Enhancing Matheuristics for a Thermal Unit Commitment Problem through Kernel Search and Local Branching

Preprint DIE-RR25-01
Graduate Program in Electrical Engineering
Department of Mechanical and Electrical Engineering
Universidad Autónoma de Nuevo León
San Nicolás de los Garza, NL, Mexico, June 2025

Uriel I. Lezama-Lope
Independent Researcher
Cuernavaca, Mexico
E-mail: *urieliram@yahoo.com*

Roger Z. Ríos-Mercado
Graduate Program in Electrical Engineering
Universidad Autónoma de Nuevo León (UANL)
San Nicolás de los Garza, NL, Mexico
E-mail: *roger@optimizer.mx*

Diana L. Huerta-Muñoz
Graduate Program in Systems Engineering
Universidad Autónoma de Nuevo León (UANL)
San Nicolás de los Garza, NL, Mexico
E-mail: *dianahuerta@yalma.fime.uanl.mx*

June 2025

Abstract

The unit commitment problem is an important problem arising in the planning and optimization of power system operations. The problem is characterized by the need to determine the optimal scheduling of electrical generators to meet fluctuating demand while minimizing production costs. In this study, we address a thermal unit commitment problem under a staircase cost structure. Despite the encouraging outcomes from the reformulation strategies to the mathematical model, this problem still presents significant challenges. Despite the encouraging outcomes from the reformulation strategies to the mathematical model, the UCP still presents significant challenges. We propose a novel two-phase matheuristic framework that first employs a constructive heuristic to generate high-quality initial solutions, followed by an enhancement phase that takes advantage of mechanisms such as local branching and kernel search methods. Experimental results on real-world instances demonstrate the effectiveness and value of each of the matheuristic components and that the proposed approach consistently outperforms standard solvers and existing heuristics, particularly for large-scale instances.

Keywords: Thermal unit commitment problem; Integer programming; Matheuristics; Kernel search; Local branching.

1 Introduction

This research aims to improve the efficiency of solving a thermal unit commitment problem (UCP), a key challenge in power system planning. The UCP involves making decisions about generator operations, such as switching units on or off and determining optimal power output levels, to meet consumer demand while satisfying operational, technical, and economic constraints. As a daily optimization task, it seeks to minimize production costs while adhering to limitations imposed by the electrical system, generator characteristics, and market requirements. Commonly, the UCP is addressed through the formulation of an adequate mixed-integer linear programming (MILP) model.

Despite recent advances on models and algorithms, solving UCPs within limited time frames remains a challenge. To tackle this problem, the purpose of this study is to provide a hybrid heuristic framework that attempts to exploit the advantages of components such as kernel search and local branching with the aim of achieving optimal or near-optimal solutions in the time allowed.

The proposed solution algorithm consists of a two-phase approach, comprising a constructive phase and an improving phase. In the first phase, we develop a relax-and-fit heuristic (named HARDUC) that builds feasible solutions to the problem. Then, in the second phase, we derive a method that aims to improve the solution by adapting local branching [14] and kernel search [2] mechanisms.

The general algorithm and its individual components were fully assessed in three groups of instances derived from Kazarlis et al. [21]. The empirical evaluation included a comparison with the constructive HGPS heuristic of Harjunkoski et al. [19] and the CPLEX solution of the thermal UCP MILP model proposed by Knueven et al. [22].

It was observed that although the branch-and-bound solver demonstrates strong performance for small-scale problem instances, our proposed method consistently achieves better results for medium- and large-scale instances within the allocated time frame. In particular, our methods reliably generate solutions across all cases, whereas the solver frequently fails to find feasible solutions in medium-, and large-scale instances.

The paper is structured as follows. The most relevant related work on algorithms, including mathematics, for the UCP is surveyed in Section 2. Section 3 presents the problem, its assumptions, and the mathematical formulation. This is followed by Section 4 that includes a full and detailed description of the proposed solution framework and its individual components. Section 5 presents an empirical evaluation of the algorithm and a comparison with existing work. Closing remarks, conclusions, and discussion of future research directions are given in Section 6.

2 Literature Survey

UCP models are classified into deterministic and stochastic. In our work, we deal with a deterministic model, therefore we focus our literature review on deterministic models and approaches. For stochastic UCPs, the reader is referred to the work by Håberg [18], who reviews models and algorithms developed for handling uncertainty in UCPs.

The literature on UCP models is vast [24]. 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 [4] outline some examples of these models.

Recent research has established the UCP as a computationally complex challenge. Anjos [3] affirm that the UCP is an NP-hard, large-scale, non-linear, non-convex, combinatorial optimization problem. On the other hand, Bendotti et al. [5] 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.

2.1 Traditional approaches

In his doctoral thesis, Morales-España [25] 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. [26] 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 [27, 29, 41, 42]. These studies have particularly focused on the thermal UCP model. In addition, Guedes et al. [17] 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. [22] 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. [31] 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 branch-and-bound (B&B) method and its variants [40]. 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 [21], Tabu Search [30], and Variable Neighborhoods Search [38] 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 sub-section comprehensively surveys the most recent matheuristic approaches for solving UCP models.

2.2 Hybrid heuristic approaches

A strategy for reducing solution times and improving accuracy in the UCP is the hybrid heuristic methods, known as matheuristics. These strategies cleverly embed exact methods into metaheuristic approaches [6, 23]. They also take advantage of advances in good MILP formulations [13].

The efficiency of branch-and-bound algorithms has been significantly improved by the implementation of heuristic ideas in MILP solvers [40]. Matheuristics can improve solution quality, convergence guarantees, and computational efficiency, leading to better solutions. Constructive heuristics, such as relax-and-fix (R&F) methods [40], the feasibility pump (FP) [14], and the relaxation-induced neighborhood search (RINS) [7], use continuous solutions from linear relaxation and rounding to guide the search for feasible integer solutions. This approach improves the quality and efficiency of the solutions by exploring the neighborhoods around relaxed solutions in combinatorial optimization problems.

Other matheuristics inspired by local search algorithms are then used to refine solution quality, such as local branching (LB) strategies introduced by Fischetti and Lodi [14], which define the neighborhoods within the MILP with maximum modifications to the incumbent solution. These strategies iteratively explore neighboring solutions by applying small changes, focusing on improving the incumbent while maintaining feasibility. The approach allows for an effective exploration of the solution space, providing higher-quality solutions by guiding the search toward more optimal regions.

The Kernel Search (KS) matheuristic, introduced by Angelelli et al. [2], is inspired by greedy algorithms. The method begins by partitioning the decision variables of an MILP formulation into two sets: a central kernel and multiple buckets. A simplified version of the problem, or sub-MILP, is then solved for each bucket, using the kernel as a starting point. Iteratively, the method adds new variables from the buckets to the kernel, progressively improving the solution. The process continues until all the variables from the buckets have been integrated into the kernel, resulting in an optimal or near-optimal solution with the variables assigned within the kernel.

Following this, a survey of the matheuristic and heuristic methods applied to variations of UCP models is presented. It is worth noting that there is a significant lack of research on the use of matheuristics in the UCP. In Table 1, the comparison between the works that have addressed the UCP problem with these heuristic approaches is presented. The table lists the main constraints and features of the UCP models addressed in each work and their corresponding solution methods.

Table 1: Survey of works on UCPs using matheuristics.

