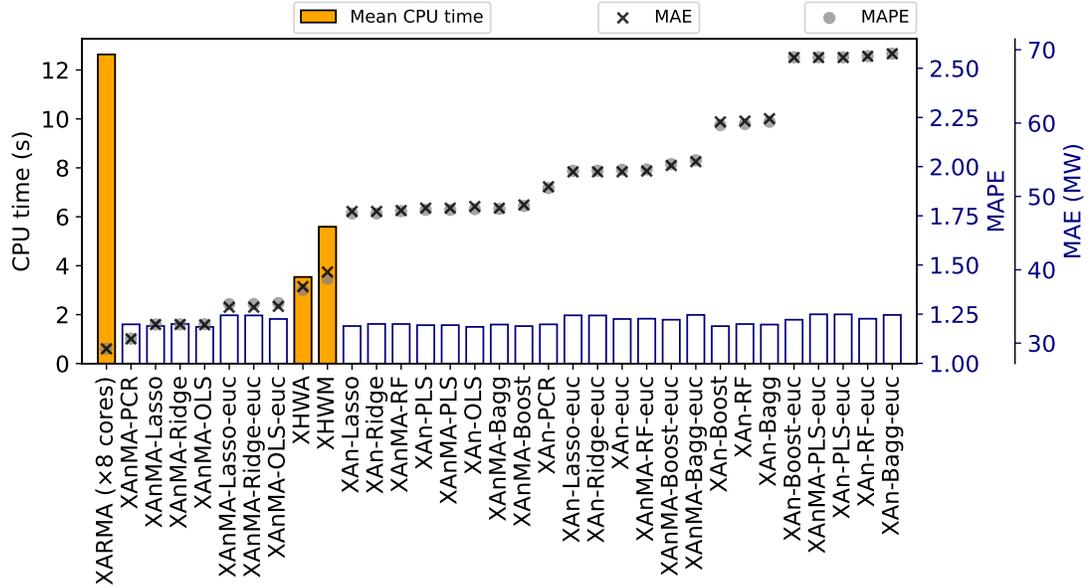


Figure 7.9: Summary of the real-time demand forecast in two-and-a-half hours.

between groups is accepted. Subsequently, a Wilcoxon paired test was performed, and significant results were obtained between AnMA-PCR and Persistence, HMA, HWM, and AnMA-OLS with  $p$ -value = 0.0000 for all hypothesis tests. This results in rejecting the null hypothesis of equality and accepting the alternative hypothesis that AnMA-PCR has lower absolute errors on average compared to the standard AnMA-OLS version and benchmark methods. In a two-and-a-half-hour test, a Friedman test was conducted and found to be significant  $p$ -value = 0.0000, leading to the rejection of the null hypothesis of equal means between groups and the acceptance of the alternative hypothesis of difference between groups. Subsequently, a Wilcoxon paired test was then performed, resulting in significant results  $p$ -value = 0.0000 for AnMA-PCR compared to Persistence, HMA, and HWM. The Wilcoxon test showed ARMA errors were significantly smaller than AnMA-PCR, with  $p$ -value  $p$ -value = 0.0000. This rejection of the null hypothesis of equality indicates that AnMA-PCR has, on average, lower absolute errors than the standard AnMA-OLS version, HMA, and HWM. Note that rejecting the null hypothesis of equality indicates that ARMA has, on average, lower absolute errors than AnMA-PLS.

All these results are further supported by Figures 7.10 and 7.11, which display the overlapping distribution of absolute errors between AnMA-PCR and the benchmark methods.

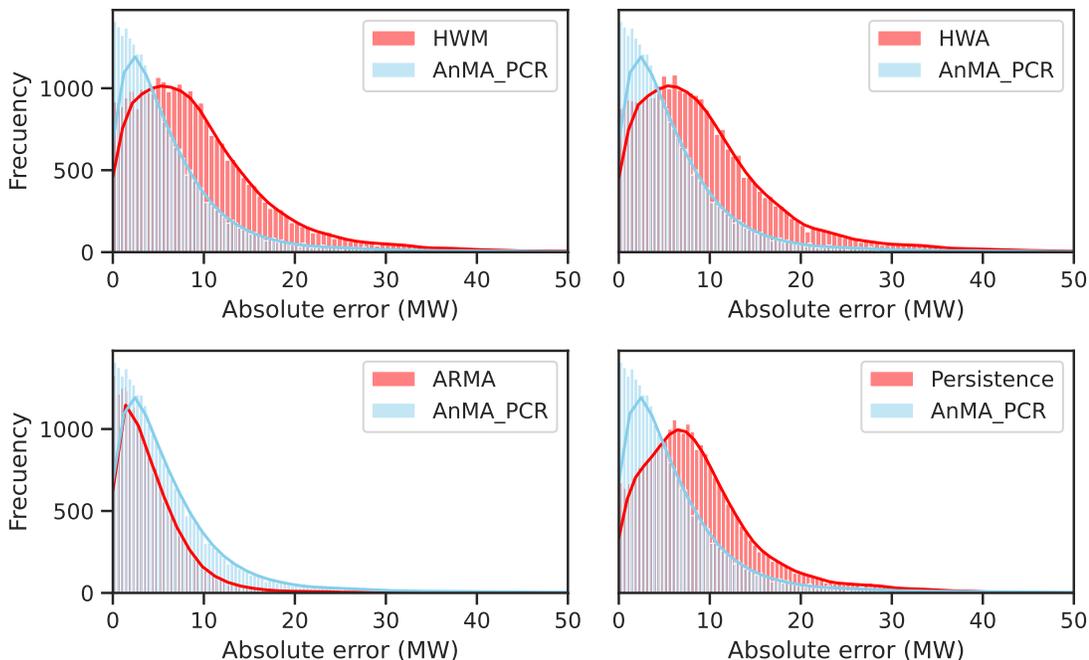


Figure 7.10: Comparison of absolute error distributions between **AnMA-PCR** and benchmarks, five-minute test.

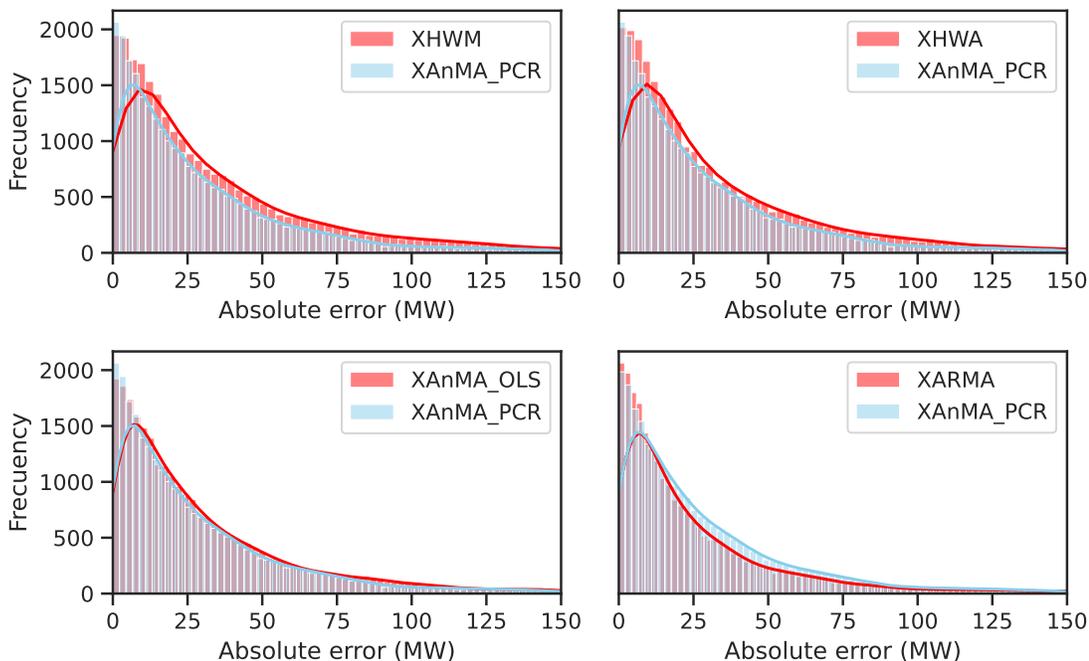


Figure 7.11: Comparison of absolute error distributions between **AnMA-PCR** and benchmarks, two-and-a-half hours' test.

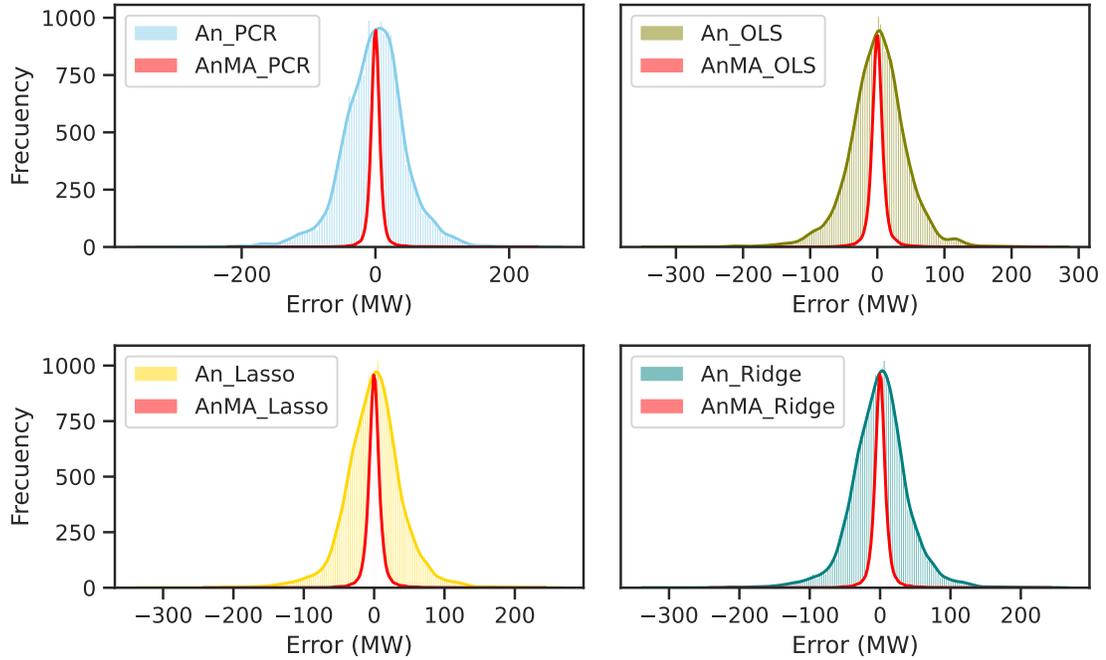


Figure 7.12: Distribution of errors between forecasts with MA correction, five-minute test.

We have observed a reduction in error variance in methods that incorporate the Moving Average (MA) component, namely AnMA-PCR, AnMA-Lasso, AnMA-Ridge, AnMA-OLS, and AnMA-OLS-euc; compared to methods that do not include this component, such as An-PCR, An-Lasso, An-Ridge, An-OLS, and An-OLS-euc. This variance reduction effect has been confirmed using Levene’s statistical test, which assumes a null hypothesis ( $H_0$ ) that the error distributions of all methods have the same variance and an alternative hypothesis ( $H_a$ ) that the error distributions have different variances. The results of the test indicate a significant difference, providing sufficient evidence to conclude that the methods using MA have lower variance than those that do not. The table containing the results of Levene’s test is provided in Table D.1. Additionally, the overlapping distribution of errors between AnMA-PCR, AnMA-Lasso, AnMA-Ridge, AnMA-OLS, AnMA-OLS-euc and the An-PCR, An-Lasso, An-Ridge, An-OLS, and An-OLS-euc methods shown in Figures 7.12 and 7.13 provides further support for these results and thus observe the benefits of the correction stage. An example of bias correction in the forecasted time series is illustrated in Figure 7.14. First, a baseline forecast ( $Y'$ ) is generated and compared to actual observations to calculate the bias. The MA model is then applied to estimate the

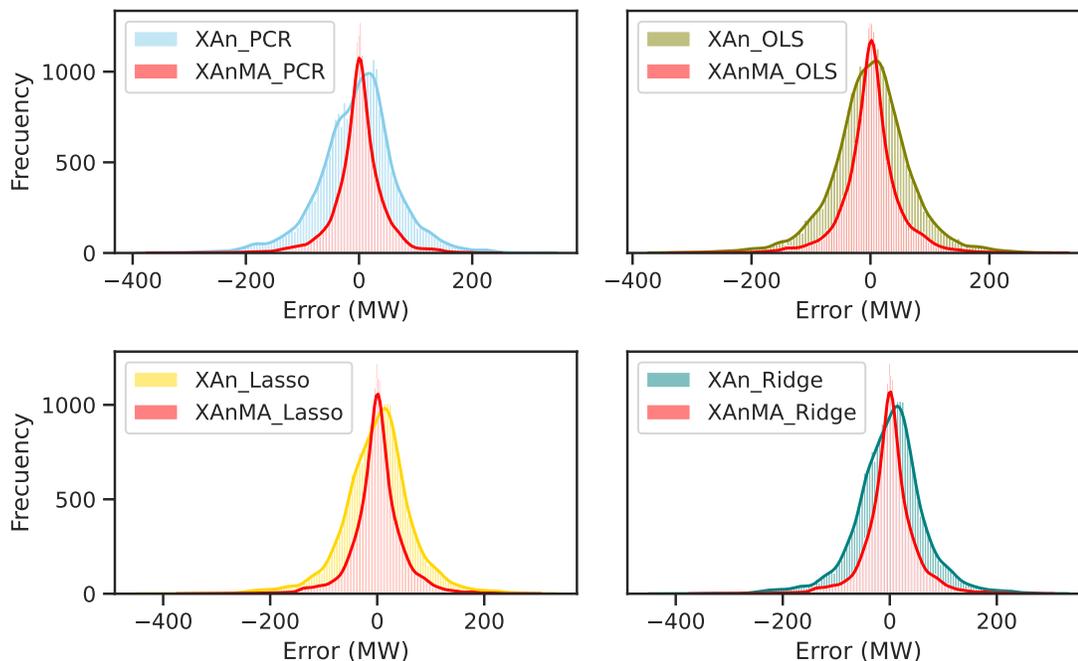


Figure 7.13: Distribution of errors between forecasts with MA correction, two-and-a-half hours' test.

expected error based on the bias and adjust the original forecast to correct for the bias. It is important to note that the baseline forecast needs to be stored for some time before the MA calculation can be performed, and the MA model will correct for the bias in the same period in which it occurs.

In Figure 7.15, we display the August forecast results for the AnMA-PCR and ARMA methods during a two-and-a-half-hour test. The graph depicts the predicted values from each method in orange and green lines, respectively, while the actual observations are shown in blue. The predicted values of both methods closely follow the observed data points, with a minimal amount of deviation between the predicted and actual values. This low deviation indicates that both methods produce accurate and reliable forecasts, making them well-suited for real-time load demand forecasting applications.

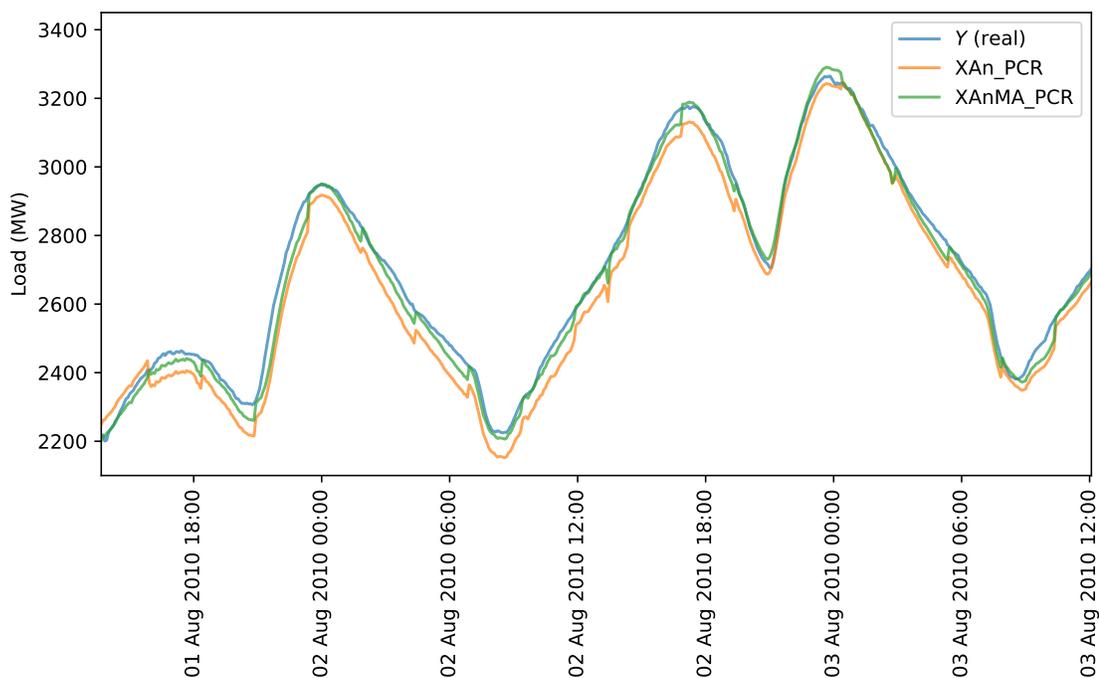


Figure 7.14: Correction for the An forecast bias through of MA model.