Article	Method	UCP type	PC	SUC	SDC	RR	GL	RUD	SR	MUDT
Fayzur et al. [12] 2014	LB-ILA, PSO-ILA	thermal	Q	Cold and hot start-up	✓	✓	✓	✓		✓
Todosijević et al. [38] 2016	VNS-linear programming	thermal	Q	Cold and hot start-up	✓	✓	✓			✓
Sabóia and Diniz [34] 2016	LB	stochastic thermal compact formulation	L	Exponential start-up cost	✓		✓	✓		✓
Dupin and Talbi [10] 2016	R&F, VNS with MIP neighborhoods. LB, RINS.	discrete thermal, real-time	L	Fixed cost		✓	✓			✓
Dupin and Talbi [11] 2018	R&F-B&B	discrete thermal, real-time	L	Fixed cost		✓	✓	✓		✓
Santos et al. [35] 2020	FP, LB	Hydrothermal	L	Fixed cost	✓		✓	✓	✓	✓
Harjunkoski et al. [19] [17] 2021	R&F	thermal	L	Fixed cost	✓	✓	✓	✓		✓
This work	R&F, LB, KS.	thermal T&C model	S	Variable start-up cost	✓	✓	✓	✓	✓	✓

PC: Production cost; SUC: Start-up cost; SDC: Shut-down cost; RR: Reserve requirements; GL: Gen limits; RUD: Ramp up/down; SR: Start-up/shut-down ramps; MUDT: min up/down times.
Q: quadratic; L: linear; S: staircase.

First, Fayzur et al. [12] present two matheuristic approaches to solve a thermal UCP. The first approach combines the original version of LB from Fischetti and Lodi [14] 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 include power balance, spinning reserve requirements, minimum up and down times, production limits, and ramps [39]. The study solves previously unsolvable instances and achieves faster optimal results for larger instances. The proposed LB-ILA algorithms significantly reduce the average CPU time for larger instances by 40%-56%. However, these algorithms perform worse for smaller instances due to unnecessary exploration of the neighborhoods.

Sabóia and Diniz [34] 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, continuously 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 on one instance.

Todosijević et al. [38] 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 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. However, the model does not incorporate ramps, which are essential constraints in real-life UCP usage. Furthermore, a particular feature of their UCP is that the production cost of the generators is quadratic.

Dupin and Talbi [10] 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 within 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 mathematical model used by Dupin [9] comprised of constraints such as demand, reserves, minimum up and down times [33], 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 results of the metaheuristic method are better than the MILP solution, which was obtained by the solver in the allotted time frame.

In follow-up work, Dupin and Talbi [11] improved the discrete thermal UCP formulation with tight and compact (T&C) features and extended the model to include min-stop ramping for a discrete thermal UCP. To address the problem, the authors propose several constructive matheuristics designed to generate good-quality feasible solutions. These solutions can be effective initial solutions for a branch-and-bound (B&B) algorithm. Among all the ingenious constructive methods presented, they implemented one with the KS ideas proposed by Guastaroba et al. [16], 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 implementation of the method. The proposed method produces precise, high-quality solutions, 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.

According to the investigation undertaken by the Santos et al. [35], who present the model and method used to plan 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. [15] adapted for binary variable decisions. They also created an objective function adapted with a mathematical term that calculates the Hamming distance. The researchers 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, an exact method would not yield a timely resolution within two hours. The authors report using the same version of LB as that of Sabóia and Diniz [34].

In the study conducted by Harjunkoski et al. [19], an R&F heuristic for the UCP is devised. The proposed method starts with solving a linear relaxation of the problem. Subsequently, they analyzed the product of the commitment binary variable with each generator's power level. The commitment binary variables were fixed and solved as an MILP problem if the product exceeded the minimum feasible generation level. This constructive method provides a warm start to the B&B algorithm and accelerates the UCP solution process.

Positioning our work in the literature

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 Sabóia and Diniz [34], Dupin and Talbi [10, 11], Harjunkoski et al. [19] and quadratic costs Fayzur et al. [12], Todosijević et al. [38]. Another notable difference is that the modeling of generator startup costs is simplified in most of the papers to a fixed startup cost Dupin and Talbi [10, 11], Santos et al. [35], Harjunkoski et al. [19] or hot and cold starts Fayzur et al. [12], Todosijević et al. [38], or, at best, exponential startup costs depending on the time the generator has been off Sabóia and Diniz [34]. In our case, we use variable startup cost constraints Morales-España et al. [26] 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 [10, 11] tackle a discrete thermal UCP for a real-time horizon with limited forward scheduling periods. [35] deal with a hydrothermal UCP problem, while Sabóia and Diniz [34] solve a stochastic UCP thermal problem. Although Fayzur et al. [12] present a hybrid LB method with cost linearization, the version they report is the original one developed by Fischetti and Lodi [14]. Alternatively, Todosijević et al. [38] apply VNS to solve the problem and incorporates linear programming to address a subproblem. Similarly, Dupin and Talbi [10] 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 [34] modify the original LB to include the power network. In their research, Santos et al. [35] report using a variant of FP as a constructive strategy and LB as an improvement strategy. Furthermore, the authors state that they use the same version of LB as Sabóia and Diniz [34].

The first paper to report on the use of the KS for UCP is Dupin and Talbi [11], 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 a complete implementation of the KS method.

Furthermore, Harjunkoski et al. [19] 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.

Summary of research contributions

In terms of solution methodologies, we proposed a novel constructive method that aims to compete with the approach of Harjunkoski et al. [19] and the solver. The solution calculated by the constructive method provides the first solution to the improving method.

In the improvement phase, we have developed a unique version of KS; our version introduces several significant innovations compared to the original method proposed by Angelelli et al. [2]. First, our KS eliminates the need for kernel construction, as a preliminary constructive stage already provides the kernel. Second, we employ the statistical rule of Sturges to determine the number of buckets in KS, a refinement that optimizes the method's performance. Third, our focus is on the dominant variables of the problem, which enhances computational efficiency. Fifth, our method calculates the reduced costs by fixing kernel variables while leaving the remaining variables free, using the linear relaxation of the problem. Finally, our kernel expansion process continues even after the buckets have been processed. If time permits, the kernel is reconfigured using the latest solution, and the expansion process is restarted, allowing for further refinement of the results.

Furthermore, we introduce four specialized versions of the Local Branching (LB) method, each incorporating refinements to enhance performance. Our adaptations present significant differences compared to the original approach by Fischetti and Lodi [14]. The first procedure, LB1, defines a restricted candidate list (RCL) based on variables that are nonzero in the constructive method (HARDUC) but fall below the minimum generator power threshold. Additionally, LB1 employs a soft-fixing constraint that retains at least 90% of the binary support variables in the solution. The second procedure, LB2, removes the soft-fixing constraint while maintaining the RCL based on the HARDUC rule. The third procedure, LB3 adheres closely to the original LB method but does not use an RCL or soft-fixing constraints, limiting its local search to binary support variables and their complements. Finally, LB4 selects the RCL based on negative reduced costs, improving the efficiency of the search process. A key feature of our LB adaptations is that they operate solely on the dominant variable $u_{g,t}$ to define the binary support, unlike the generic LB approach, which considers all binary variables.

3 Description of the Thermal Unit Commitment Problem

3.1 Problem definition and assumptions

The Unit Commitment Problem (UCP) aims to minimize the total operational cost of a set of generators $g \in \mathcal{G}$ over a planning horizon $t \in \mathcal{T}$. The objective function considers several cost components: production costs c^{P}_{gt}, fixed costs C^{R}_{g}, start-up costs c^{SU}_{gt}, and shutdown costs c^{SD}_{gt}. Achieving this minimization requires adherence to constraints that reflect the physical and operational limits of the system, ensuring feasible and optimized generator scheduling. These constraints govern aspects such as power generation bounds, minimum uptime and downtime requirements, ramping limits, demand-reserve satisfaction, and staircase cost modeling, and are represented using binary and continuous variables for generator states and power outputs.

The constraints of the UCP encapsulate the physical and operational characteristics of power generation systems, ensuring solutions are both feasible and realistic. Each constraint represents a specific aspect of generator behavior or grid requirements:

Power limits: Each generator must operate within its designed capacity, defined by the bounds $\underline{P}_g u_{gt} \leq p_{gt} \leq \overline{P}_g u_{gt}$ for all $g \in \mathcal{G}$ and $t \in \mathcal{T}$. This prevents overloading and maintains efficient operation.

Minimum uptime and downtime: Generators must stay online for at least UT_g periods after starting up and offline for at least DT_g periods after shutting down. These constraints, $\sum_{\tau=t}^{t+UT_g-1} u_{g\tau} \geq UT_g v_{gt}$ and $\sum_{\tau=t}^{t+DT_g-1}(1-u_{g\tau}) \geq DT_g w_{gt}$, prevent excessive wear and ensure reliable operation.

Ramping constraints: The rate at which power output changes is limited by ramp-up and ramp-down constraints, defined as $p_{gt} - p_{g(t-1)} \leq RU_g$ and $p_{g(t-1)} - p_{gt} \leq RD_g$. These capture the physical limitations of generator components.

Start-up and shutdown costs: These costs, c^{SU}_{gt} and c^{SD}_{gt}, depend on generator downtime and reflect additional fuel use and equipment wear during transitions.

Demand and reserve requirements: Total generation must satisfy demand D_t and maintain a reserve margin R_t, expressed as $\sum_{g\in\mathcal{G}} p_{gt} \geq D_t + R_t$ for all $t \in \mathcal{T}$. This ensures grid stability and reliability.

Staircase cost structure: Production costs vary staircase with output, modeled as $p_{gt} = \sum_{l\in\mathcal{L}_g} p^l_{gt}$, where p^l_{gt} represents the power produced in each segment l. This mirrors fuel efficiency dynamics and captures operational states.

Initial conditions: The initial states and outputs of generators are predefined, with $u_{g0}, p_{g0}, v_{g0}, w_{g0}$ specified to ensure a seamless transition from previous operations.

System constraints: These maintain the balance between generation, demand, and reserves, ensuring grid-wide operational reliability.

3.2 Mixed-integer linear programming model

This section introduces a comprehensive Thermal Unit Commitment formulation that is a robust benchmark for evaluating subsequent matheuristic methods. Our model incorporates diverse constraints, including power limits, minimum uptime/downtime, ramp capabilities, variable start-up costs, and demand-reserve requirements, while also narrowing down solution space with staircase cost production and valid inequalities.

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; ($t \in \mathcal{T}$)

$\mathcal{T}^0$ $= \mathcal{T} \setminus \{|\mathcal{T}|\}$ Set of time periods in the planning horizon except the last one

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

$C^{\mathrm{S}}_{g,s}$	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 $[\underline{T}_{g,s}, \overline{T}_{g,s})$; in \$/h
C^{SD}_{g}	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
$\overline{P}_g$,$\underline{P}_g$	Maximum and minimum generation value of generator ($g \in \mathcal{G}$); in MW
$\overline{P}^l_g$	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
RU_g	Ramp-up rate of generator ($g \in \mathcal{G}$) to increase power between two consecutive periods; in MW/h
$\lvert\mathcal{S}_g\rvert$	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^{RU}_g	Time that a generator ($g \in \mathcal{G}$) spends ramping to go from SU_g to $\overline{P}_{gt}$; in h
T^{RD}_g	Time that a generator ($g \in \mathcal{G}$) spends ramping to go from $\overline{P}_{gt}$ 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_{gt}	Equal to 1 if generator ($g \in \mathcal{G}$) is online in period ($t \in \mathcal{T}$), and 0 otherwise
v_{gt}	Equal to 1 if generator ($g \in \mathcal{G}$) starts up at the beginning of period ($t \in \mathcal{T}$), and 0 otherwise
w_{gt}	Equal to 1 if generator ($g \in \mathcal{G}$) is shut-down at the beginning of period ($t \in \mathcal{T}$), and 0 otherwise

δ_{gts} 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^{P}_{gt} Production cost over $\underline{P}$ of generator $(g \in \mathcal{G})$ in period $(t \in \mathcal{T})$; in \$

c^{SD}_{gt} Shut-down cost of generator $(g \in \mathcal{G})$ in period $(t \in \mathcal{T})$; in \$

c^{SU}_{gt} Start-up cost of generator $(g \in \mathcal{G})$ in period $(t \in \mathcal{T})$; in \$

p_{gt} Amount of power a generator $(g \in \mathcal{G})$ produces in period $(t \in \mathcal{T})$, in MW

p'_{gt} Amount of power above minimum $\underline{P}_g$ that a generator $(g \in \mathcal{G})$ produces in period $(t \in \mathcal{T})$, in MW

$\overline{p}_{gt}$ Maximum power available from generator $(g \in \mathcal{G})$ produces in period $(t \in \mathcal{T})$, in MW

$\overline{p}'_{gt}$ Maximum power available above minimum from generator $(g \in \mathcal{G})$ produces in period $(t \in \mathcal{T})$, in MW

p^l_{gt} 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_{gt} Spinning reserves provided by a generator $(g \in \mathcal{G})$ in period $(t \in \mathcal{T})$, in MW

The parameters C^v_{gl} and C^w_{gl} are calculated as follows [22].

$$C^v_{gl} = \begin{cases} 0 & \text{if } \overline{P}^l_g \leq SU_g \\ \overline{P}^l_g - SU_g & \text{if } \overline{P}^{l-1}_g < SU_g < \overline{P}^l_g \\ \overline{P}^l_g - \overline{P}^{l-1}_g & \text{if } \overline{P}^{l-1}_g \geq SU_g \end{cases}$$

$$C^w_{gl} = \begin{cases} 0 & \text{if } \overline{P}^l_g \leq SD_g \\ \overline{P}^l_g - SD_g & \text{if } \overline{P}^{l-1}_g < SD_g < \overline{P}^l_g \\ \overline{P}^l_g - \overline{P}^{l-1}_g & \text{if } \overline{P}^{l-1}_g \geq SD_g \end{cases}$$

Finally, T^{RU}_g and T^{RD}_g are also computed as follows [22]:

$$T^{\mathrm{RU}}_g = \left\lfloor \frac{\overline{P}_g - SU_g}{RU_g} \right\rfloor,$$

$$T^{\mathrm{RD}}_g = \left\lfloor \frac{\overline{P}_g - SD_g}{RD_g} \right\rfloor.$$

Mathematical formulation

The MILP model is given by:

$$\text{(TUCP) Minimize} \quad \sum_{t\in\mathcal{T}}\sum_{g\in\mathcal{G}} C_g^{\mathrm{R}} u_{gt} + c_{gt}^{\mathrm{p}} + c_{gt}^{\mathrm{SU}} + c_{gt}^{\mathrm{SD}} \tag{1}$$