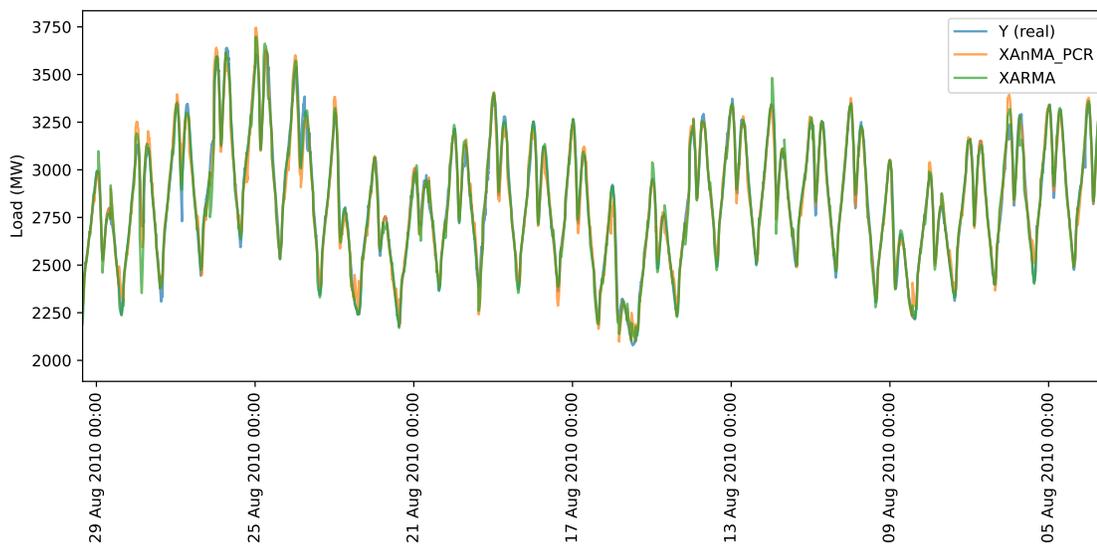


Figure 7.15: August forecast for AnMA-PCR and ARMA methods, two and a half hours test.

## 7.4 DISCUSSION

The objective of this study was to compare the performance of various methods for load demand forecasting, including our proposed method, in terms of accuracy and computational efficiency. The experimental results are discussed as follows.

The two tests showed that the AnMA method with its PCR, Lasso, and Ridge variants had a faster average computational time and the most significant accuracy compared to the HWA and HWM methods. Furthermore, although the persistent model was much faster, the accuracy was higher for AnMA. On the other hand, although the AnMA method was much faster than ARMA, ARMA was more accurate; however, the runtime of ARMA exceeded the established runtime of 10 seconds. Moreover, the dimension-reduced and shrinkers variants of AnMA performed better than the standard AnMA version using stepwise OLS regression. Statistical tests support all findings.

The best variants of AnMA use Pearson's coefficient as a similarity metric instead of Euclidean distance, taking advantage of the seasonal and repetitive behaviors in load data to quickly find the most similar past days. This makes the method fast and efficient, with computational complexity proportional only to the number of windows being compared, making it an ideal choice for load demand forecasting. Therefore, the Pearson correlation assumption enhances the efficiency of the following regression process, making it a synergistic feature for the AnMA method.

Moreover, AnMA is particularly suitable for real-time applications due to its calculation speed and low CPU usage. It can also be shared with other processes on the same machine, providing an indirect advantage.

The tests' finding shows that incorporating bias correction techniques such as MA can improve the accuracy and reliability of their forecasts in real-time, supported by statistical tests.

The CPU time and resource consumption for the AnMA-PCR method were significantly lower using one core to 15% capacity, in contrast to the ARMA model, which proved to be the most computationally expensive, occupying all eight processor cores with a mean

CPU utilization of 92%.

Furthermore, the environmental impact of each method was assessed by calculating their carbon footprint, which takes into account the CPU usage and core count, as well as the location and computer specifications. The results, including the energy consumption (in kWh) and CO<sub>2</sub> emissions, are presented in Table 7.2, along with the total runtime of each method, the number of cores used, and the energy spent in kWh. The calculator footprint used is available online at <http://calculator.green-algorithms.org/>.

Table 7.2: Carbon footprint per method, [55].

Method	Runtime	Cores	% CPU	kWh	kg-CO <sub>2</sub> e
ARMA	92:53:00	8	92.0	7.7500	3.3400
HWM	41:06:00	1	17.1	1.0400	0.4472
HWA	26:07:00	1	16.8	0.6581	0.2839
AnMA-Ridge	11:54:00	1	18.0	0.3010	0.1298
AnMA-PCR	11:47:00	1	15.0	0.2952	0.1273
AnMA-Lasso	11:17:00	1	27.5	0.2941	0.1268
Persistence	00:44:00	1	1.0	0.0078	0.0180

## 7.5 CONCLUSIONS

The main contribution of this chapter is developing the AnMA method as a novel real-time load demand forecasting approach that combines Analog and Moving Average techniques within a flexible framework with exchangeable components. Its accuracy is competitive with the current state-of-the-art, requiring less time and CPU use than the benchmarks. The method's main advantage is its low computational cost, allowing it to operate in real-time scenarios. Additionally, its MA component adapts immediately to changes in data behavior, ensuring its accuracy and efficiency.

To evaluate the performance of our AnMA method and its variants, we conducted a simulation of three months of real-time load demand forecasting. The simulation results showed that our AnMA method with PCR, Lasso, and Ridge variants outperformed the other methods in terms of accuracy and computational efficiency. Specifically, our method

had a faster mean computational time and higher accuracy compared to the HWA, HWM, and persistent methods and had accuracy similar to ARMA but at a much lower computational cost. This was achieved using Pearson’s coefficient as a similarity metric, dimension reduction, and shrinking regressions. Based on these results, we conclude that AnMA with PCR, Lasso, and Ridge variants is a promising real-time load demand forecasting approach with high accuracy and computational efficiency.

AnMA adapts to demand changes by estimating residual errors of a baseline forecast using a moving average MA component. The MA correction reduces variance and improves performance.

Compared to benchmark statistical methods, our developed approach enables automated demand forecasting every five minutes, 24/7, yielding results in less than two seconds and demonstrating competitive accuracy.

We are currently working on improving the accuracy of the AnMA algorithm while maintaining its efficiency. In the next version, we plan to experiment with selecting neighbors from subsets of days. Additionally, we plan to work on parallelizing the algorithm, expecting it to lead to better performance. Our goal is to continue innovating and enhancing the AnMA algorithm to better serve the field of load demand forecasting with an efficient method.

## 7.6 FUTURE RESEARCH

- The accuracy can be improved by pre-selecting the days on weekdays, Saturdays, Sundays, and holidays.
- Other similarity measures, such as dynamic time warping, can be used and evaluated.
- It would be interesting to compare this method against pattern similarity-based methods, where the forecasting problem can be simplified, as proposed by [23].
- Other computationally inexpensive error smoothing methods, such as Kalman filtering, can be tested.

- 
- Computation time can be further reduced, and more historical data can be used by parallelizing the sample selection process.
  - It is noted that the largest errors of the AnMA method are found in the day's peaks. A new version of the AnMA method that includes the prediction of peak demand can be included in the method.
  - The accuracy of load demand forecasts can be improved by incorporating weather variables and adding a corrective phase.

# MAIN CONCLUSIONS

---

## 8.1 SUMMARY OF MAIN FINDINGS

This dissertation tackles significant and relevant power system operational planning issues, specifically a thermal unit commitment problem and a real-time power demand forecasting problem. Both problems need the development of efficient algorithms capable of effectively addressing the inherent trade-off between accuracy and speed. This requirement stems from the limited time available to solve these problems and the imperative to achieve high accuracy.

First, we address the generator scheduling problem by solving the unit commitment problem using five matheuristic methods, four local branching (LB) versions, and one kernel search (KS) version. Furthermore, a new constructive method belonging to the relax & fix family, called HARDUC, was developed. This method has successfully generated high-quality, feasible initial solutions for the five matheuristic methods. In addition, we proposed a new thermal unit commitment model with features of tightness and compactness with a staircase cost function.

Tests were conducted on the developed matheuristic with time limits of 4000 seconds and 7200 seconds to find a solution. These tests were designed to reflect practical scenarios where an analyst needs to deliver a generation schedule within approximately one or two hours. A comparison was made against a state-of-the-art solver such as CPLEX, considering two cases: (i) running the solver from scratch and (ii) feeding an initial feasible

solution found by the heuristic.

The results revealed that, for small instances, the off-the-shelf solver outperformed the matheuristic methods. However, in medium-sized instances, the solver sometimes fails to find a solution. When a solution was found, no significant difference was observed between the solver and the methods. However, for large instances where the solver left many cases unsolved, the proposed methods successfully solved all instances and achieved outstanding results compared to the solver. Statistical tests provided support for all the results.

Additionally, tests were conducted on the proposed constructive method with a maximum runtime limit of 1200 seconds. A comparison was made against the best UCP constructive method available in the literature, as well as against the best solution found within the solver's time limit. Unlike the other two methods, the proposed method consistently found feasible solutions for all tested instances, with a better optimality gap. Statistical tests supported all the obtained results.

It is worth highlighting that our implementation of the KS algorithm exhibited superior performance, particularly in the more challenging instances, outperforming both the solver and the LB method regarding the optimality gap. This finding is significant as the full potential of utilizing the KS for solving the UCP has not been fully explored until now. It highlights promising avenues for further research that remain open.

As expected, we observed that as the problem instances became more complicated, the matheuristic methods outperformed the solver by finding better and faster solutions. Matheuristics have demonstrated remarkable effectiveness when applied to tackle other problems. They have been incorporated as heuristics in off-the-shelf solvers, enabling efficient solutions for a wide range of mixed-integer linear programming (MILP) problems. However, in this research, we have leveraged our understanding of the mathematical structure of the thermal UCP to enhance the efficiency of our methods. By identifying dominant variables and selecting candidate variables to be included in the solution, we customized the matheuristics specifically for the UCP. Additionally, we tackled a common problem researchers face in implementing the KS method by using the Sturges statistical rule to calculate the appropriate number of buckets. These specific adaptations of the methods

enhance the efficacy of matheuristics and make them exceptionally powerful for addressing the UCP.

In the context of the very-short term load demand forecasting problem, we introduced a novel approach called analogue with moving average method (AnMA). This method was specifically designed to leverage the seasonal characteristics of the time series of load demand, selecting the most correlated days. Furthermore, its capability to adapt to real-time data makes it highly suitable for adjusting to new demand patterns, correcting biases, and improving accuracy.

The AnMA method was compared to other benchmark methods from the literature, known for their high efficiency and accuracy. The results clearly demonstrated that the AnMA method outperformed naive algorithms and exponential smoothing methods regarding the accuracy, computational speed, and low computational cost. Moreover, it achieved comparable accuracy to ARIMA models while maintaining significantly lower computational costs and considerably shorter runtime.

As a result, we achieved a fast and highly effective approach that stands competitively in accuracy compared to benchmark forecasting methods. Notably, it excels in terms of runtime, showcasing exceptional efficiency. Finally, AnMA was a method adapting from meteorology to real-time demand forecasting, one example of how techniques from other fields were incorporated.

It has been demonstrated by our research that both the matheuristic methods and the analogies with the moving average method developed outperform the solver and benchmark forecasting methods in terms of speed and accuracy. The effectiveness of these approaches in addressing critical power system planning problems is highlighted.

We believe that the success of our proposed methods can be attributed to our understanding of the specific features of both problems and our ability to leverage these features when designing solution methods. We anticipate that these findings will offer valuable insights for developing and implementing future hybrid heuristic methods based on mixed-integer programming or matheuristics and forecasting methods based on Analogues. This holds big potential for driving substantial practical advancements in the electricity industry.

## 8.2 FUTURE RESEARCH

In addition to the extensive discussion of future work for each research project at the end of each chapter, an additional list of potential research topics is provided below.

- Multi-objective optimization can be considered: Different trade-offs can be explored by including multiple objectives, such as minimizing operating costs, reducing state changes or movements in generators, or reducing greenhouse gas emissions. In practice, for market operators and generators, it is desirable that the UCP solution not only be the most cost-effective for the system but also incorporate only the necessary or indispensable changes to generators. Therefore, the multi-objective UCP problem would involve introducing a new objective function that minimizes either the number of state changes of generators or the total power difference of generators between periods. To our knowledge, this trade-off has not been studied yet.
- Uncertainty and robustness should be addressed: Techniques to handle uncertainties in the UCP, such as electricity demand, renewable energy sources, or unit outages, can be explored. Some optimization methods to tackle these problems are robust optimization or scenario-based methods to obtain resilient solutions to uncertain conditions. The developed matheuristics can effectively solve these problems when modeled as MILP. Drawing from the approach used in solving the deterministic thermal UCP, our implementation of LB and KS incorporates concepts such as soft-fixing, restricted candidate list, and dominant variable selection. These matheuristics can be applied to address the stochastic UCP and achieve efficient solutions.
- It is recommended to combine the proposed and conventional methods to diversify the approaches. Running in parallel on different computers can offer to obtain a variety of solutions. This way, the best solution can be chosen, and additional tools can be added to complement the existing methods.
- The thermal UCP can be addressed using a matheuristic approach that combines constraint programming and mathematical programming to generate an initial feasible solution for the thermal Unit Commitment Problem (UCP). The strategy involves discretizing the operating range of generators and incorporating various constraints,

such as generator limits, minimum up/down times, ramping constraints, spinning reserves, and demand-supply balancing. This enables the determination of the commitment status of generators and the calculation of power levels through an economic dispatch calculation, solving a linear problem. This approach is expected to significantly reduce the construction time required for the initial solution compared to our proposed HARDUC method.

- One main challenge is efficiently designing models that address the UPC with network constraints and consider the allocation of hydro and renewable units considering the hydro grid and all the characteristics of such units. Another challenge is considering the couplings between units in combined cycle package plants.

## APPENDIX A

# DESCRIPTION OF INSTANCES

---

Table A.1: x7day\_small instances group.

Group	File names	Periods	Gen	Configuration							
				G1	G2	G3	G4	G5	G6	G7	G8
x7day_small	uc.061	168	28	12	11	0	0	1	4	0	0
x7day_small	uc.062	168	35	13	15	2	0	4	0	0	1
x7day_small	uc.063	168	44	15	13	2	6	3	1	1	3
x7day_small	uc.064	168	45	15	11	0	1	4	5	6	3
x7day_small	uc.065	168	49	15	13	3	7	5	3	2	1
x7day_small	uc.066	168	50	10	10	2	5	7	5	6	5
x7day_small	uc.067	168	51	17	16	1	3	1	7	2	4
x7day_small	uc.068	168	51	17	10	6	5	2	1	3	7
x7day_small	uc.069	168	52	12	17	4	7	5	2	0	5
x7day_small	uc.070	168	54	13	12	5	7	2	5	4	6
x7day_small	uc.101	168	60	11	18	6	3	7	8	2	5
x7day_small	uc.102	168	62	16	17	5	3	5	6	7	3
x7day_small	uc.103	168	64	13	20	4	9	4	6	2	6
x7day_small	uc.104	168	66	17	16	8	8	0	6	6	5
x7day_small	uc.105	168	70	14	21	9	5	8	6	3	4
x7day_small	uc.106	168	71	25	18	2	4	7	5	2	8
x7day_small	uc.107	168	75	16	25	5	9	3	3	8	6
x7day_small	uc.108	168	76	21	21	4	3	8	10	4	5
x7day_small	uc.109	168	80	17	24	8	9	8	1	8	5
x7day_small	uc.110	168	81	22	21	4	9	4	3	8	10

Table A.2: x7day\_medium instances group.