Subject to:

$$\sum_{i=t-UT_g+1}^{t} v_{gi} \leq u_{gt} \qquad g\in\mathcal{G},\, t\in\{UT_g,\ldots,|\mathcal{T}|\}, \tag{2}$$

$$\sum_{i=t-DT_g+1}^{t} w_{gi} \leq 1-u_{gt} \qquad g\in\mathcal{G},\, t\in\{DT_g,...,|\mathcal{T}|\}, \tag{3}$$

$$\sum_{i=1}^{\min\{U_g,|\mathcal{T}|\}} u_{gi} = \min\{U_g,|\mathcal{T}|\} \qquad g\in\mathcal{G} \tag{4}$$

$$\sum_{i=1}^{\min\{D_g,|\mathcal{T}|\}} u_{gi} = 0 \qquad g\in\mathcal{G} \tag{5}$$

$$u_{gt} - u_{g,t-1} = v_{gt} - w_{gt} \qquad g\in\mathcal{G},\, t\in\mathcal{T} \tag{6}$$

$$p'_{gt} + r_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right)u_{gt} - \left(\overline{P}_g - SU_g\right)v_{gt} - \left[SU_g - SD_g\right]^+ w_{g,t+1} \qquad g\in\mathcal{G}^1,\, t\in\mathcal{T}^0 \tag{7}$$

$$p'_{gt} + r_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right)u_{gt} - \left(\overline{P}_g - SD_g\right)w_{g,t+1} - \left[SD_g - SU_g\right]^+ v_{gt} \qquad g\in\mathcal{G}^1,\, t\in\mathcal{T}^0 \tag{8}$$

$$p_{gt} \leq \overline{P}_g u_{gt} - \sum_{i=0}^{T_g^{\mathrm{RU}}}\left(\overline{P}_g - (SU_g + iRU_g)\right)v_{g,t-i} - \sum_{i=0}^{T_g^{\mathrm{RD}}}\left(\overline{P}_g - (SD_g + iRD_g)\right)w_{g,t+1+i} \qquad g\in\mathcal{G}, t\in\{T_g^{\mathrm{RU}}, |\mathcal{T}| - T_g^{\mathrm{RD}}\} \tag{9}$$

$$p'_{gt} + r_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right)u_{gt} - \left(\overline{P}_g - SU_g\right)v_{gt} - \left(\overline{P}_g - SD_g\right)w_{g,t+1} \qquad g\in\mathcal{G}^{>1},\, t\in\mathcal{T}^0 \tag{10}$$

$$p'_{gt} + r_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right)u_{gt} - \left(\overline{P}_g - SU_g\right)v_{gt} \qquad g\in\mathcal{G}^1, t\in\mathcal{T} \tag{11}$$

$$p'_{gt} + r_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right)u_{gt} - \left(\overline{P}_g - SD_g\right)w_{g,t+1} \qquad g\in\mathcal{G}^1, t\in\mathcal{T}^0 \tag{12}$$

$$\overline{p}'_{gt} - p'_{g,t-1} \leq (SU_g - \underline{P}_g - RU_g)v_{gt} + RU_g u_{gt} \qquad g\in\mathcal{G}, t\in\mathcal{T} \tag{13}$$

$$p'_{g,t-1} - p'_{gt} \leq (SD_g - \underline{P}_g - RD_g)w_{gt} + RD_g u_{g,t-1} \qquad g\in\mathcal{G}, t\in\mathcal{T} \tag{14}$$

$$p_{gt}^l \leq \left(\overline{P}_g^l - \overline{P}_g^{l-1}\right)u_{gt} \qquad g\in\mathcal{G}, t\in\mathcal{T}, l\in\mathcal{L}_g \tag{15}$$

$$\sum_{l\in\mathcal{L}_g} p_{gt}^l = p'_{gt} \qquad g\in\mathcal{G}, t\in\mathcal{T} \tag{16}$$

$$\sum_{l\in\mathcal{L}_g} C_g^l p_{gt}^l = c_{gt}^{\mathrm{p}} \qquad g\in\mathcal{G}, t\in\mathcal{T} \tag{17}$$

$$p'_{gt} \leq \left(\overline{P}_g - \underline{P}_g\right) u_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{18}$$

$$p^l_{gt} \leq \left(\overline{P}^l_g - \overline{P}^{l-1}_g\right) u_{gt} - C^v_{gl} v_{gt} - C^w_{gl} w_{g,t+1} \qquad g \in \mathcal{G}^{>1},\ t \in \mathcal{T}^0,\ l \in \mathcal{L}_g \tag{19}$$

$$p^l_{gt} \leq \left(\overline{P}^l_g - \overline{P}^{l-1}_g\right) u_{gt} - C^v_{gl} v_{gt} \qquad g \in \mathcal{G}^1, t \in \mathcal{T}, l \in \mathcal{L}_g \tag{20}$$

$$p^l_{gt} \leq \left(\overline{P}^l_g - \overline{P}^{l-1}_g\right) u_{gt} - C^w_{gl} w_{g,t+1} \qquad g \in \mathcal{G}^1,\ t \in \mathcal{T}^0,\ l \in \mathcal{L}_g \tag{21}$$

$$p^l_{gt} \leq \left(\overline{P}^l_g - \overline{P}^{l-1}_g\right) u_{gt} - C^v_{gl} v_{gt} - \left[C^v_{gl} - C^w_{gl}\right]^+ w_{g,t+1} \qquad g \in \mathcal{G}^{1*},\ t \in \mathcal{T}^0,\ l \in \mathcal{L}_g \tag{22}$$

$$p^l_{gt} \leq \left(\overline{P}^l_g - \overline{P}^{l-1}_g\right) u_{gt} - C^w_{gl} w_{g,t+1} - \left[C^w_{gl} - C^v_{gl}\right]^+ v_{gt} \qquad g \in \mathcal{G}^{1*},\ t \in \mathcal{T}^0,\ l \in \mathcal{L}_g \tag{23}$$

$$\delta_{gts} \leq \sum_{i=\underline{T}_{g,s}}^{\underline{T}_{g,s+1}-1} w_{g,t-i} \qquad g \in \mathcal{G}, t \in \mathcal{T}, s \in [1, |\mathcal{S}_g|) \tag{24}$$

$$v_{gt} = \sum_{s=1}^{|\mathcal{S}_g|} \delta_{gts} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{25}$$

$$c^{\text{SU}}_{gt} = \sum_{s=1}^{|\mathcal{S}_g|} C^{\text{S}}_g \delta_{gts} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{26}$$

$$\delta_{gts} = 0 \qquad g \in \mathcal{G}, s \in [1, |\mathcal{S}_g|),\ t \in \left(\underline{T}_{g,s+1} - DT^0_g, \underline{T}_{g,s+1}\right) \tag{27}$$

$$c^{\text{SD}}_{gt} = C^{\text{SD}}_g w_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{28}$$

$$\sum_{g \in \mathcal{G}} p_{gt} = De_t \qquad t \in \mathcal{T} \tag{29}$$

$$\sum_{g \in \mathcal{G}} \overline{p}_{gt} \geq De_t + R_t \qquad t \in \mathcal{T} \tag{30}$$

$$\sum_{g \in \mathcal{G}} r_{gt} \geq R_t \qquad t \in \mathcal{T} \tag{31}$$

$$p_{gt} = p'_{gt} + \underline{P}_g u_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{32}$$

$$\overline{p}_{gt} = \overline{p}'_{gt} + \underline{P}_g u_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{33}$$

$$\overline{p}_{gt} = p'_{gt} + r_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{34}$$

$$\overline{p}_{gt} = p_{gt} + r_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{35}$$

$$p_{gt} \leq \overline{p}_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{36}$$

$$p'_{gt} \leq \overline{p}'_{gt} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{37}$$

$$u_{gt}, v_{gt}, w_{gt}, \delta^s_g \in \{0, 1\} \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{38}$$

$$p_{gt}, p'_{gt}, \overline{p}'_{gt}, c^{\text{p}}_{gt}, c^{\text{SU}}_{gt}, r_{gt} \geq 0 \qquad g \in \mathcal{G}, t \in \mathcal{T} \tag{39}$$

The objective function (1) seeks to minimize the total cost, which is composed of the energy production cost c^{p}_{gt}, the fixed cost of operating at a minimum production level C^{R}_g. The variable startup cost c^{SU}_{gt} and the shutdown cost c^{SD}_{gt} for each generator $g \in \mathcal{G}$ during a specific time period $t \in \mathcal{T}$.

The minimum uptime and minimum downtime constraints are represeented by Constraints (2)–(5). The logic of the generators' start-up, shut-down, and operation is modeled by Constraints (6). The generation limit requirements are imposed by Constraints (7)–(9). The start and shut-down ramp limits are modeled by Cosntraints (10)–(12). The ramp-up and ramp-down limits are represented by Constraints (13)–(14). The staircase production cost are represented by Constraints (15)–(23). Note that Constraints (19), (20), (21), (22), and (23) are not necessary in the formulation, but they serve to tighten the variables' staircase production. The shut-down cost constraints are Given by Constraints(28).

Constraints (24)–(27) indicate the cost of the segment of the variable of start-up function cost $c^{S_g}_{gt}$ of the generator g, where $C^{\mathrm{S}}_{g,s}$ is the start-up cost in the category s of generator g in \$/MWh. The demand and reserve requirement are modeled by Constraints (29) and (31). Constraints (32)–(37) establish linear relationships between the power variables $p_{gt}, p'_{gt}, \overline{p}'_{gt}, \overline{p}_{gt}$. These constraints are used in the best UCP formulations [22], and help constrain the convex hull of the problem. Lastly, (38) and (39) define the nature of the decision variables.

Initial conditions: Constraints (6),(13),(14) 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 (27) assure the start-up variable δ_{gt} to be zero during the initial periods considering its minimum shut-down time if the generator has been offline [26].

4 Proposed Matheuristics

The proposed solution method for the thermal UCP consists of two phases: a construction phase, where an initial feasible solution is built, and an improvement phase, where five procedures, including KS and four tailored-made versions of LB, are applied to the constructed solution. In the remainder of the section, all these components are further explained.

4.1 Construction Phase

To the best of our knowledge, the best construction method for UCP is due to Harjunkoski et al. [19]. Their heuristic, referred as HGPS, is of the type relax and fit (R&F). The solution strategy of R&F methods is to reduce the size of the MILP to a smaller, easier-to-solve model called subproblem, fixing those variables likely to keep their values in the optimal solution and letting the solver decide among the others. Finally, the subproblem is input into the solver, expecting that the solver returns a high-quality solution. Our idea is to develop am enhanced R&F heuristic that better exploits the structure of our problem.

The HGPS heuristic 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 subproblem are those that satisfy the condition $\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \geq \underline{P}$. We will call this condition: Harjunkoski's rule. Let $\tilde{u}_{g,t}$ and $\tilde{p}_{g,t}$ be the values taken by the variables in the linear relaxation solution. Once the variables are fixed, the subproblem is solved, deciding over the rest of the non-fixed variables.

Our heuristic, named HARDUC, starts by solving a linear relaxation of the UCP. Then, to form the subproblem, 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.

Figure 1 illustrate the difference between these two heuristics, indicating the space of $u_{g,t}$ variables. On the left-hand side, 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 solving the subproblem. On the right-hand side, 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 criterion $u_{g,t} : \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} < \underline{P}\} \cap \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \neq 0\}$ will be used for the construction of a restricted candidate list (RCL). We expect these variables are more likely to be in the optimal solution than those with zero values. This RCL is essential for other improving methods based on local branching described in this work.

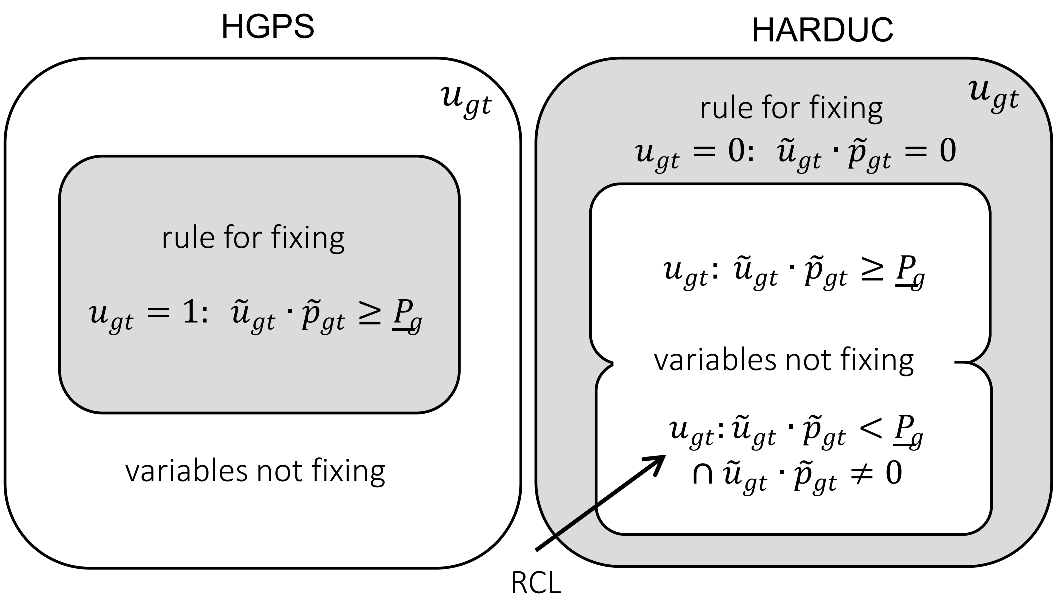


Figure 1: Rules for R&F construction methods. The shaded area represents the subset of variables to be fixed in the subproblem.

After obtaining a feasible solution, the search for an improved solution starts as a second phase of the solution algorithm. We develop a family of five metaheuristics, including one based on KS and four different versions of LB. All these methods are described in the following sections.

4.2 Improvement Phase

Once an initial feasible solution has been obtained, an improvement phase is applied in order to improve the solution. For this purpose, two main solution frameworks are proposed and explained below.

Kernel Search

Kernel search is a matheuristic initially developed by Angelelli et al. [2] to solve a multidimensional knapsack problem with outstanding results. We have not seen any KS implementation for UCPs, to the best of our knowledge. 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 the LP relaxation) and divided into small groups called buckets.

The expansion phase consists of solving the multiple MILP subproblems formed by each bucket concatenated one by one with the kernel, one step 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 subproblem 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 that are not part of the subproblem must be fixed to zero. The conventional KS terminates when each bucket has been resolved with the kernel or the allowed time is met [2].

Implementation of KS for the UCP

We present a version of KS that can be seen in Algorithm 1 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. KS is only used as an improvement method, as a constructive method that 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 [36]: $nbucks = 1 + 3.322 \ln(|U|)$.

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^{desc}.

The set of buckets of size $nbucks$, into which the variables of set U^{desc} are divided, is denoted as B_i, where i is an index ranging from 1 to $nbucks$.