Group	File names	Periods	Gen	Configuration							
				G1	G2	G3	G4	G5	G6	G7	G8
x7day_medium	uc_111	168	85	19	27	6	6	8	9	3	7
x7day_medium	uc_112	168	86	24	23	6	7	4	8	6	8
x7day_medium	uc_113	168	90	24	28	6	7	6	3	12	4
x7day_medium	uc_114	168	91	26	24	9	4	5	10	6	7
x7day_medium	uc_115	168	94	22	28	4	10	5	9	7	9
x7day_medium	uc_116	168	97	27	26	8	14	5	6	7	4
x7day_medium	uc_117	168	98	24	30	4	8	10	7	10	5
x7day_medium	uc_118	168	102	29	28	10	6	5	9	9	6
x7day_medium	uc_119	168	104	25	32	9	0	10	11	9	8
x7day_medium	uc_120	168	106	30	29	11	9	5	2	11	9
x7day_medium	uc_121	168	109	32	34	9	7	8	9	4	6
x7day_medium	uc_122	168	111	32	36	6	4	10	7	10	6
x7day_medium	uc_123	168	114	29	36	11	10	6	5	7	10
x7day_medium	uc_124	168	116	34	32	7	5	9	9	11	9
x7day_medium	uc_125	168	119	32	37	9	5	7	11	9	9
x7day_medium	uc_126	168	121	35	34	7	11	12	4	7	11
x7day_medium	uc_127	168	125	32	39	5	8	5	15	13	8
x7day_medium	uc_128	168	126	37	36	10	12	4	12	10	5
x7day_medium	uc_129	168	129	33	40	9	7	7	14	5	14
x7day_medium	uc_130	168	131	38	37	14	10	6	7	5	14
x7day_medium	uc_131	168	131	34	42	8	8	8	11	14	6
x7day_medium	uc_071	168	132	46	45	8	0	5	0	12	16
x7day_medium	uc_132	168	135	37	44	8	8	9	9	13	7
x7day_medium	uc_133	168	138	42	40	7	11	9	10	6	13
x7day_medium	uc_134	168	140	38	45	8	7	10	14	10	8
x7day_medium	uc_135	168	143	43	42	7	12	7	15	10	7
x7day_medium	uc_136	168	146	40	47	9	7	15	13	7	8
x7day_medium	uc_137	168	148	45	44	7	16	9	10	10	7
x7day_medium	uc_138	168	151	41	48	9	16	11	8	9	9
x7day_medium	uc_139	168	154	46	45	10	15	10	3	15	10
x7day_medium	uc_140	168	156	43	48	16	11	9	12	9	8
x7day_medium	uc_072	168	156	40	54	14	8	3	15	9	13
x7day_medium	uc_073	168	156	50	41	19	11	4	4	12	15

Table A.3: x7day\_large instances group.

Group	File names	Periods	Gen	Configuration							
				G1	G2	G3	G4	G5	G6	G7	G8
x7day_large	uc.074	168	165	51	58	17	19	16	1	2	1
x7day_large	uc.075	168	167	43	46	17	15	13	15	6	12
x7day_large	uc.076	168	172	50	59	8	15	1	18	4	17
x7day_large	uc.077	168	182	53	50	17	15	16	5	14	12
x7day_large	uc.078	168	182	45	57	19	7	19	19	5	11
x7day_large	uc.079	168	183	58	50	15	7	16	18	7	12
x7day_large	uc.080	168	187	55	48	18	5	18	17	15	11
x7day_large	uc.141	168	185	52	57	15	12	12	13	11	13
x7day_large	uc.142	168	195	55	62	15	7	18	12	11	15
x7day_large	uc.143	168	188	54	52	16	4	16	22	14	10
x7day_large	uc.144	168	205	60	64	17	13	19	10	10	12
x7day_large	uc.145	168	219	58	52	18	15	16	25	18	17
x7day_large	uc.146	168	200	58	60	16	15	21	9	9	12
x7day_large	uc.147	168	212	58	67	16	7	15	22	16	11
x7day_large	uc.148	168	212	63	60	17	15	22	15	7	13
x7day_large	uc.149	168	221	66	65	19	3	13	25	15	15
x7day_large	uc.150	168	204	65	60	17	22	24	5	9	2
x7day_large	uc.151	168	219	67	62	19	9	15	21	14	12
x7day_large	uc.152	168	243	65	64	21	23	25	16	12	17
x7day_large	uc.153	168	263	72	67	25	22	19	8	21	29
x7day_large	uc.154	168	270	80	66	15	15	21	7	40	26
x7day_large	uc.155	168	285	95	80	15	25	25	15	15	15
x7day_large	uc.156	168	300	80	95	33	33	21	21	10	7
x7day_large	uc.157	168	315	90	90	30	30	20	30	15	10
x7day_large	uc.158	168	330	85	130	45	35	7	15	10	3
x7day_large	uc.159	168	345	120	70	60	53	15	7	10	10
x7day_large	uc.160	168	360	50	90	30	70	60	10	10	40
x7day_large	uc.161	168	375	70	85	45	22	25	50	31	47
x7day_large	uc.162	168	390	55	99	70	53	34	30	12	37
x7day_large	uc.163	168	405	130	111	35	81	21	12	7	8

# RESULTS OF COMPARISON AMONG SOLUTION ALGORITHMS

---

## B.1 RESULTS ON INSTANCES FROM GROUP X7DAY\_SMALL

Instances extracted from Kazarlis et al. [52], seven days, and a number of generators between 85 and 156, files uc\_061-uc\_070 and uc\_101-uc\_110. All files are available in JSON format in the following repository: [https://github.com/urieliram/tc\\_uc](https://github.com/urieliram/tc_uc).

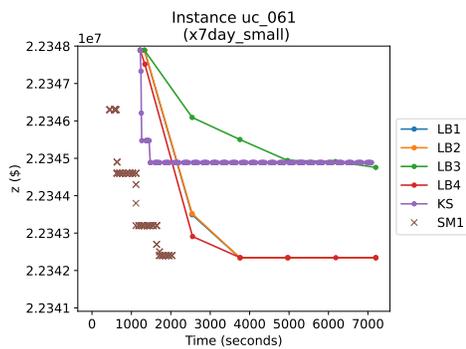


Figure B.1: Comparison of algorithms on instance 061.

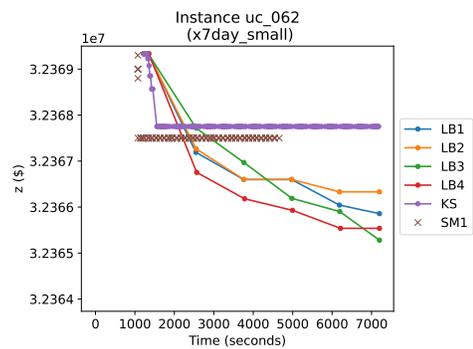


Figure B.2: Comparison of algorithms on instance 062.

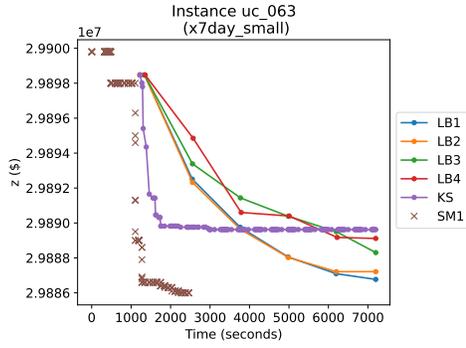


Figure B.3: Comparison of algorithms on instance 063.

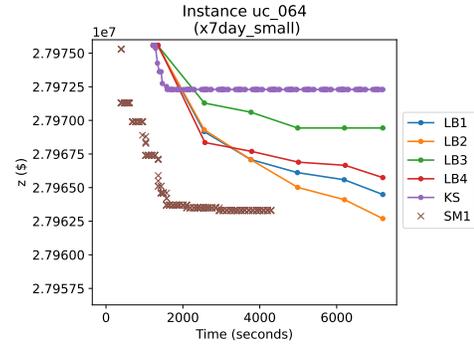


Figure B.4: Comparison of algorithms on instance 064.

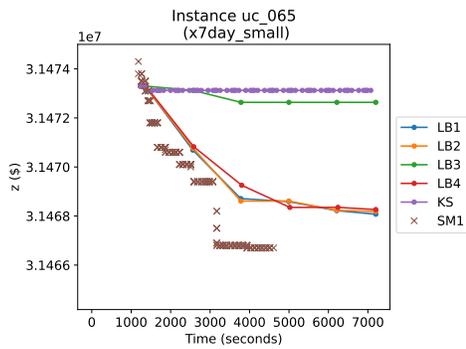


Figure B.5: Comparison of algorithms on instance 065.

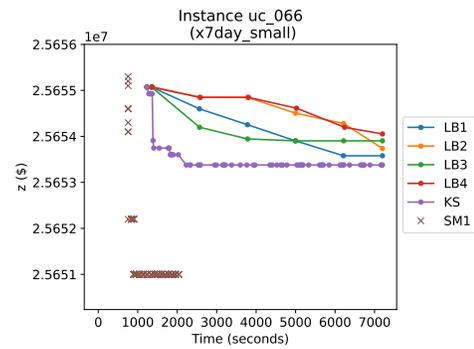


Figure B.6: Comparison of algorithms on instance 066.

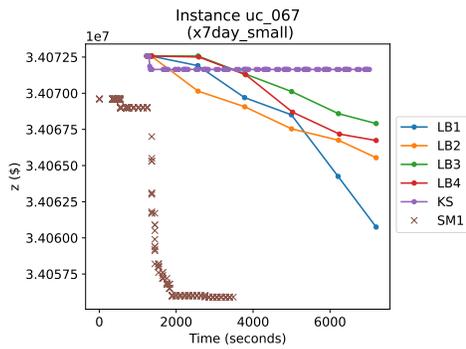


Figure B.7: Comparison of algorithms on instance 067.

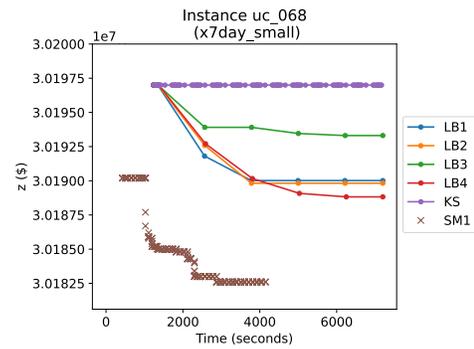


Figure B.8: Comparison of algorithms on instance 068.

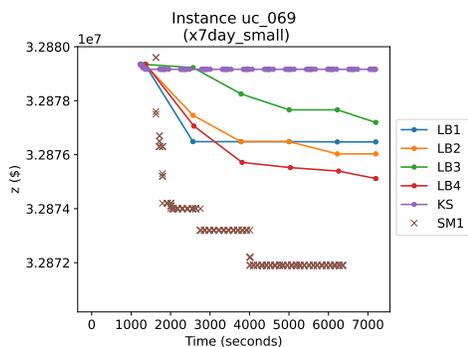


Figure B.9: Comparison of algorithms on instance 069.

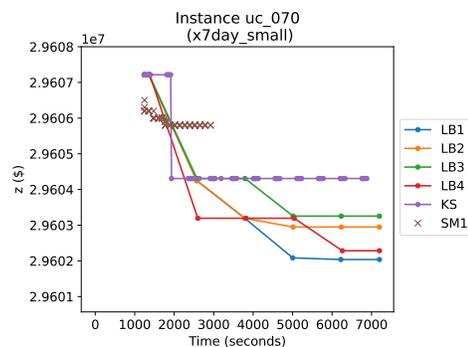


Figure B.10: Comparison of algorithms on instance 070.

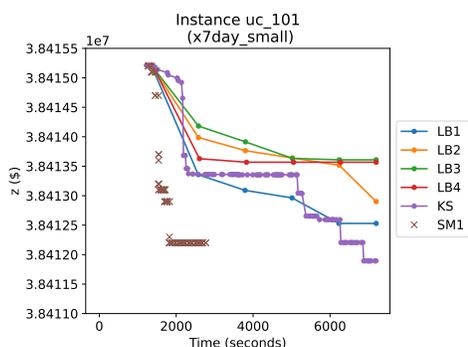


Figure B.11: Comparison of algorithms on instance 101.

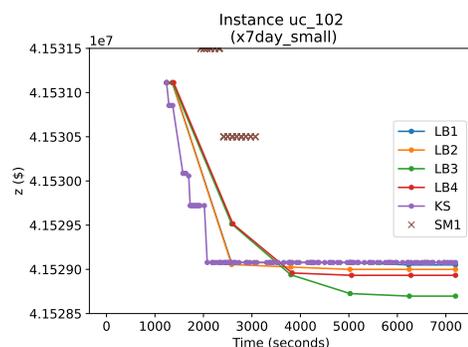


Figure B.12: Comparison of algorithms on instance 102.

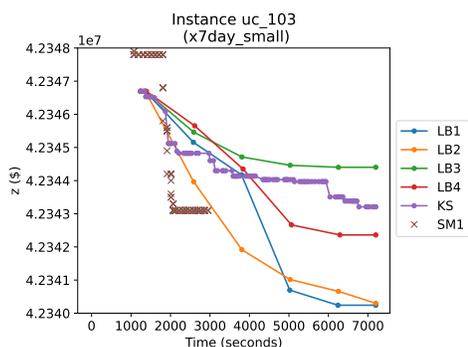


Figure B.13: Comparison of algorithms on instance 103.

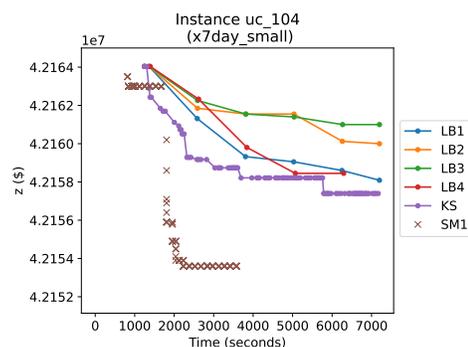


Figure B.14: Comparison of algorithms on instance 104.

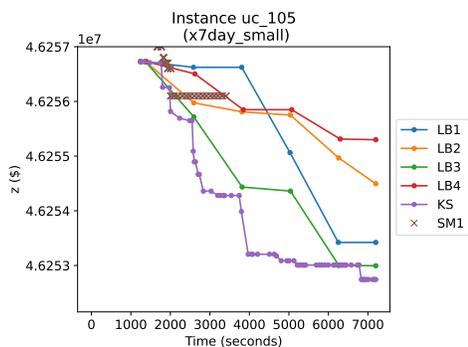


Figure B.15: Comparison of algorithms on instance 105.

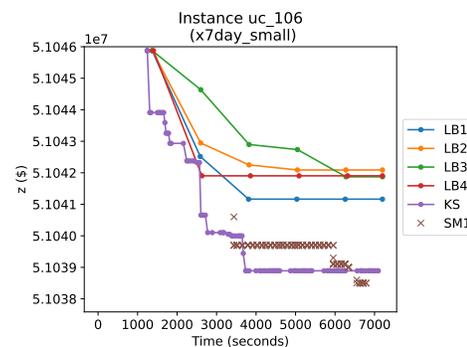


Figure B.16: Comparison of algorithms on instance 106.

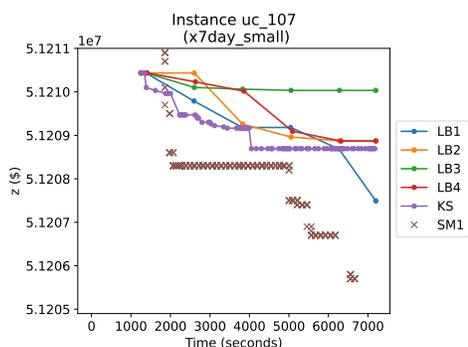


Figure B.17: Comparison of algorithms on instance 107.

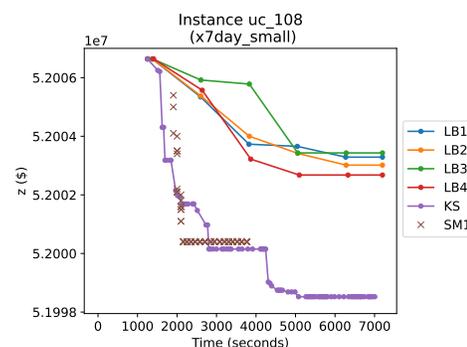


Figure B.18: Comparison of algorithms on instance 108.

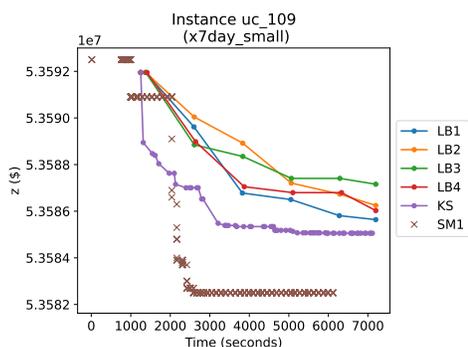


Figure B.19: Comparison of algorithms on instance 109.

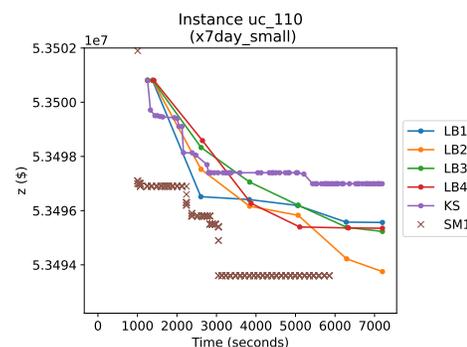


Figure B.20: Comparison of algorithms on instance 110.

## B.2 RESULTS ON INSTANCES FROM GROUP x7day\_medium

Most of these instances were constructed based on the parameters of the eight generators proposed by [52], seven days, and a number of generators between 85 and 156, files uc\_111-uc\_131, uc\_071-uc\_073, uc\_132-uc\_140. All files are available in JSON format in the following repository: [https://github.com/urieliram/tc\\_uc](https://github.com/urieliram/tc_uc).