Algorithm 1 `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
    cutoff ← z*                                                        ▷ Initialization phase
    K ← kernel formed from the variables that fulfill u_{g,t} : u_{g,t} = 1 ∈ x̄
    P^K ← fixing(1,K,P) fixing to 1 all the variables into kernel
    {x̂, ẑ} ← solveLR(P^{fix(K)})
    U ← select the variables that fulfill u_{g,t} ∉ K
    nbucks ← 1 + 3.322 ln(|U|)
    U^desc ← sort U in descending order using its reduced cost values
    {B_i}_{i=1}^{nbucks} ← buildbuckets(U^desc,nbucks)
    tl ← (t_total − elapsed_time)/n
    for (i = 1, nbucks) do                                             ▷ Expansion phase
        P^{K∪B_i} ← fixing(0,B̄_i,P)                    ▷ fixing to 0 all variables of the other buckets
        P^{K∪B_i} ← P^{K∪B_i} ∪ {∑_{j∈B_i} x_j ≥ 1} ▷ A variable from the bucket is forced into entering the kernel
        {stat, x̃, z̃} ← solve(P^{K∪B_i}, cutoff, tl)
        if (stat = feasible) then
            if (z̃ < z*) then
                z* ← z̃                                             ▷ Update best known objective
                cutoff ← z̃                                         ▷ Tighten cutoff for next iteration
                x* ← x̃                                             ▷ Update best solution
            end if
            K ← update the kernel with the variables in B_i such that u_{g,t} = 1 in x̃.
        end if
    end for
end while
return x*
```

Function `buildbuckets`(U^{desc},*nbucks*), depicted in Algorithm 2, 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^{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, `buildbuckets` returns all the buckets $\{B_i\}_{i=1}^{nbucks}$ as the output.

During the expansion phase of 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*

Algorithm 2 `buildbuckets()`

Input:
U^{desc}=list of $u_{g,t}$ outside the kernel, in descending order,
nbucks=number of buckets
Output:
A set of buckets $\{B_i\}_{i=1}^{nbucks}$

$\{B_i\} \leftarrow \{\phi\}$
$start \leftarrow 0$; $end \leftarrow 0$
$n \leftarrow |U^{\text{desc}}|$
$k \leftarrow \lfloor n/nbucks \rfloor$
$remainder \leftarrow n \mod (nbucks)$
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 $\{B_i\}_{i=1}^{nbucks}$

representing the upper bound value, and *tl* representing the maximum solution time. At each iteration, the kernel is updated, adding the bucket 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.

Local Branching

This matheuristic was proposed by Fischetti and Lodi [14] 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 $N(\overline{x}, k)$. The distance from solution $\overline{x}$ to other solutions is calculated using the hamming distance $\Delta(x, \overline{x})$, 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 $\overline{x}$ to other variables outside of binary support ($\overline{BS}$). The binary support of a solution $\overline{x}$ 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) (40) to problem P.

Unlike Fischetti and Lodi, which considers all binary variables of a problem to form a *BS*, in this work, 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 the number k as defined by the equation:

$$\Delta(x, \bar{x}) = \sum_{j \in BS} (1 - u_j) + \sum_{j \in \overline{BS}} u_j \leq k \tag{40}$$

In our UCP, the binary variables $v_{g,t}$, $w_{g,t}$, and $\delta_{g,t}$ contained in the formulation are not considered to build 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. [8] and is used in this particular version of LB to solve a graph edit distance problem.

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

$$\text{left-branch :} \quad \Delta(x, \bar{x}) \leq k \tag{41}$$

$$\text{right-branch :} \quad \Delta(x, \bar{x}) \geq k + 1 \tag{42}$$

$$\text{tabu :} \quad \Delta(x, \bar{x}) \geq 1 \tag{43}$$

$$\text{soft-fixing :} \quad \sum_{j \in BS} \bar{x}_j x_j \geq 0.9 \sum_{j \in BS} \bar{x}_j \tag{44}$$

The constraints in the LB enlisted, as proposed by Fischetti and Lodi in 2006, are crucial for efficiently narrowing the feasible space in solving MILPs. These constraints, which include left-branching, right-branching, tabu constraints, and a soft-fixing strategy, play a key role in the algorithm's effectiveness. This approach, outlined in Fischetti and Lodi's original work, significantly contributes to solution improvement by adjusting the parameter k and implementing diversification mechanisms. Overall, these constraints are essential for the LB method to target near-optimal solutions in MILP problems.

Our implementation of LB for the UCP

This section outlines four versions of LB that differ from the original idea [14]. 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 $u_{g,t} : \{\tilde{u}_{g,t} \cdot p_{g,t} < \underline{P}\} \cap \{\tilde{u}_{g,t} \cdot \tilde{p}_{g,t} \neq 0\}$, a representation of this subset of variables can be shown in the Figure 1; note that the RCL contains the variables that are not zero but are below the minimum power value of the generator $\overline{P}_g$, these variables are discarded by Harjunkoski et al. [19] 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 $\sum\limits_{j\in BS} \bar{x}_j x_j \geq 0.9 \sum\limits_{j\in BS} \bar{x}_j$ proposed by Fischetti and Lodi [14]. 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 (6) 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. [26] and Morales-España et al. [28] 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 (44) are not applied.

- LB3: This version is the closest to the original one proposed by Fischetti and Lodi [14]. 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 of 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 $u_{g,t}$ variables to define the BS, unlike the generic LB of Fischetti and Lodi [14], where all binary variables participate in the local search.