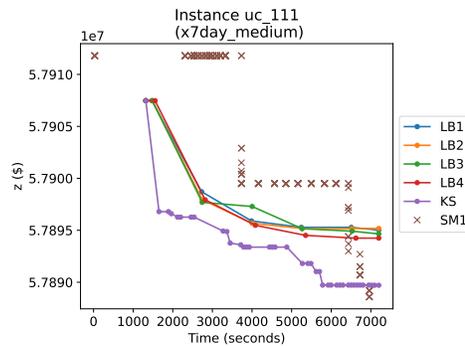


Figure B.21: Comparison of algorithms on instance 111.

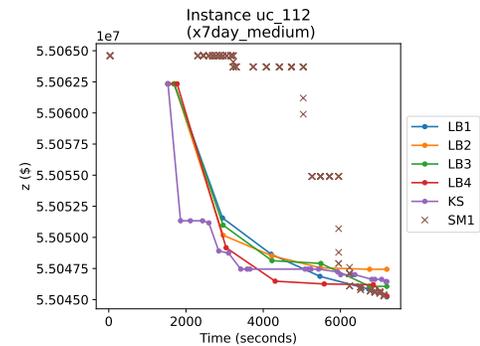


Figure B.22: Comparison of algorithms on instance 112.

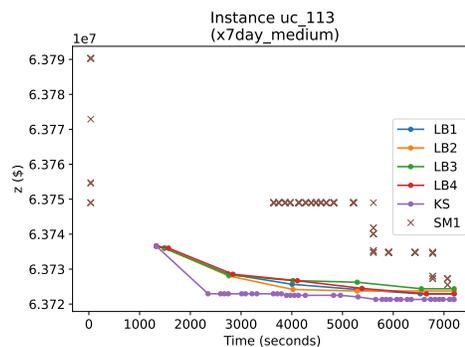


Figure B.23: Comparison of algorithms on instance 113.

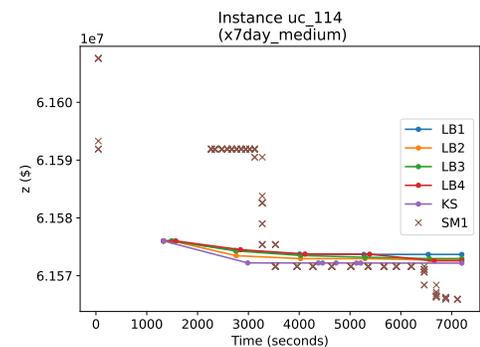


Figure B.24: Comparison of algorithms on instance 114.

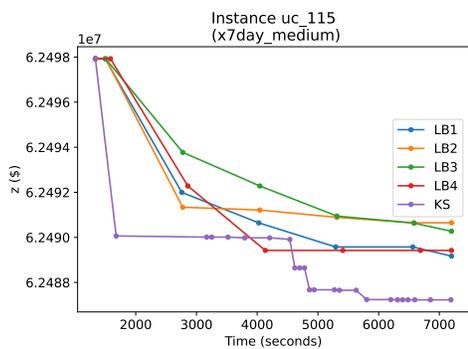


Figure B.25: Comparison of algorithms on instance 115.

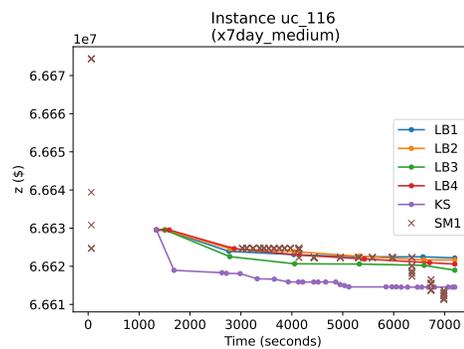


Figure B.26: Comparison of algorithms on instance 116.

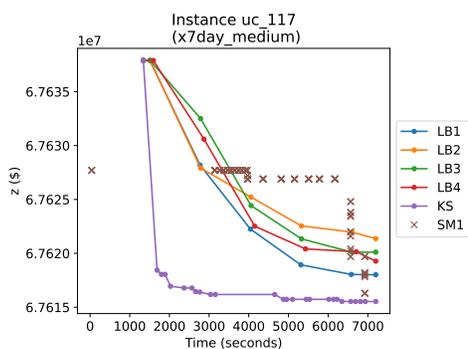


Figure B.27: Comparison of algorithms on instance 117.

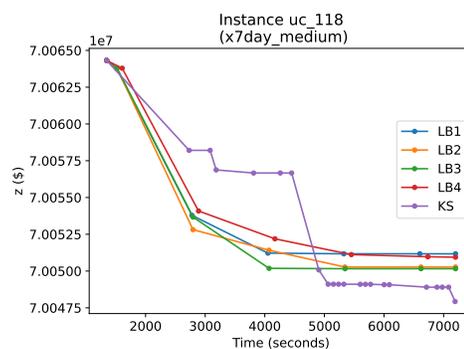


Figure B.28: Comparison of algorithms on instance 118.

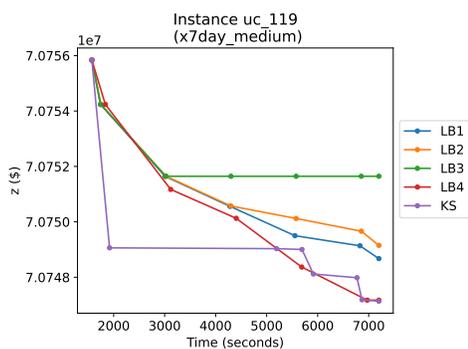


Figure B.29: Comparison of algorithms on instance 119.

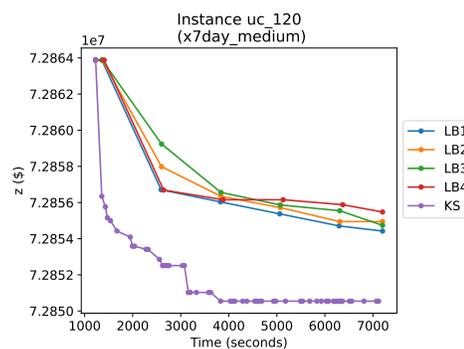


Figure B.30: Comparison of algorithms on instance 120.

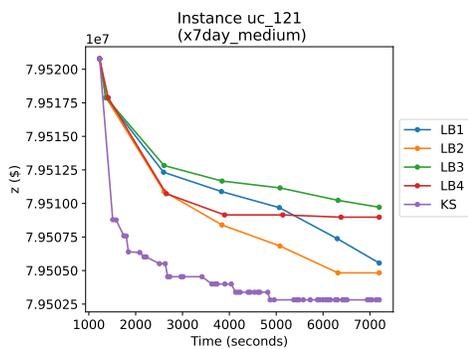


Figure B.31: Comparison of algorithms on instance 121.

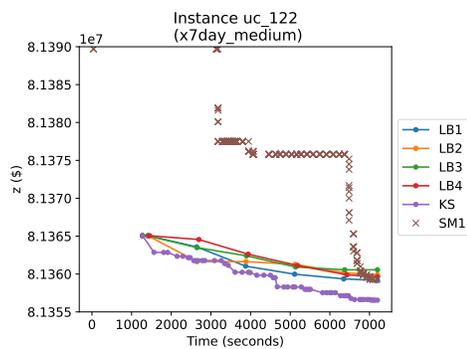


Figure B.32: Comparison of algorithms on instance 122.

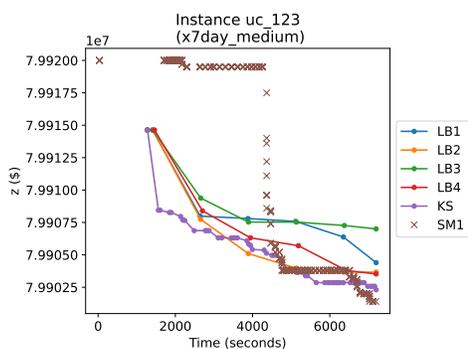


Figure B.33: Comparison of algorithms on instance 123.

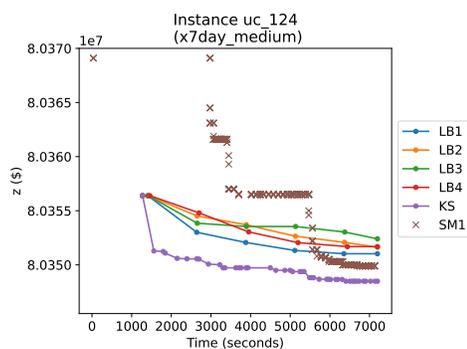


Figure B.34: Comparison of algorithms on instance 124.

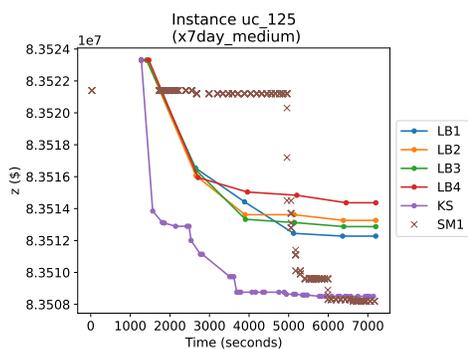


Figure B.35: Comparison of algorithms on instance 125.

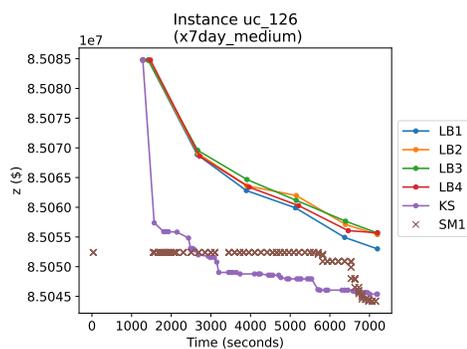


Figure B.36: Comparison of algorithms on instance 126.

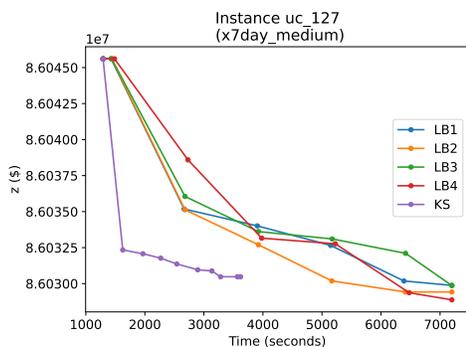


Figure B.37: Comparison of algorithms on instance 127.

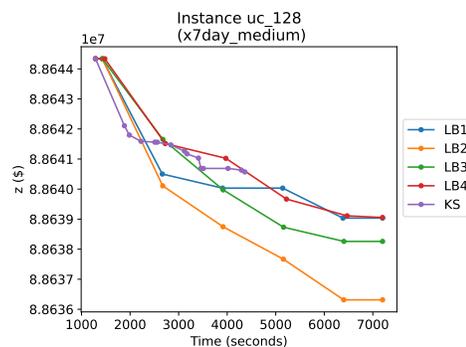


Figure B.38: Comparison of algorithms on instance 128.

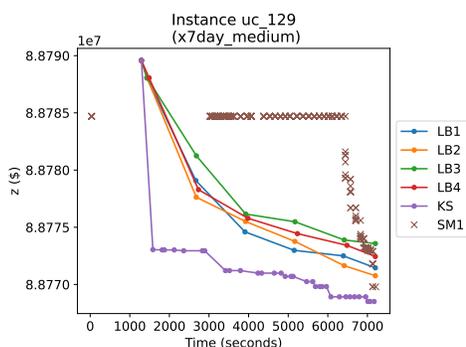


Figure B.39: Comparison of algorithms on instance 129.

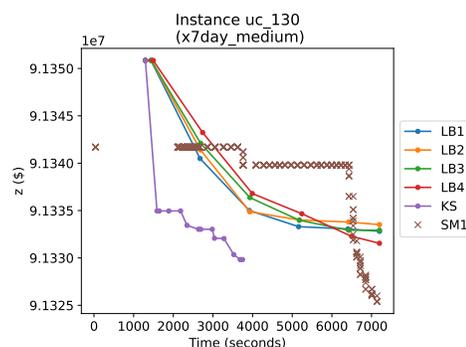


Figure B.40: Comparison of algorithms on instance 130.

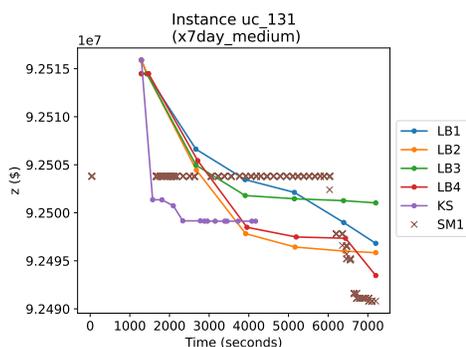


Figure B.41: Comparison of algorithms on instance 131.

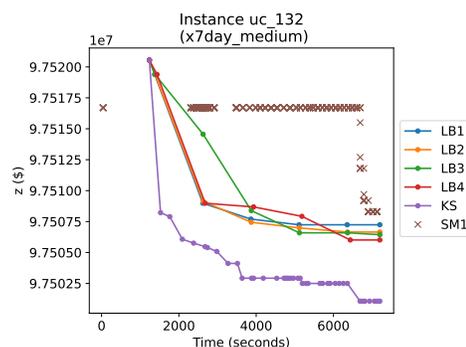


Figure B.42: Comparison of algorithms on instance 132.

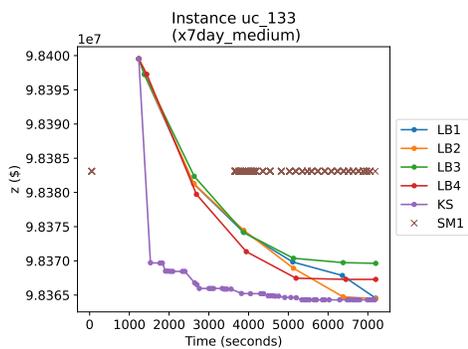


Figure B.43: Comparison of algorithms on instance 133.

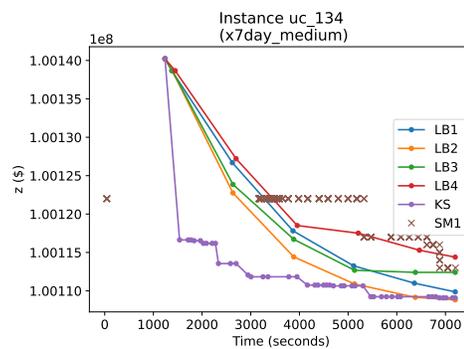


Figure B.44: Comparison of algorithms on instance 134.

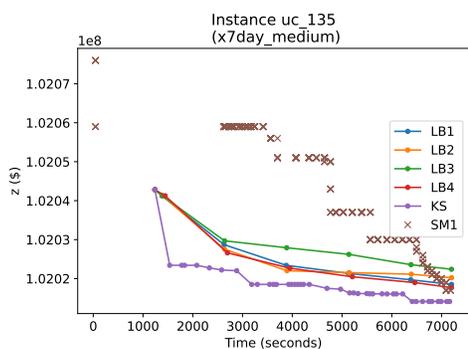


Figure B.45: Comparison of algorithms on instance 135.

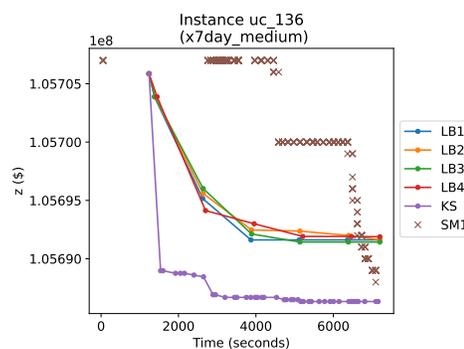


Figure B.46: Comparison of algorithms on instance 136.

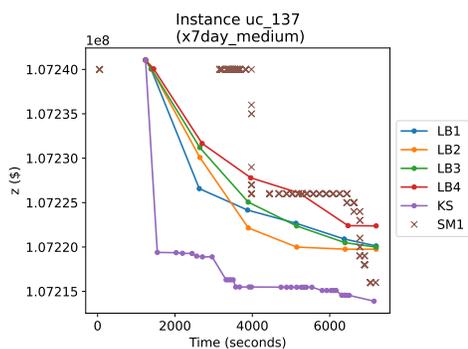


Figure B.47: Comparison of algorithms on instance 137.

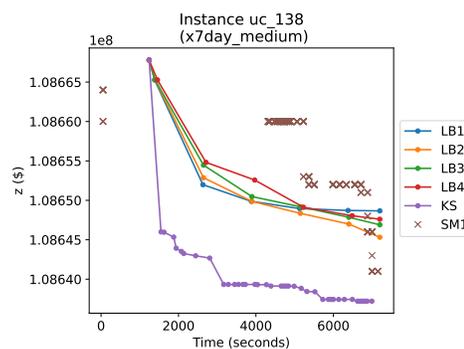


Figure B.48: Comparison of algorithms on instance 138.

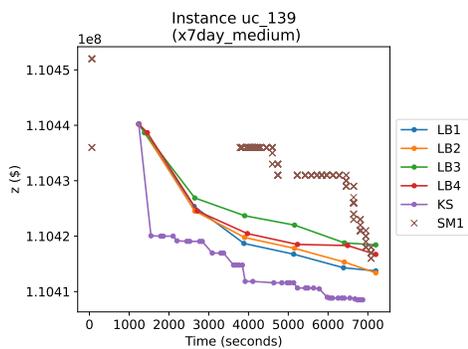


Figure B.49: Comparison of algorithms on instance 139.

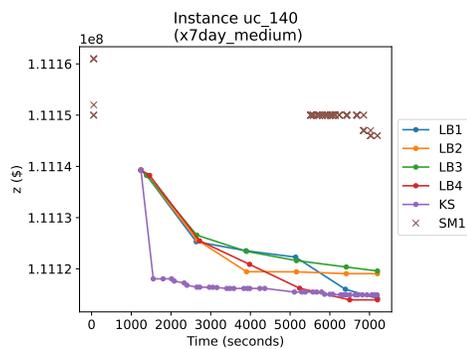


Figure B.50: Comparison of algorithms on instance 140.

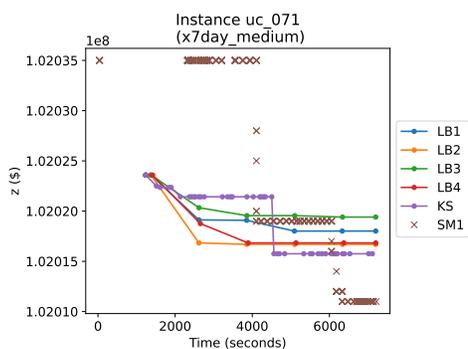


Figure B.51: Comparison of algorithms on instance 071.

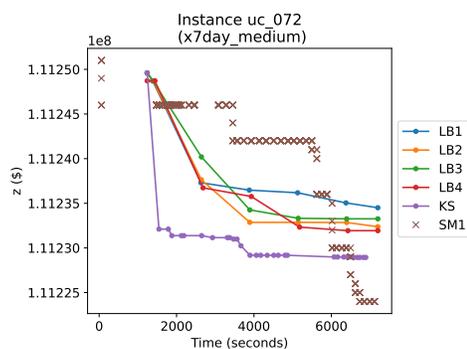


Figure B.52: Comparison of algorithms on instance 072.

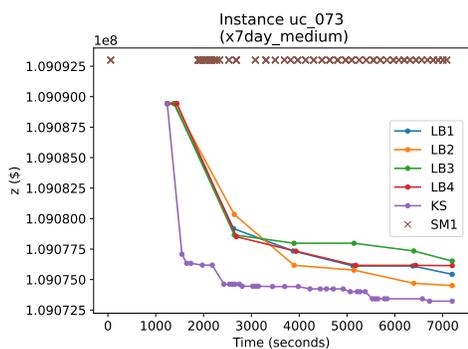


Figure B.53: Comparison of algorithms on instance 073.

### B.3 RESULTS ON INSTANCES FROM GROUP X7DAY\_LARGE

Most of these instances were constructed based on the parameters of the eight generators proposed by [52], seven days, and a number of generators between 165 and 405, files uc\_074-uc\_080 and uc\_141-uc\_163. All files are available in JSON format in the following repository: [https://github.com/urieliram/tc\\_uc](https://github.com/urieliram/tc_uc).

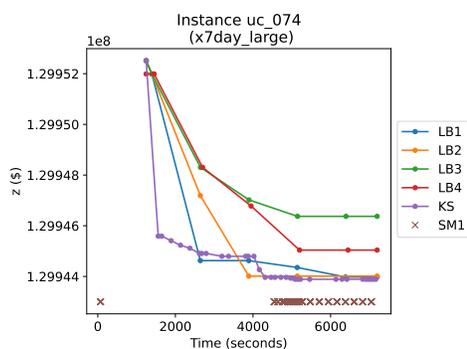


Figure B.54: Comparison of algorithms on instance 074.

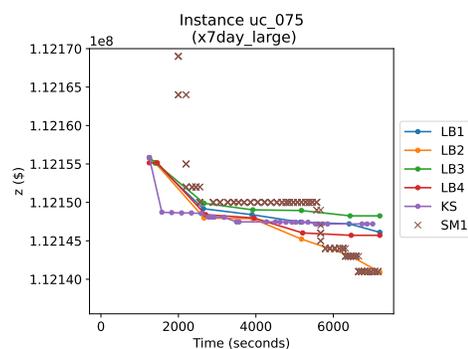


Figure B.55: Comparison of algorithms on instance 075.

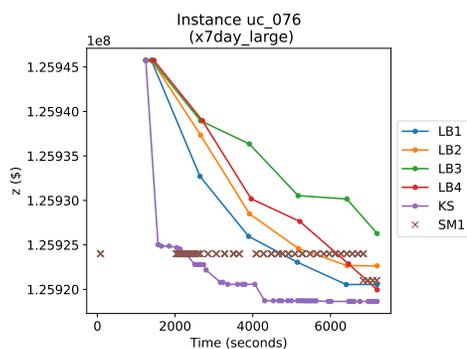


Figure B.56: Comparison of algorithms on instance 076.

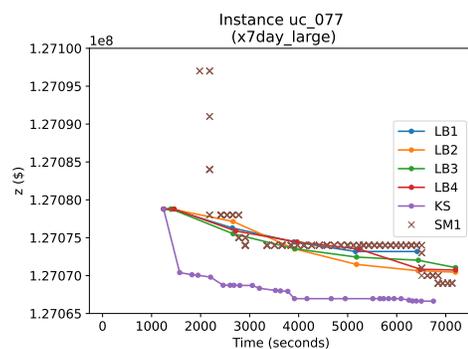


Figure B.57: Comparison of algorithms on instance 077.

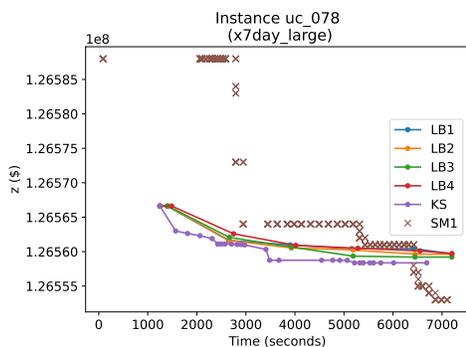


Figure B.58: Comparison of algorithms on instance 078.

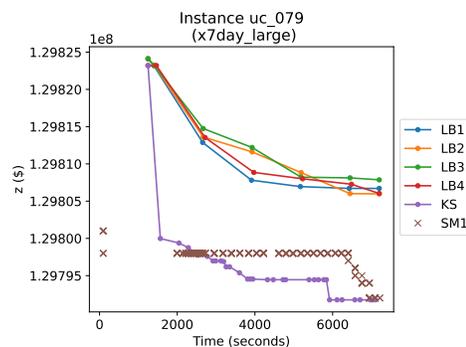


Figure B.59: Comparison of algorithms on instance 079.

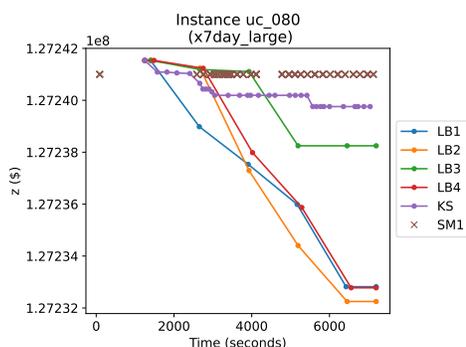


Figure B.60: Comparison of algorithms on instance 080.

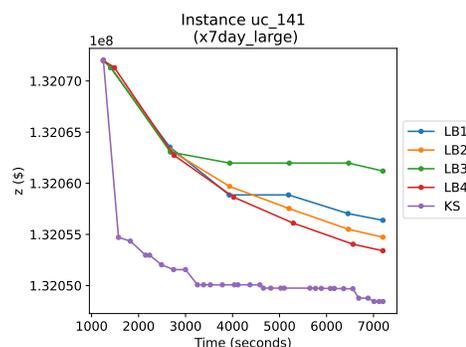


Figure B.61: Comparison of algorithms on instance 141.

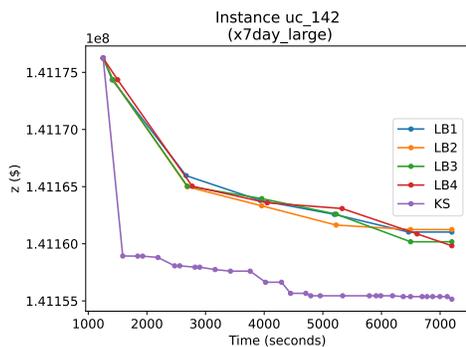


Figure B.62: Comparison of algorithms on instance 142.

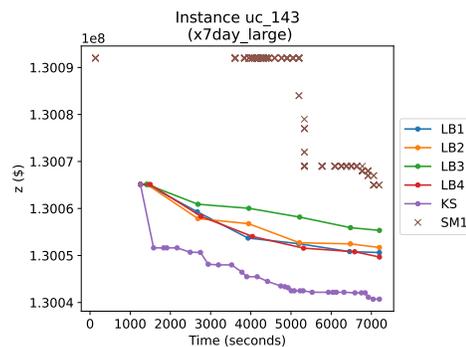


Figure B.63: Comparison of algorithms on instance 143.

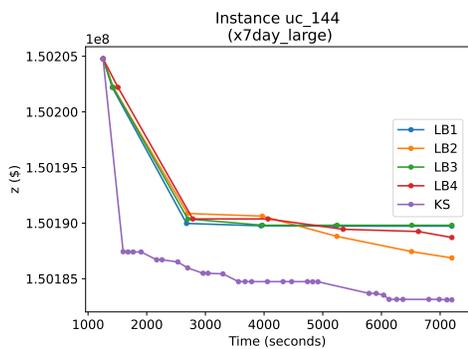


Figure B.64: Comparison of algorithms on instance 144.

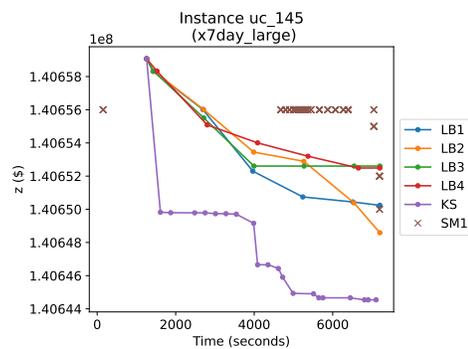


Figure B.65: Comparison of algorithms on instance 145.

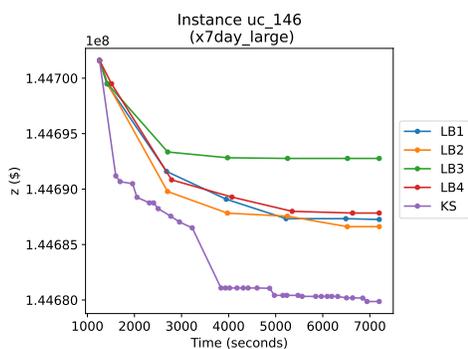


Figure B.66: Comparison of algorithms on instance 146.

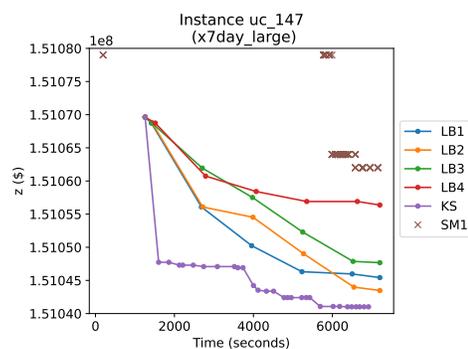


Figure B.67: Comparison of algorithms on instance 147.

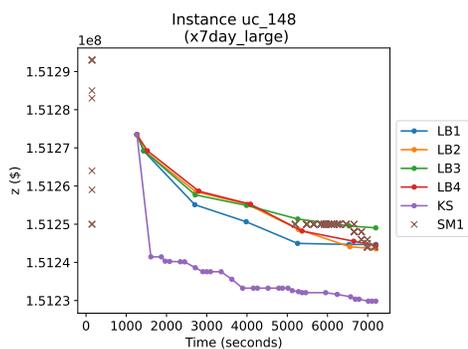


Figure B.68: Comparison of algorithms on instance 148.

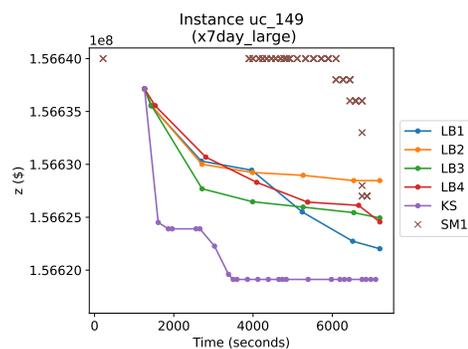


Figure B.69: Comparison of algorithms on instance 149.

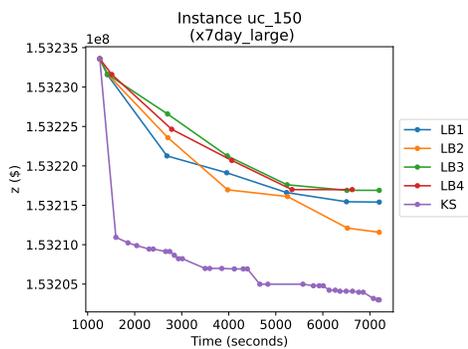


Figure B.70: Comparison of algorithms on instance 150.

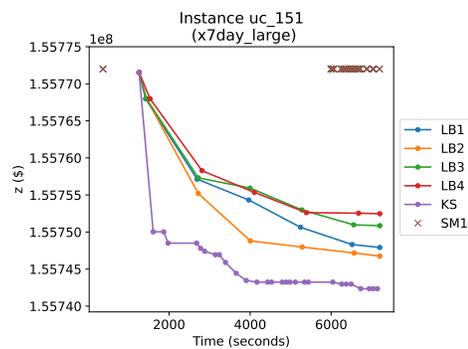


Figure B.71: Comparison of algorithms on instance 151.

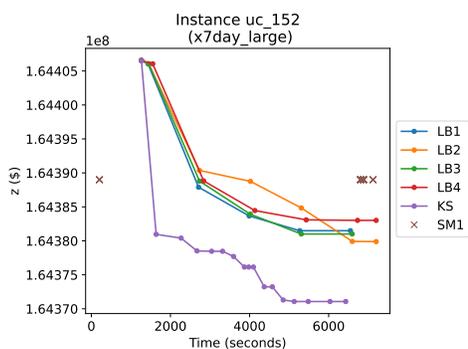


Figure B.72: Comparison of algorithms on instance 152.

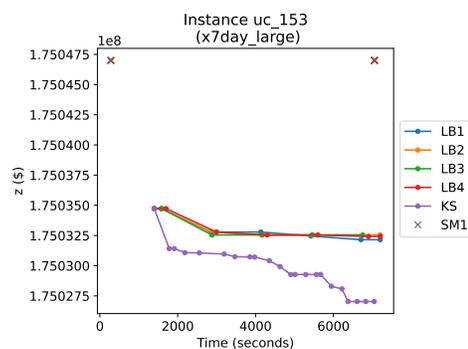


Figure B.73: Comparison of algorithms on instance 153.

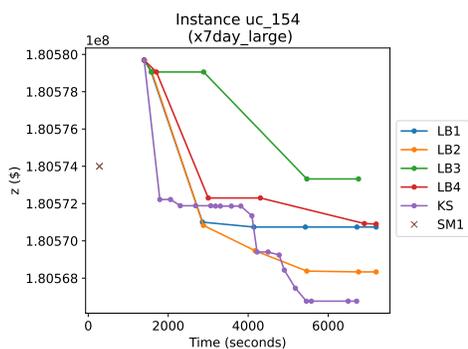


Figure B.74: Comparison of algorithms on instance 154.

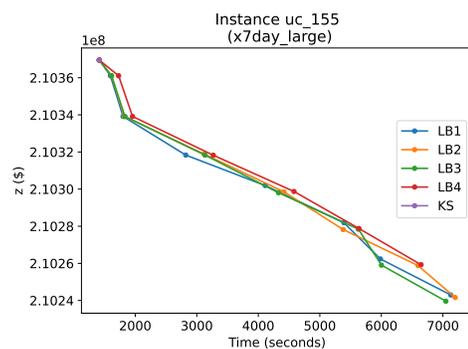


Figure B.75: Comparison of algorithms on instance 155.

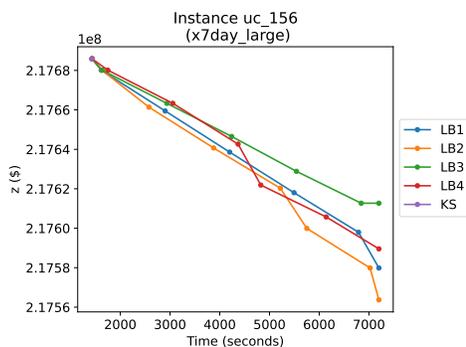


Figure B.76: Comparison of algorithms on instance 156.