5 Computational Experience

In this section, we present the evaluation of the proposed algorithms. All methods were coded in Python version 3.10.0 and the algebraic modeling language Pyomo version 6.6.1 using IBM(R) ILOG(R) CPLEX(R) version 22.1.0.0 as the optimization solver. Tests were carried out in a 64-bit platform with 64 GB of RAM and 2.50 GHz Intel(R) i7(R) 11700 CPU, 65W, on a Linux Ubuntu version 20.0 operating system.

For all tests the relative optimality gap (ROG) was computed as follows:

$$\text{ROG} = \frac{|LowBound - z\_best|}{((1 \times 10^{-10}) + |z\_best|)} \cdot 100\%$$

where $LowBound$ corresponds to the largest lower (dual) bound obtained by the branch-and-bound solver and z_best, to the best feasible integer solution found by the corresponding method.

5.1 Description of test instances

A total of 83 instances, divided into three main subsets (see Table 2), were used to evaluate the proposed methods. The first subset, x7day_small, consists of 20 instances (10 from Morales-España et al. [26] and 10 derived from an eight-generator dataset proposed by Ostrowski et al. [32]); while the second (x7day_medium) and third (x7day_large) subsets include 33 and 30 instances, respectively, constructed by gradually increasing the number of generators and randomly combining them to enhance complexity, based on data from Ostrowski et al. [32]. The planning horizon for all the sets is composed of 168 periods, equivalent to seven days.

Table 2: Description of test instances.

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

The demand profile for each instance, shown in Table 3, is obtained by multiplying the demand profile and 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 must be met for each hour.

Table 3: 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%

5.2 Comparison of constructive methods

The goal of this first experiment is to evaluate the proposed construction heuristic, HARDUC. To this end, we compare HARDUC, HGPS [19], and the best solution found by CPLEX (CBS, for CPLEX Best Solution) on all the instances. CPLEX algorithmic parameters such as *emphasis_mip*, *mip_strategy_file*, and *mip_tolerances_mipgap*, were configured as detailed in Table 4. The maximum computing time was set to 1200 s. The table shows the average ROG (AROG) and its standard deviation (ROGSD), as well as the number of feasible solutions (NFS) that the each method successfully found.

Table 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	1×10^{-5}	Relative tolerance between the best integer and the best lower bound

Table 5 shows the comparison of the obtained results. These results indicate that the proposed HARDUC achieved the lowest AROG among all the sets of instances, indicating solutions closer to optimality than CBS and HGPS. It also obtained the lowest ROGSD, ensuring more stable performance. CBS showed the highest AROG and ROGSD in most of the cases. HGPS performed better than CBS but slightly worse than HARDUC. In terms of the number of feasible solutions, HARDUC successfully found feasible solutions for all the sets of instances. HGPS identified 20/32/29 feasible solutions for the sets of small-, medium-, and large-scale instances, respectively. Finally, CPLEX found 20 feasible solutions in the x7day_small group (100%), 26 out of 33 in the x7day_medium group (78.7%), and 18 out of 30 in the x7day_large group (63.3%).

Table 5: Comparison of constructive methods.

	x7day_small			x7day_medium			x7day_large		
Method	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

Therefore, HARDUC was chosen to obtain an initial feasible solution for the proposed matheuristics evaluated in the next section since it obtained the best performance.

5.3 Evaluation of metaheuristics performance

In the following tests, the proposed methods (KS and the four versions of LB) are evaluated under two different computation times: 4000 s. and 7200 s. (including the time for obtaining the initial feasible solution). For all LB versions, each iteration has a time limit of 1200 s. Furthermore, the results obtained by solving model TUCP ((1)-(39)) using CPLEX, are analyzed. For this purpose, two versions were considered: SM1, in which the model is completely solved by CPLEX, and SM2, where an initial feasible solution is given as a staring solution (MIPStart) to the solver. For all methods (except SM1), the same initial solution is considered. Certain parameters used by CPLEX were adjusted as specified in Table 6.

Table 6: CPLEX algoritmic parameters for matheuristic evaluation.

Method	Parameter	Value	Description
SM1	emphasis_mip	1	Emphasize feasibility over optimality
	mip_strategy_file	3	Node file on disk and compressed
	mip_tolerances_mipgap	1×10^{-5}	Relative tolerance between the best integer and the best lower bound
Remaining	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	1×10^{-5}	Relative tolerance between the best integer and the best lower bound
	preprocessing_symmetry	0	Turn off symmetry breaking

For the relevant criteria specific to each proposed solution method, Table 7 illustrates how these were applied.

Table 7: Criteria considered for the KS and the LB methods.

LB1	LBC built using dominant variable $u_{g,t}$. RCL from Harjukovski's rule [19]. Soft-fixing to 90% of binary support.
LB2	LBC built using dominant variable $u_{g,t}$. RCL from Harjukovski's rule [19].
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	Kernel and buckets are built only with dominant variables $u_{g,t}$. Number of buckets is calculated by Sturge's rule [36]. Buckets are built using the reduced costs from linear relaxation (fixing kernel).