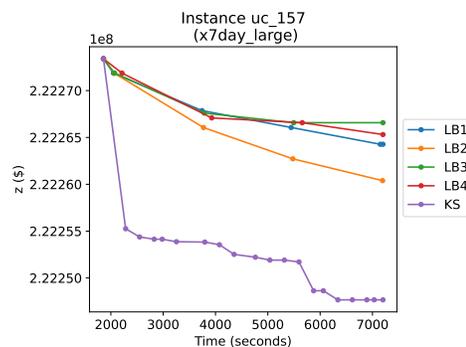


Figure B.77: Comparison of algorithms on instance 157.

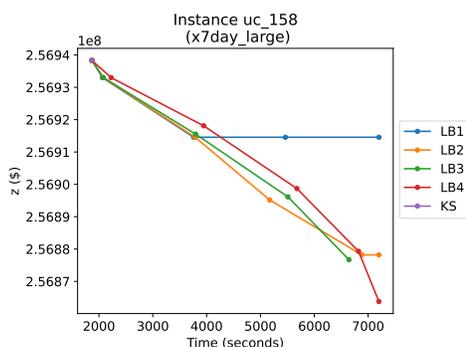


Figure B.78: Comparison of algorithms on instance 158.

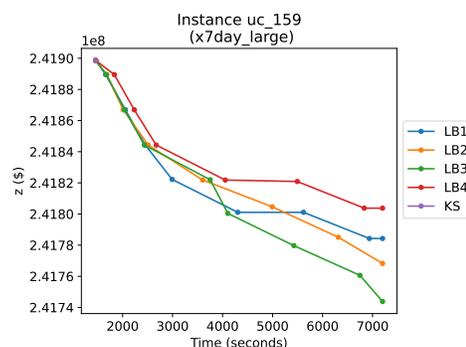


Figure B.79: Comparison of algorithms on instance 159.

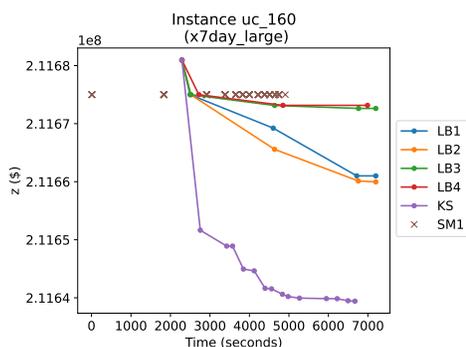


Figure B.80: Comparison of algorithms on instance 160.

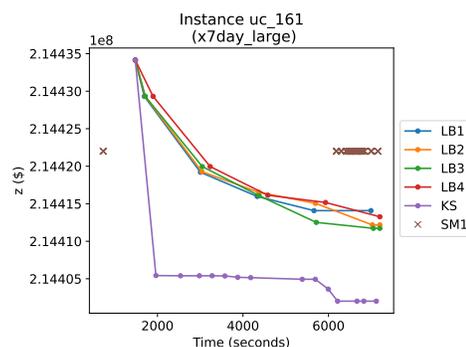


Figure B.81: Comparison of algorithms on instance 161.

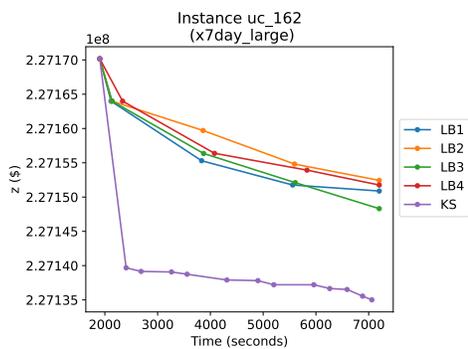


Figure B.82: Comparison of algorithms on instance 162.

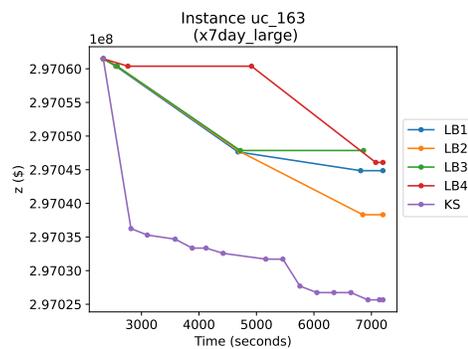


Figure B.83: Comparison of algorithms on instance 163.

APPENDIX C

# STATISTICAL RESULTS

Table C.1: Normality hypothesis test summary for instances in the group x7day\_small.

Null hypothesis	Test	p-value	Decision
HGPS is from the normal distribution	Shapiro-Wilk	1.0000	We fail to reject $H_o$
HARDUC is from the normal distribution	Shapiro-Wilk	0.3069	We fail to reject $H_o$
CBS is from the normal distribution	Shapiro-Wilk	0.0000*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
7200 seconds time limit			
LB1 is from the normal distribution	Shapiro-Wilk	0.1540	We fail to reject $H_o$
LB2 is from the normal distribution	Shapiro-Wilk	0.0360*	We reject $H_o$ and accept $H_a$ : sample is not from the normal distribution
LB3 is from the normal distribution	Shapiro-Wilk	0.2751	We fail to reject $H_o$
LB4 is from the normal distribution	Shapiro-Wilk	0.2282	We fail to reject $H_o$
KS is from the normal distribution	Shapiro-Wilk	0.0786	We fail to reject $H_o$
SM1 is from the normal distribution	Shapiro-Wilk	0.2493	We fail to reject $H_o$
SM2 is from the normal distribution	Shapiro-Wilk	0.1540	We fail to reject $H_o$
4000 seconds time limit			
LB1.1h is from the normal distribution	Shapiro-Wilk	0.0614	We fail to reject $H_o$
LB2.1h is from the normal distribution	Shapiro-Wilk	0.0307*	We reject $H_o$ and accept $H_a$ : sample is not from the normal distribution
LB3.1h is from the normal distribution	Shapiro-Wilk	0.0997	We fail to reject $H_o$
LB4.1h is from the normal distribution	Shapiro-Wilk	0.0441*	We reject $H_o$ and accept $H_a$ : sample is not from the normal distribution
KS.1h is from the normal distribution	Shapiro-Wilk	0.0735	We fail to reject $H_o$
SM1.1h is from the normal distribution	Shapiro-Wilk	0.4482	We fail to reject $H_o$
SM2.1h is from the normal distribution	Shapiro-Wilk	0.2626	We fail to reject $H_o$

\* Significance level 0.05

Table C.2: Normality hypothesis test summary for instances in the group x7day\_medium.

Null hypothesis	Test	p-value	Decision
HGPS is from the normal distribution	Shapiro-Wilk	1.0000	We fail to reject $H_o$
HARDUC is from the normal distribution	Shapiro-Wilk	0.5222	We fail to reject $H_o$
CBS is from the normal distribution	Shapiro-Wilk	0.0015*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
4000 seconds time limit			
LB1_1h is from the normal distribution	Shapiro-Wilk	0.0089*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
LB2_1h is from the normal distribution	Shapiro-Wilk	0.0022*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
LB3_1h is from the normal distribution	Shapiro-Wilk	0.0078*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
LB4_1h is from the normal distribution	Shapiro-Wilk	0.005*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
KS_1h is from the normal distribution	Shapiro-Wilk	0.0152*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM1_1h is from the normal distribution	Shapiro-Wilk	0.0021*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM2_1h is from the normal distribution	Shapiro-Wilk	0.0089*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
7200 seconds time limit			
LB1 is from the normal distribution	Shapiro-Wilk	0.0187*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
LB2 is from the normal distribution	Shapiro-Wilk	0.0138*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
LB3 is from the normal distribution	Shapiro-Wilk	0.2274	We fail to reject $H_o$
LB4 is from the normal distribution	Shapiro-Wilk	0.0587	We fail to reject $H_o$
KS is from the normal distribution	Shapiro-Wilk	0.0038*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM1 is from the normal distribution	Shapiro-Wilk	0.0188*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM2 is from the normal distribution	Shapiro-Wilk	0.7656	We fail to reject $H_o$

\* Significance level 0.05

Table C.3: Normality hypothesis test summary for instances in the group x7day\_large.

Null hypothesis	Test	p-value	Decision
HGPS is from the normal distribution	Shapiro-Wilk	1.00000	We fail to reject $H_o$
HARDUC is from the normal distribution	Shapiro-Wilk	0.75220	We fail to reject $H_o$
CBS is from the normal distribution	Shapiro-Wilk	0.0000*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
7200 seconds time limit			
LB1 is from the normal distribution	Shapiro-Wilk	0.73600	We fail to reject $H_o$
LB2 is from the normal distribution	Shapiro-Wilk	0.87180	We fail to reject $H_o$
LB3 is from the normal distribution	Shapiro-Wilk	0.85790	We fail to reject $H_o$
LB4 is from the normal distribution	Shapiro-Wilk	0.92370	We fail to reject $H_o$
KS is from the normal distribution	Shapiro-Wilk	0.12470	We fail to reject $H_o$
SM1 is from the normal distribution	Shapiro-Wilk	0.0080*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM2 is from the normal distribution	Shapiro-Wilk	0.73600	We fail to reject $H_o$
4000 seconds time limit			
LB1.1h is from the normal distribution	Shapiro-Wilk	0.85130	We fail to reject $H_o$
LB2.1h is from the normal distribution	Shapiro-Wilk	0.84750	We fail to reject $H_o$
LB3.1h is from the normal distribution	Shapiro-Wilk	0.31810	We fail to reject $H_o$
LB4.1h is from the normal distribution	Shapiro-Wilk	0.46750	We fail to reject $H_o$
KS.1h is from the normal distribution	Shapiro-Wilk	0.03830*	We reject $H_o$ and accept $H_a$ : sample not from a normal distribution
SM1.1h is from the normal distribution	Shapiro-Wilk	0.46900	We fail to reject $H_o$
SM2.1h is from the normal distribution	Shapiro-Wilk	0.60920	We fail to reject $H_o$

\* Significance level 0.05

Table C.4: Analysis of variance summary of the constructive methods HARDUC, HGPS, and CBS.

Null hypothesis	Instances	Test	p-value	Decision
The mean for each population is equal	x7day_small	Kruskal-Wallis	0.0000*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest
The mean for each population is equal	x7day_medium	Kruskal-Wallis	0.0000*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest
The mean for each population is equal	x7day_large	Kruskal-Wallis	0.0000*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest

\* Significance level 0.05

Table C.5: Differences mean test summary between the constructive methods HARDUC and HGPS.

Null hypothesis	Instances	Test	p-value	Decision
The means difference of the samples from the same distribution	x7day_small	Mann-Whitney	0.0002*	We reject $H_o$ and accept $H_a$ : HARDUC's mean is less than HGPS's mean
The means difference of the samples from the same distribution	x7day_medium	Mann-Whitney	0.0000*	We reject $H_o$ and accept $H_a$ : HARDUC's mean is less than HGPS's mean
The means difference of the samples from the same distribution	x7day_large	Mann-Whitney	0.0000*	We reject $H_o$ and accept $H_a$ : HARDUC's mean is less than HGPS's mean

\* Significance level 0.05

Table C.6: Analysis of variance summary for LB1\_1h, LB2\_1h, LB3\_1h, LB4\_1h, KS\_1h, SM1\_1h, SM2\_1h under a running time limit of 4000 seconds.

Null hypothesis	Instances	Test	p-value	Decision
The mean for each population is equal	x7day_small	Kruskal-Wallis	0.9758	We fail to reject $H_o$
The mean for each population is equal	x7day_medium	Kruskal-Wallis	0.0000*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest
The mean for each population is equal	x7day_large	Kruskal-Wallis	0.0066*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest

\* Significance level 0.05

Table C.7: Analysis of variance summary for LB1, LB2, LB3, LB4, KS, MILP, SM2 under a running time limit of 7200 seconds.

Null hypothesis	Instances	Test	p-value	Decision
The mean for each population is equal	x7day_small	Kruskal-Wallis	0.7289	We fail to reject $H_o$
The mean for each population is equal	x7day_medium	Kruskal-Wallis	0.0000*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest
The mean for each population is equal	x7day_large	Kruskal-Wallis	0.0007*	We reject $H_o$ and accept $H_a$ : at least one population mean different from the rest

\* Significance level 0.05

Table C.8: Means difference statistical test summary among all methods for instances from group x7day\_small under a running time limit of 4000 seconds.

Null hypothesis	Test	p-value	Decision
SM1.1h-LB2.1h: There is no difference between the two population means	Mann-Whitney	0.3779	We fail to reject $H_0$
SM1.1h-KS.1h: There is no difference between the two population means	T-test for two samples	0.2725	We fail to reject $H_0$
SM1.1h-LB1.1h: There is no difference between the two population means	T-test for two samples	0.2499	We fail to reject $H_0$
SM1.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	0.1737	We fail to reject $H_0$
SM1.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.1318	We fail to reject $H_0$
SM1.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	0.0888	We fail to reject $H_0$
LB2.1h-KS.1h: There is no difference between the two population means	Mann-Whitney	0.5484	We fail to reject $H_0$
LB2.1h-LB1.1h: There is no difference between the two population means	Mann-Whitney	0.4302	We fail to reject $H_0$
LB2.1h-LB4.1h: There is no difference between the two population means	Mann-Whitney	0.2714	We fail to reject $H_0$
LB2.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	0.2367	We fail to reject $H_0$
LB2.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	0.1617	We fail to reject $H_0$
KS.1h-LB1.1h: There is no difference between the two population means	T-test for two samples	0.492	We fail to reject $H_0$
KS.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	0.4016	We fail to reject $H_0$
KS.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.3336	We fail to reject $H_0$
KS.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	0.2923	We fail to reject $H_0$
LB1.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	0.4039	We fail to reject $H_0$
LB1.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.332	We fail to reject $H_0$
LB1.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	0.287	We fail to reject $H_0$
LB4.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.4211	We fail to reject $H_0$
LB4.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	0.3808	We fail to reject $H_0$
LB3.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	0.4675	We fail to reject $H_0$

\* Significance level 0.05

Table C.9: Means difference statistical test summary among all methods for instances from group x7day\_small under a running time limit of 7200 seconds.

Null hypothesis	Test	p-value	Decision
SM1-LB2: There is no difference between the two population means	Mann-Whitney	0.1584	We fail to reject $H_o$
SM1-LB1: There is no difference between the two population means	Mann-Whitney	0.1396	We fail to reject $H_o$
SM1-LB4: There is no difference between the two population means	Mann-Whitney	0.117	We fail to reject $H_o$
SM1-SM2: There is no difference between the two population means	T-test for two samples	*0.0362	We reject $H_o$ and accept $H_a$ : MILP's mean less than MILP2's
SM1-KS: There is no difference between the two population means	T-test for two samples	0.0501	We fail to reject $H_o$
SM1-LB3: There is no difference between the two population means	T-test for two samples	*0.0358	We reject $H_o$ and accept $H_a$ : MILP's mean less than LB3's
LB2-LB1: There is no difference between the two population means	Mann-Whitney	0.4569	We fail to reject $H_o$
LB2-LB4: There is no difference between the two population means	Mann-Whitney	0.4143	We fail to reject $H_o$
LB2-SM2: There is no difference between the two population means	Mann-Whitney	0.2326	We fail to reject $H_o$
LB2-KS: There is no difference between the two population means	Mann-Whitney	0.4091	We fail to reject $H_o$
LB2-LB3: There is no difference between the two population means	Mann-Whitney	0.2989	We fail to reject $H_o$
LB1-LB4: There is no difference between the two population means	Mann-Whitney	0.4515	We fail to reject $H_o$
LB1-SM2: There is no difference between the two population means	Mann-Whitney	0.2669	We fail to reject $H_o$
LB1-KS: There is no difference between the two population means	Mann-Whitney	0.4623	We fail to reject $H_o$
LB1-LB3: There is no difference between the two population means	Mann-Whitney	0.2896	We fail to reject $H_o$
LB4-SM2: There is no difference between the two population means	Mann-Whitney	0.3036	We fail to reject $H_o$
LB4-KS: There is no difference between the two population means	Mann-Whitney	0.4623	We fail to reject $H_o$
LB4-LB3: There is no difference between the two population means	Mann-Whitney	0.3375	We fail to reject $H_o$
SM2-KS: There is no difference between the two population means	T-test for two samples	0.3651	We fail to reject $H_o$
SM2-LB3: There is no difference between the two population means	T-test for two samples	0.3434	We fail to reject $H_o$
KS-LB3: There is no difference between the two population means	T-test for two samples	0.4909	We fail to reject $H_o$

\* Significance level 0.05

Table C.10: Means difference statistical test summary among all methods for instances from group x7day\_medium under a running time limit of 4000 seconds.

Null hypothesis	Test	p-value	Decision
KS.1h-LB2.1h: There is no difference between the two population means	Mann-Whitney	*0.0108	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB2.1h's
KS.1h-LB1.1h: There is no difference between the two population means	Mann-Whitney	*0.0031	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB1.1h's
KS.1h-LB4.1h: There is no difference between the two population means	Mann-Whitney	*0.0022	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB4.1h's
KS.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	*0.001	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB3.1h's
KS.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than SM1.1h's
KS.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than SM2.1h's
LB2.1h-LB1.1h: There is no difference between the two population means	Mann-Whitney	0.2699	We fail to reject $H_o$
LB2.1h-LB4.1h: There is no difference between the two population means	Mann-Whitney	0.2405	We fail to reject $H_o$
LB2.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	0.1593	We fail to reject $H_o$
LB2.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0002	We reject $H_o$ and accept $H_a$ : LB2.1h's mean less than SM1.1h's
LB2.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0002	We reject $H_o$ and accept $H_a$ : LB2.1h's mean less than SM2.1h's
LB1.1h-LB4.1h: There is no difference between the two population means	Mann-Whitney	0.4745	We fail to reject $H_o$
LB1.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	0.3538	We fail to reject $H_o$
LB1.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0003	We reject $H_o$ and accept $H_a$ : LB1.1h's mean less than SM1.1h's
LB1.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0003	We reject $H_o$ and accept $H_a$ : LB1.1h's mean less than SM2.1h's
LB4.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	0.3606	We fail to reject $H_o$
LB4.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0005	We reject $H_o$ and accept $H_a$ : LB4.1h's mean less than SM1.1h's
LB4.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0004	We reject $H_o$ and accept $H_a$ : LB4.1h's mean less than SM2.1h's
LB3.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0009	We reject $H_o$ and accept $H_a$ : LB3.1h's mean less than SM1.1h's
LB3.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0005	We reject $H_o$ and accept $H_a$ : LB3.1h's mean less than SM2.1h's
SM1.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	0.2699	We fail to reject $H_o$

\* Significance level 0.05

Table C.11: Means difference statistical test summary among all methods for instances from group x7day\_medium under a running time limit of 7200 seconds.

Null hypothesis	Test	p-value	Decision
KS-SM1: There is no difference between the two population means	Mann-Whitney	0.3813	We fail to reject $H_o$
KS-LB1: There is no difference between the two population means	Mann-Whitney	*0.0165	We reject $H_o$ and accept $H_a$ : KS's mean less than LB1's
KS-LB4: There is no difference between the two population means	Mann-Whitney	*0.0125	We reject $H_o$ and accept $H_a$ : KS's mean less than LB4's
KS-LB2: There is no difference between the two population means	Mann-Whitney	*0.0131	We reject $H_o$ and accept $H_a$ : KS's mean less than LB2's
KS-LB3: There is no difference between the two population means	Mann-Whitney	*0.0012	We reject $H_o$ and accept $H_a$ : KS's mean less than LB3's
KS-SM2: There is no difference between the two population means	Mann-Whitney	*0.0	We reject $H_o$ and accept $H_a$ : KS's mean less than MILP2's
SM1-LB1: There is no difference between the two population means	Mann-Whitney	0.061	We fail to reject $H_o$
SM1-LB4: There is no difference between the two population means	Mann-Whitney	0.061	We fail to reject $H_o$
SM1-LB2: There is no difference between the two population means	Mann-Whitney	0.0691	We fail to reject $H_o$
SM1-LB3: There is no difference between the two population means	Mann-Whitney	*0.0103	We reject $H_o$ and accept $H_a$ : MILP's mean less than LB3's
SM1-SM2: There is no difference between the two population means	Mann-Whitney	*0.0	We reject $H_o$ and accept $H_a$ : MILP's mean less than MILP2's
LB1-LB4: There is no difference between the two population means	Mann-Whitney	0.4418	We fail to reject $H_o$
LB1-LB2: There is no difference between the two population means	Mann-Whitney	0.5219	We fail to reject $H_o$
LB1-LB3: There is no difference between the two population means	Mann-Whitney	0.1463	We fail to reject $H_o$
LB1-SM2: There is no difference between the two population means	Mann-Whitney	*0.0001	We reject $H_o$ and accept $H_a$ : LB1's mean less than MILP2's
LB4-LB2: There is no difference between the two population means	Mann-Whitney	0.551	We fail to reject $H_o$
LB4-LB3: There is no difference between the two population means	T-test for two samples	0.2583	We fail to reject $H_o$
LB4-SM2: There is no difference between the two population means	T-test for two samples	*0.0001	We reject $H_o$ and accept $H_a$ : LB4's mean less than MILP2's
LB2-LB3: There is no difference between the two population means	Mann-Whitney	0.1638	We fail to reject $H_o$
LB2-SM2: There is no difference between the two population means	Mann-Whitney	*0.0002	We reject $H_o$ and accept $H_a$ : LB2's mean less than MILP2's
LB3-SM2: There is no difference between the two population means	Mann-Whitney	*0.0004	We reject $H_o$ and accept $H_a$ : LB3's mean less than MILP2's

\* Significance level 0.05

Table C.12: Means difference statistical test summary among all methods for instances from group x7day\_large under a running time limit of 4000 seconds.

Null hypothesis	Test	p-value	Decision
KS.1h-LB1.1h: There is no difference between the two population means	T-test for two samples	*0.0071	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB1.1h's
KS.1h-LB2.1h: There is no difference between the two population means	T-test for two samples	*0.0038	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB2.1h's
KS.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	*0.0005	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB4.1h's
KS.1h-LB3.1h: There is no difference between the two population means	Mann-Whitney	*0.0021	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than LB3.1h's
KS.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0003	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than SM1.1h's
KS.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	*0.0001	We reject $H_o$ and accept $H_a$ : KS.1h's mean less than SM2.1h's
LB1.1h-LB2.1h: There is no difference between the two population means	T-test for two samples	0.2958	We fail to reject $H_o$
LB1.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	0.1441	We fail to reject $H_o$
LB1.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.1603	We fail to reject $H_o$
LB1.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	*0.0425	We reject $H_o$ and accept $H_a$ : LB1.1h's mean less than SM1.1h's
LB1.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	*0.0033	We reject $H_o$ and accept $H_a$ : LB1.1h's mean less than SM2.1h's
LB2.1h-LB4.1h: There is no difference between the two population means	T-test for two samples	0.3234	We fail to reject $H_o$
LB2.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.329	We fail to reject $H_o$
LB2.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	0.1208	We fail to reject $H_o$
LB2.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	*0.0155	We reject $H_o$ and accept $H_a$ : LB2.1h's mean less than SM2.1h's
LB4.1h-LB3.1h: There is no difference between the two population means	T-test for two samples	0.4934	We fail to reject $H_o$
LB4.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	0.2347	We fail to reject $H_o$
LB4.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	*0.032	We reject $H_o$ and accept $H_a$ : LB4.1h's mean less than SM2.1h's
LB3.1h-SM1.1h: There is no difference between the two population means	Mann-Whitney	0.1853	We fail to reject $H_o$
LB3.1h-SM2.1h: There is no difference between the two population means	T-test for two samples	*0.0408	We reject $H_o$ and accept $H_a$ : LB3.1h's mean less than SM2.1h's
SM1.1h-SM2.1h: There is no difference between the two population means	Mann-Whitney	0.1208	We fail to reject $H_o$

\* Significance level 0.05

Table C.13: Means difference statistical test summary among all methods for instances from group x7day\_large under a running time limit of 7200 seconds.

Null hypothesis	Test	p-value	Decision
KS-LB2: There is no difference between the two population means	Mann-Whitney	*0.0164	We reject $H_o$ and accept $H_a$ : KS's mean less than LB2's
KS-LB1: There is no difference between the two population means	Mann-Whitney	*0.0105	We reject $H_o$ and accept $H_a$ : KS's mean less than LB1's
KS-LB4: There is no difference between the two population means	Mann-Whitney	*0.0072	We reject $H_o$ and accept $H_a$ : KS's mean less than LB4's
KS-LB3: There is no difference between the two population means	Mann-Whitney	*0.0019	We reject $H_o$ and accept $H_a$ : KS's mean less than LB3's
KS-SM1: There is no difference between the two population means	Mann-Whitney	*0.0021	We reject $H_o$ and accept $H_a$ : KS's mean less than MILP's
KS-SM2: There is no difference between the two population means	Mann-Whitney	*0.0001	We reject $H_o$ and accept $H_a$ : KS's mean less than MILP2's
LB2-LB1: There is no difference between the two population means	T-test for two samples	0.3812	We fail to reject $H_o$
LB2-LB4: There is no difference between the two population means	T-test for two samples	0.2823	We fail to reject $H_o$
LB2-LB3: There is no difference between the two population means	T-test for two samples	0.1417	We fail to reject $H_o$
LB2-SM1: There is no difference between the two population means	T-test for two samples	0.0858	We fail to reject $H_o$
LB2-SM2: There is no difference between the two population means	T-test for two samples	*0.0012	We reject $H_o$ and accept $H_a$ : LB2's mean less than MILP2's
LB1-LB4: There is no difference between the two population means	T-test for two samples	0.3952	We fail to reject $H_o$
LB1-LB3: There is no difference between the two population means	T-test for two samples	0.2175	We fail to reject $H_o$
LB1-SM1: There is no difference between the two population means	T-test for two samples	0.1337	We fail to reject $H_o$
LB1-SM2: There is no difference between the two population means	T-test for two samples	*0.0022	We reject $H_o$ and accept $H_a$ : LB1's mean less than MILP2's
LB4-LB3: There is no difference between the two population means	T-test for two samples	0.2932	We fail to reject $H_o$
LB4-SM1: There is no difference between the two population means	T-test for two samples	0.1826	We fail to reject $H_o$
LB4-SM2: There is no difference between the two population means	T-test for two samples	*0.0032	We reject $H_o$ and accept $H_a$ : LB4's mean less than MILP2's
LB3-SM1: There is no difference between the two population means	T-test for two samples	0.3414	We fail to reject $H_o$
LB3-SM2: There is no difference between the two population means	T-test for two samples	*0.0101	We reject $H_o$ and accept $H_a$ : LB3's mean less than MILP2's
SM1-SM2: There is no difference between the two population means	T-test for two samples	*0.0290	We reject $H_o$ and accept $H_a$ : MILP's mean less than MILP2's

\* Significance level 0.05

APPENDIX D

LOAD DEMAND FORECASTING RESULTS

---

Table D.1: Results of the Levene test among methods with and without correction with moving average (MA) under a 2-hours forecast.

Method 1	Method 2	Null Hypothesis	Test	p-value	Decision
XAnMA_PCR	XAn_PCR	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_Lasso	XAn_Lasso	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_Ridge	XAn_Ridge	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_OLS	XAn_OLS	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_Lasso_euc	XAn_Lasso_euc	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_Ridge_euc	XAn_Ridge_euc	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_OLS_euc	XAn_OLS_euc	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_RF	XAn_RF	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_PLS	XAn_PLS	The variances are equal between samples	Levene	1.0000	We fail to reject $H_o$ ; $H_o$ accepted
XAnMA_Boost	XAn_Boost	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples
XAnMA_Bagg	XAn_Bagg	The variances are equal between samples	Levene	*0.0000	We reject $H_o$ and accept $H_a$ : the variances are not equal between samples

\* Significance level 0.05

Table D.2: Summary of results of the first test under a five-minute forecast.

Method	MAE	MAPE	Mean time (s)
AnMA_PCR	6.81e+0	2.51e-1	1.60e+0
AnMA_Lasso	7.83e+0	2.89e-1	1.53e+0
AnMA_Ridge	7.84e+0	2.89e-1	1.62e+0
AnMA_OLS	8.19e+0	3.03e-1	1.50e+0
AnMA_Lasso_euc	8.79e+0	3.25e-1	1.97e+0
AnMA_Ridge_euc	8.79e+0	3.25e-1	1.97e+0
AnMA_OLS_euc	8.89e+0	3.29e-1	1.82e+0
Persistent	9.03e+0	3.31e-1	0.00e+0
HWM	9.82e+0	3.60e-1	5.58e+0
HWA	9.91e+0	3.64e-1	3.55e+0
AnMA_RF	1.05e+1	3.90e-1	1.61e+0
AnMA_Boost	1.21e+1	4.54e-1	1.53e+0
AnMA_RF_euc	1.23e+1	4.58e-1	1.83e+0
AnMA_Bagg	1.26e+1	4.69e-1	1.59e+0
AnMA_Boost_euc	1.43e+1	5.33e-1	1.79e+0
AnMA_Bagg_euc	1.49e+1	5.54e-1	1.98e+0
An_RF	2.25e+1	8.34e-1	1.61e+0
An_Boost	2.26e+1	8.35e-1	1.53e+0
An_Bagg	2.33e+1	8.62e-1	1.59e+0
An_RF_euc	2.78e+1	1.03e+0	1.83e+0
An_PLS_euc	2.81e+1	1.04e+0	2.01e+0
An_Boost_euc	2.81e+1	1.04e+0	1.79e+0
AnMA_PLS_euc	2.81e+1	1.04e+0	2.01e+0
An_Bagg_euc	2.88e+1	1.07e+0	1.98e+0
An_Ridge	3.02e+1	1.11e+0	1.62e+0
An_Lasso	3.02e+1	1.11e+0	1.53e+0
AnMA_PLS	3.16e+1	1.16e+0	1.57e+0
An_PLS	3.16e+1	1.16e+0	1.56e+0
An_OLS	3.24e+1	1.19e+0	1.50e+0
An_Ridge_euc	3.42e+1	1.27e+0	1.97e+0
An_Lasso_euc	3.42e+1	1.27e+0	1.97e+0
An_PCR	3.53e+1	1.30e+0	1.60e+0
An_OLS_euc	3.63e+1	1.35e+0	1.82e+0

Table D.3: Summary of results of the second test under a two-and-a-half-hour forecast.

Method	MAE	MAPE	Mean time (s)
XAnMA_PCR	3.06e+1	1.13e+0	1.60e+0
XAnMA_Lasso	3.25e+1	1.20e+0	1.53e+0
XAnMA_Ridge	3.25e+1	1.20e+0	1.62e+0
XAnMA_OLS	3.25e+1	1.20e+0	1.50e+0
XAnMA_Lasso_euc	3.49e+1	1.30e+0	1.97e+0
XAnMA_Ridge_euc	3.49e+1	1.30e+0	1.97e+0
XAnMA_OLS_euc	3.50e+1	1.31e+0	1.82e+0
XHWA	3.77e+1	1.37e+0	3.54e+0
XHWM	3.97e+1	1.43e+0	5.59e+0
XAn_Lasso	4.79e+1	1.76e+0	1.53e+0
XAn_Ridge	4.80e+1	1.76e+0	1.62e+0
XAnMA_RF	4.81e+1	1.77e+0	1.62e+0
XAn_PLS	4.84e+1	1.78e+0	1.57e+0
XAnMA_PLS	4.84e+1	1.78e+0	1.57e+0
XAn_OLS	4.86e+1	1.78e+0	1.50e+0
XAnMA_Bagg	4.84e+1	1.79e+0	1.60e+0
XAnMA_Boost	4.88e+1	1.80e+0	1.53e+0
XAn_PCR	5.13e+1	1.89e+0	1.60e+0
XAn_Lasso_euc	5.33e+1	1.98e+0	1.97e+0
XAn_Ridge_euc	5.33e+1	1.98e+0	1.97e+0
XAn_euc	5.34e+1	1.98e+0	1.82e+0
XAnMA_RF_euc	5.35e+1	1.99e+0	1.83e+0
XAnMA_Boost_euc	5.42e+1	2.01e+0	1.79e+0
XAnMA_Bagg_euc	5.47e+1	2.03e+0	1.99e+0
XAn_Boost	6.02e+1	2.21e+0	1.53e+0
XAn_RF	6.03e+1	2.22e+0	1.62e+0
XAn_Bagg	6.06e+1	2.23e+0	1.59e+0
XAnMA_PLS_euc	6.89e+1	2.56e+0	2.01e+0
XAn_Boost_euc	6.89e+1	2.56e+0	1.79e+0
XAn_PLS_euc	6.89e+1	2.56e+0	2.01e+0
XAn_RF_euc	6.91e+1	2.56e+0	1.83e+0
XAn_Bagg_euc	6.95e+1	2.57e+0	1.99e+0

# BIBLIOGRAPHY

---

- [1] I. Abdou and M. Tkiouat. Unit commitment problem in electrical power system: A literature review. *International Journal of Electrical and Computer Engineering*, 8(3):1357–1372, 2018.
- [2] T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.
- [3] S. Alessandrini, L. Delle Monache, S. Sperati, and G. Cervone. An analog ensemble for short-term probabilistic solar power forecast. *Applied Energy*, 157:95–110, 2015.
- [4] S. Alessandrini, L. Delle Monache, S. Sperati, and J.N. Nissen. A novel application of an analog ensemble for short-term wind power forecasting. *Renewable Energy*, 76:768–781, 2015.
- [5] E. Angelelli, R. Mansini, and M. Grazia Speranza. Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11):2017–2026, 2010.
- [6] M. F. Anjos. Recent progress in modeling unit commitment problems. In L. F. Zuluaga and T. Terlaky, editors, *Modeling and Optimization: Theory and Applications*, pages 1–29, New York, USA, October 2013. Springer.
- [7] M. F. Anjos and A. J. Conejo. Unit commitment in electric energy systems. *Foundations and Trends in Electric Energy Systems*, 1(4):220–310, 2017.
- [8] A. Azeem, I. Ismail, S. M. Jameel, and V. R. Harindran. Electrical load forecasting models for different generation modalities: A review. *IEEE Access*, 9:142239–142263, 2021.

- 
- [9] M. Azevedo, R. Ruiz-Cárdenas, L. Brioschi, and M. Oliveira. Dynamic time scan forecasting for multi-step wind speed prediction. *Renewable Energy*, 177:584–595, 2021.
- [10] P. Bendotti, P. Fouilhoux, and C. Rottner. On the complexity of the unit commitment problem. *Annals of Operations Research*, 274(1):119–130, 2019.
- [11] C. Bergmeir, R. J. Hyndman, and B. Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, 2018.
- [12] T. Berthold. *Primal Heuristics for Mixed Integer Programs*. PhD thesis, Konrad-Zuse-Zentrum für Information Stechnik Berlin, Berlin, Germany, January 2006.
- [13] T. Berthold, A. Lodi, and D. Salvagnin. Ten years of feasibility pump, and counting. *EURO Journal on Computational Optimization*, 7(1):1–14, 2019.
- [14] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, New York, USA, 5th edition, 2015.
- [15] M. L. Bynum, G. A. Hackebeil, W. E. Hart, C. D. Laird, B. L. Nicholson, J. D. Sirola, Jean-Paul Watson, and D. L. Woodruff. *Pyomo—optimization modeling in python*, volume 67 of *Optimization and Its Applications*. Springer, Cham, Switzerland, 3th edition, 2021.
- [16] M. Capuno, J.-S. Kim, and H. Song. Very short-term load forecasting using hybrid algebraic prediction and support vector regression. *Mathematical Problems in Engineering*, 2017:8298531, 2017.
- [17] 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.
- [18] P. Damci-Kurt, S. Küçükyavuz, D. Rajan, and A. Atamtürk. A polyhedral study of production ramping. *Mathematical Programming*, 158:175–205, 2016.
- [19] E. Danna, E. Rothberg, and C.L. Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Math Programming*, 102(1):71–90, 2005.

- 
- [20] L. Dannecker. *Energy Time Series Forecasting: Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain*. Springer, Dresden, Germany, 2015.
- [21] M. Darwiche, D. Conte, R. Raveaux, and V. T'Kindt. A local branching heuristic for solving a graph edit distance problem. *Computers & Operations Research*, 106:225–235, 2019.
- [22] G. Dudek. Pattern similarity-based methods for short-term load forecasting – Part 1: Principles. *Applied Soft Computing*, 37:277–287, 2015.
- [23] G. Dudek. Pattern-based local linear regression models for short-term load forecasting. *Electric Power Systems Research*, 130:139–147, 2016.
- [24] N. Dupin. Tighter MIP formulations for the discretised unit commitment problem with min-stop ramping constraints. *EURO Journal on Computational Optimization*, 5(1):149–176, 2017.
- [25] N. Dupin and E. Talbi. Matheuristics for the discrete unit commitment problem with min-stop ramping constraints. In *Proceedings of the Sixth International Workshop on Model-based Metaheuristics*, pages 72–81, Brussels, Belgium, May 2016. IRIDIA.
- [26] N. Dupin and E. Talbi. Parallel matheuristics for the discrete unit commitment problem with min-stop ramping constraints. *International Transactions in Operational Research*, 27(1):219–244, 2018.
- [27] B. Eldridge, R. O'Neill, and B. F. Hobbs. Near-optimal scheduling in day-ahead markets: Pricing models and payment redistribution bounds. *IEEE Transactions on Power Systems*, 35(3):1684–1694, 2020.
- [28] J. Y. Fan and J. D. McDonald. A real-time implementation of short-term load forecasting for distribution power systems. *IEEE Transactions on Power Systems*, 9(2):988–994, 1994.
- [29] M. Farrokhbadi. Day-ahead electricity demand forecasting: Post-COVID paradigm. IEEE Dataport, doi: 10.21227/67vy-bs34.s, November 2020.
- [30] D. Fayzur, A. Viana, and J. P. Pedroso. Metaheuristic search based methods for unit commitment. *International Journal of Electrical Power & Energy Systems*, 59:14–22, 2014.

- 
- [31] M. Fischetti and M. Fischetti. Matheuristics. In R. Martí, P. M. Pardalos, and M. G. C. Resende, editors, *Handbook of Heuristics*, chapter 5, pages 121–153. Springer, Cham, Switzerland, May 2018.
- [32] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1):23–47, 2003.
- [33] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104(1):91–104, 2019.
- [34] L. Garver. Power generation scheduling by integer programming-development of theory. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, 81(3):730–734, 1962.
- [35] F. Garza, M. M. Canseco, C. Challú, and K. G. Olivares. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, 2022. URL <https://github.com/Nixtla/statsforecast>. <https://github.com/Nixtla/statsforecast>.
- [36] C. Gentile, G. Morales-España, and A. Ramos. A tight MIP formulation of the unit commitment problem with start-up and shut-down constraints. *EURO Journal on Computational Optimization*, 5(1):177–201, 2017.
- [37] P. G. Gould, A. B. Koehler, J. K. Ord, R. D. Snyder, R. J. Hyndman, and F. Vahid-Araghi. Forecasting time series with multiple seasonal patterns. *European Journal of Operational Research*, 191(1):207–222, 2008.
- [38] G. Guastaroba, M. Savelsbergh, and M.G. Speranza. Adaptive kernel search: A heuristic for solving mixed integer linear programs. *European Journal of Operational Research*, 263(3):789–804, 2017.
- [39] L.S.M. Guedes, P.D. Maia, A.C. Lisboa, D.A. Gomes Vieira, and R. R. Saldanha. A unit commitment algorithm and a compact MILP model for short-term hydro-power generation scheduling. *IEEE Transactions on Power Systems*, 32(5):3381–3390, 2017.
- [40] J. Gunawan and Chin-Ya Huang. An extensible framework for short-term holiday load forecasting combining dynamic time warping and LSTM network. *IEEE Access*, 9:106885–106894, 2021.

- 
- [41] I. Harjunoski, M. Giuntoli, J. Poland, and S. Schmitt. Matheuristics for speeding up the solution of the unit commitment problem. In *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–5, Espoo, Finland, October 2021.
- [42] W. E Hart, Jean-Paul Watson, and D. L Woodruff. Pyomo: modeling and solving mathematical programs in python. *Mathematical Programming Computation*, 3(3): 219–260, 2011.
- [43] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, USA, 2nd edition, 2009.
- [44] B.F. Hobbs, M.H. Rothkopf, R.P. O’Neill, and H. Chao, editors. *The Next Generation of Electric Power Unit Commitment Models*. International Series in Operations Research and Management Science. Springer, Boston, USA, 2001.
- [45] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [46] R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22, 2008.
- [47] IBM. ILOG CPLEX Optimization Studio documentation. URL <https://www.ibm.com/docs/en/icos/20.1.0?topic=parameters-relative-mip-gap-tolerance>. Last visited: 19 June 2023.
- [48] R. B. Johnson, S. S. Oren, and A. J. Svoboda. Equity and efficiency of unit commitment in competitive electricity markets. *Utilities Policy*, 6(1):9–19, 1997.
- [49] C. Junk, L. D. Monache, and S. Alessandrini. Analog-based ensemble model output statistics. *Monthly Weather Review*, 143(7):2909–2917, 2015.
- [50] S. Kang and S. Cho. Approximating support vector machine with artificial neural network for fast prediction. *Expert Systems with Applications*, 41(10):4989–4995, 2014.
- [51] O. A. Karabiber and G. Xydis. Electricity price forecasting in the danish day-ahead market using the TBATS, ANN and ARIMA methods. *Energies*, 12(5), 2019.
- [52] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis. A genetic algorithm solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 11(1):83–92, 1996.

- [53] B. Knueven, J. Ostrowski, and J. P. Watson. Exploiting identical generators in unit commitment. *IEEE Transactions on Power Systems*, 33(4):4496–4507, 2018.
- [54] B. Knueven, J. Ostrowski, and J. P. Watson. On mixed-integer programming formulations for the unit commitment problem. *INFORMS Journal on Computing*, 32(4):857–876, 2020.
- [55] L. Lannelongue, J. Grealey, and M. Inouye. Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12):2100707, 2021.
- [56] A. M. De Livera, R. J. Hyndman, and R. D. Snyder. Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527, 2011.
- [57] X. Ma, H. Song, M. Hong, J. Wan, Y. Chen, and E. Zak. The security-constrained commitment and dispatch for Midwest ISO day-ahead co-optimized energy and ancillary service market. In *2009 IEEE Power Energy Society General Meeting*, pages 1–8, Calgary, Canada, July 2009.
- [58] V. Maniezzo, Marco A. Boschetti, and T. Stützle. *Matheuristics: Algorithms and Implementations*. EURO Advanced Tutorials on Operational Research. Springer, Cham, Switzerland, 2021.
- [59] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera. A methodology for applying  $k$ -nearest neighbor to time series forecasting. *Artificial Intelligence Review*, 52(3):2019–2037, 2019.
- [60] Mexican Secretary of Government. Acuerdo por el que se emite el manual de pronósticos. Diario Oficial de la Federación, 24 November 2017. URL [https://dof.gob.mx/nota\\_detalle.php?codigo=5505475&fecha=23/11/2017](https://dof.gob.mx/nota_detalle.php?codigo=5505475&fecha=23/11/2017). In Spanish.
- [61] L. D. Monache, F. A. Eckel, D. L. Rife, B. Nagarajan, and K. Searight. Probabilistic weather prediction with an analog ensemble. *Applied Energy*, 141:3498–3516, 2013.
- [62] L. Montero, A. Bello, and J. Reneses. A review on the unit commitment problem: Approaches, techniques, and resolution methods. *Energies*, 15(4):1296, 2022.

- 
- [63] G. Morales-España. *Unit Commitment: Computational Performance, System Representation and Wind Uncertainty Management*. PhD thesis, Universidad Pontificia Comillas, Madrid, Spain, October 2014.
- [64] G. Morales-España, J. M. Latorre, and A. Ramos. Tight and compact MILP formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 28(4):4897–4908, 2013.
- [65] G. Morales-España, J. M. Latorre, and A. Ramos. Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment. *IEEE Transactions on Power Systems*, 28(2):1288–1296, 2013.
- [66] G. Morales-España, A. Ramos, and J. García-González. An MIP formulation for joint market-clearing of energy and reserves based on ramp scheduling. *IEEE Transactions on Power Systems*, 29(1):476–488, 2014.
- [67] G. Morales-España, A. Ramos, and J. García-González. Tight and compact MIP formulation of configuration-based combined-cycle units. *IEEE Transactions on Power Systems*, 31(2):1350–1359, 2016.
- [68] H. Mori and T. Usami. Unit commitment using tabu search with restricted neighborhood. In *Proceedings of International Conference on Intelligent System Application to Power Systems*, pages 422–427, Orlando, USA, 1996.
- [69] H. Nano, S. New, A. Cohen, and B. Ramachandran. Load forecasting using multiple linear regression with different calendars. In R. K. Chauhan and K. Chauhan, editors, *Distributed Energy Resources in Microgrids*, chapter 16, pages 405–417. Academic Press, Greater Noida, India, 2019.
- [70] V. Ngo, W. Wu, B. Zhang, Z. Li, and Y. Wang. Ultra-short-term load forecasting using robust exponentially weighted method in distribution networks. In *2015 IEEE Power & Energy Society General Meeting*, pages 1–5, Denver, USA, July 2015.
- [71] E. Nycander, G. Morales-España, and L. Söder. Security constrained unit commitment with continuous time-varying reserves. *Electric Power Systems Research*, 199:107276, 2021.

- [72] J. Ostrowski, M. F. Anjos, and A. Vannelli. Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Transactions on Power Systems*, 27(1):39–46, 2012.
- [73] K. Pan and Y. Guan. A polyhedral study of the integrated minimum-up/-down time and ramping polytope. Preprint arXiv:1604.02184, 2016.
- [74] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [75] D. Rajan and S. Takriti. Minimum up/down polytopes of the unit commitment problem with start-up costs. Research report RC23628 (W0506-050), IBM, Yorktown Heights, USA, June 2005.
- [76] A. M. N. C. Ribeiro, P. R. X. do Carmo, P. T. Endo, P. Rosati, and T. Lynn. Short- and very short-term firm-level load forecasting for warehouses: A comparison of machine learning and deep learning models. *Energies*, 15(3):750, 2022.
- [77] C. Sabóia and A. Diniz. A local branching approach for network-constrained thermal unit commitment problem under uncertainty. In *Power Systems Computation Conference*, pages 1–7, Genoa, Italy, June 2016. IEEE.
- [78] H. Sadaat. *Power System Analysis*. McGraw-Hill, New York, USA, 1999.
- [79] T. N. Santos, A. L. Diniz, C. H. Saboia, R. N. Cabral, and L. F. Cerqueira. Hourly pricing and day-ahead dispatch setting in Brazil: The DESSEM model. *Electric Power Systems Research*, 189:106709, 2020.
- [80] B. Saravanan, D. Siddharth, S. Surbhi, and K. Dwarkadas. A solution to the unit commitment problem—a review. *Frontiers in Energy*, 7(2):223–236, 2013.
- [81] D. W. Scott. Sturges’ rule. *WIREs Computational Statistics*, 1(3):303–306, 2009.
- [82] S. Seabold and J. Perktold. Statsmodels: Econometric and statistical modeling with Python. In S. van der Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference (SciPy 2010)*, pages 92–96, Austin, USA, 2010.

- [83] R. Sioshansi, R. O'Neill, and S. S. Oren. Economic consequences of alternative solution methods for centralized unit commitment in day-ahead electricity markets. *IEEE Transactions on Power Systems*, 23(2):344–352, 2008.
- [84] R. Todosijević, M. Mladenović, S. Hanafi, N. Mladenović, and I. Crévits. Adaptive general variable neighborhood search heuristics for solving the unit commitment problem. *International Journal of Electrical Power & Energy Systems*, 78:873–883, 2016.
- [85] C. Tong, L. Zhang, H. Li, and Y. Ding. Temporal inception convolutional network based on multi-head attention for ultra-short-term load forecasting. *IET Generation, Transmission & Distribution*, 16(8):1680–1696, 2022.
- [86] A. Viana and J. P. Pedroso. A new MILP-based approach for unit commitment in power production planning. *International Journal of Electrical Power and Energy Systems*, 44(1):997–1005, 2013.
- [87] L. A. Wolsey. *Integer Programming*. Wiley, New York, USA, 2nd edition, 2021.
- [88] L. Wu. Accelerating NCUC via binary variable-based locally ideal formulation and dynamic global cuts. *IEEE Transactions on Power Systems*, 31(5):4097–4107, 2016.
- [89] L. Yang, C. Zhang, J. Jian, K. Meng, Y. Xu, and Z. Dong. A novel projected two-binary-variable formulation for unit commitment in power systems. *Applied Energy*, 187:732–745, 2017.
- [90] L. Yang, W. Li, Y. Xu, C. Zhang, and S. Chen. Two novel locally ideal three-period unit commitment formulations in power systems. *Applied Energy*, 284:116081, 2021.
- [91] J. Zhu. *Optimization of Power System Operation for Large Systems*. Wiley, Hoboken, USA, 2009.
- [92] M. A. Zuniga-Garcia, G. Santamaría-Bonfil, G. Arroyo-Figueroa, and R. Batres. Prediction interval adjustment for load-forecasting using machine learning. *Applied Sciences*, 9(24):5269, 2019.

# AUTOBIOGRAPHY

---

Uriel Iram Lezama Lope



Candidate to obtain the degree of  
Doctorado en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León  
Facultad de Ingeniería Mecánica y Eléctrica

Thesis:

EFFICIENT METHODS FOR SOLVING POWER SYSTEM OPERATION SCHEDULING  
CHALLENGES: THE THERMAL UNIT COMMITMENT PROBLEM WITH STAIRCASE  
COST AND THE VERY SHORT-TERM LOAD FORECASTING PROBLEM

Uriel was born on September 9, 1980, in Monterrey, Mexico. He earned a Bachelor of Science in Electrical Engineering with a specialization in Electrical Power Systems from the Instituto Politécnico Nacional (IPN), Mexico 2003. In 2010, he obtained a Master of Science in Systems Engineering from the same institution. In 2009, Uriel joined the Instituto Nacional de Electricidad y Energías Limpias (INEEL), where he worked as a researcher and project manager, developing advanced software to schedule the operation of power systems. He has contributed to developing tools to manage the short-term energy market system in the wholesale electricity market in Mexico. Uriel is also a member of the Institute of Electrical and Electronics Engineers (IEEE) and its chapter of the Power Energy Society (PES). His areas of expertise include the operation scheduling of power systems, operations research for power systems, load demand and energy forecasting, and software development.