Table 8 shows the AROG, the ROGSD, and the NFS obtained for each method considering the set of test instances. As can be seen, on average, increasing the maximum computation time reduce the average ROG and its corresponding standard deviation in most cases. Considering the significant impact that improvements can have, even if they are perceived to be small, we can visualize that by assigning one more hour of optimization (which is still within the time allowed in the real case), we can further improve the solutions obtained when the optimization time is 4000 s.

Our proposed construction heuristic, HARDUC, found feasible solutions for all instances within the time limit. However, some medium- and large-scale instances were excluded in the computations under the 7200 s. option because CPLEX, the base solver, failed to solve the linear programming relaxation at the root node of the branch-and-bound method so that no lower bound was available.

Finally, the results are also plotted for all methods and each set of instances. A statistical analysis is conducted only for those instances in which the method successfully found a solution. Figures 2a, 3a, and 4a present the distributions of the relative optimality gap for each method. Each series represents a method, with the horizontal axis indicating the relative optimality gap. The Y-axis represents the density, which is a continuous estimate of the relative frequency. Therefore, the area under the curve must equal 1. The figures show that as instance difficulty increases, the KS and LB methods yield results with significantly lower variance compared to the solver (SM1, SM2). Additionally, the KS and LB methods achieve higher accuracy, with an average relative optimality gap closer to zero than the solver. To assess the significance of these improvements, we conducted statistical tests by analyzing variance and comparing means.

Similarly, Figures 2b, 3b, and 4b provide evidence of the relative optimality gap distributions for each method at 7200 s. It can be noticed that, as instance difficulty increases, the KS outperforms the other methods, showing a lower variance and a superior accuracy of AROG.

An analysis of variance (ANOVA) was conducted for each set of instances in both cases. The results reject the null hypothesis of equal means, confirming that at least one method differs sig-

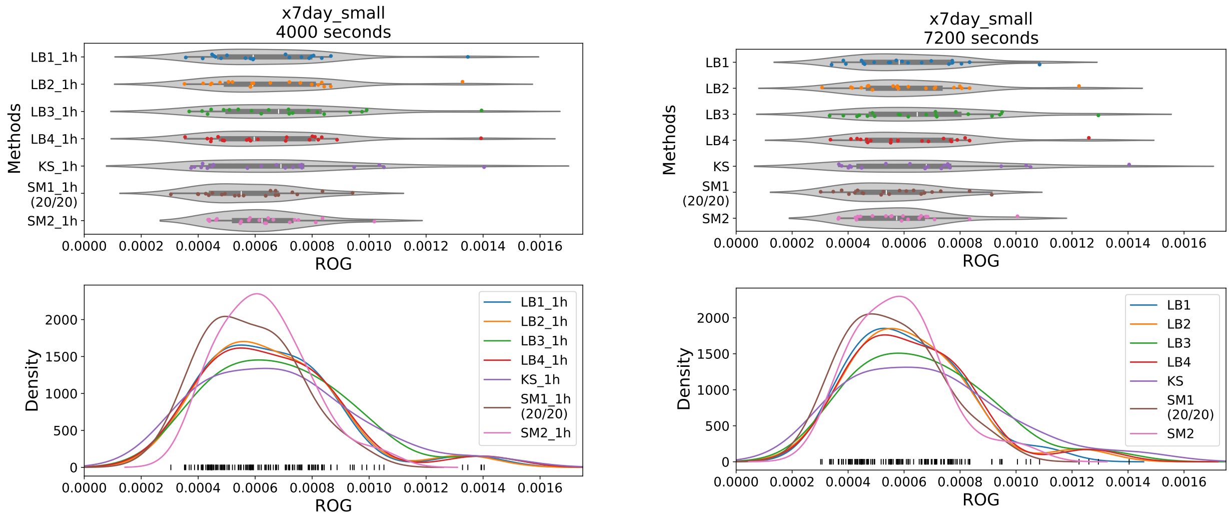
Table 8: Performance comparison among all solution methods considering different maximum time limits

Set	Method	4000 s			7200 s		
		AROG	ROGSD	NFS	AROG	ROGSD	NFS
x7day_small	LB1	6.65E-4	2.20E-4	20	6.11E-4	1.82E-4	20
	LB2	6.56E-4	2.18E-4	20	6.22E-4	2.00E-4	20
	LB3	6.98E-4	2.44E-4	20	6.67E-4	2.31E-4	20
	LB4	6.60E-4	2.30E-4	20	6.31E-4	2.06E-4	20
	KS	6.85E-4	2.63E-4	20	6.78E-4	2.67E-4	20
	SM1	5.79E-4	1.59E-4	20	5.56E-4	1.59E-4	20
	SM2	6.28E-4	1.49E-4	20	5.81E-4	1.55E-4	20
x7day_medium	LB1	3.74E-4	9.08E-5	26	3.36E-4	7.25E-5	26
	LB2	3.66E-4	9.49E-5	26	3.39E-4	7.67E-5	26
	LB3	3.82E-4	9.44E-5	26	3.50E-4	6.46E-5	26
	LB4	3.76E-4	9.19E-5	26	3.38E-4	6.91E-5	26
	KS	3.10E-4	7.48E-5	26	2.98E-4	6.73E-5	26
	SM1	4.73E-4	1.18E-4	26	3.13E-4	8.77E-5	26
	SM2	3.74E-4	9.08E-5	26	4.45E-4	1.07E-4	26
x7day_large	LB1	2.71E-4	6.42E-5	18	2.47E-4	5.75E-5	18
	LB2	2.81E-4	7.17E-5	18	2.42E-4	5.82E-5	18
	LB3	2.91E-4	7.36E-5	18	2.63E-4	6.44E-5	18
	LB4	2.90E-4	6.64E-5	18	2.54E-4	6.30E-5	18
	KS	2.25E-4	3.95E-5	18	2.08E-4	3.82E-5	18
	SM1	3.07E-4	8.77E-5	18	2.72E-4	7.33E-5	18
	SM2	3.33E-4	7.79E-5	18	3.19E-4	8.34E-5	18

nificantly from the others. Similar results are obtained when the maximum time limit allowed is 7200 s (see Appendix B for details).

It is important to realize that even very small relative optimality gaps, on the order of 0.01% or less, can lead to materially different market outcomes. These differences may not significantly affect the total cost, but they can shift unit commitment decisions enough to alter the marginal prices used for market settlements. Since these prices determine generator payments, slight variations can result in economic mismatches and fairness concerns [20].

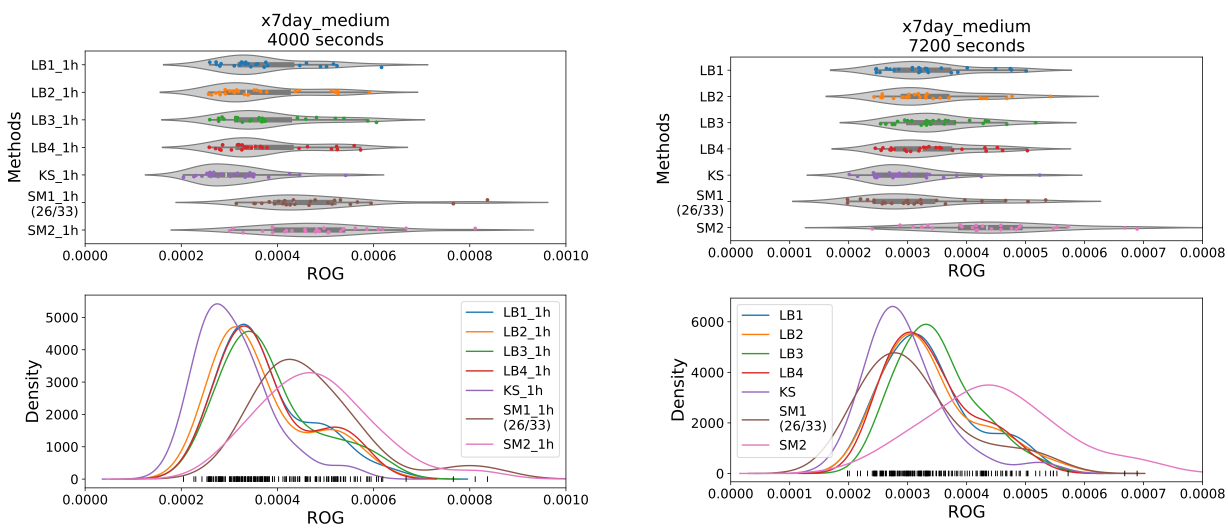
This sensitivity has been confirmed in several studies, where near-optimal solutions with nearly identical objective values led to significantly different pricing outcomes. As shown by Sioshansi et al. [37], small optimality gaps do not necessarily eliminate payment deviations or price volatility. Therefore, reducing the gap—even marginally—remains relevant for achieving more stable and economically efficient market results. In this work, we aim to contribute to reducing the optimality gap as a means to help mitigate, among other issues, the payment instability faced by generators.



(a) Time limit of 4000 seconds

(b) Time limit of 7200 seconds

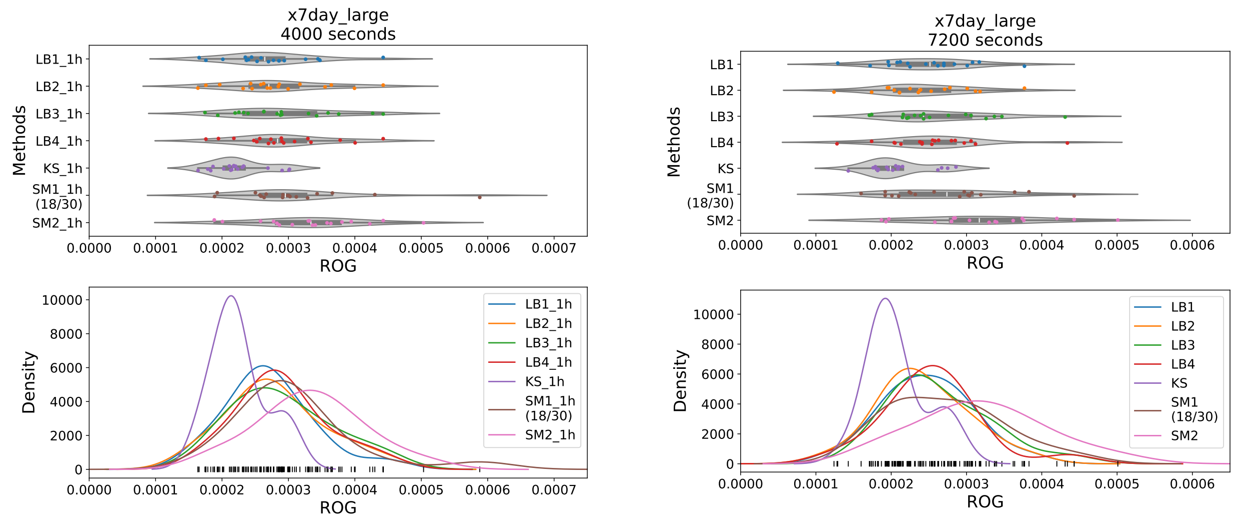
Figure 2: Relative optimality gap distribution of improvement methods on set x7day_large.



(a) Time limit of 4000 seconds

(b) Time limit of 7200 seconds

Figure 3: Relative optimality gap distribution of improvement methods on set x7day_medium.



(a) Time limit of 4000 seconds

(b) Time limit of 7200 seconds

Figure 4: Relative optimality gap distribution of improvement methods on set x7day_large.

5.4 Evaluation of special cases

To gain deeper insight, a more detailed analysis of the following cases is carried out as follows:

Case 1: Performance of methods in two instances where SM1 was unable to find a solution

Case 2: Performance of methods on two instances where SM1 found a solution but performed worse than the remaining methods

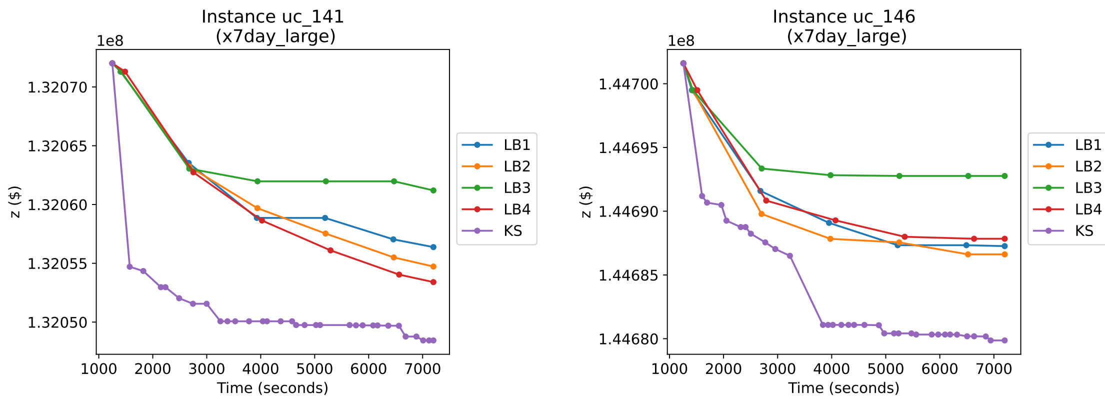
Case 3: Performance of methods on two instances for which SM1 obtained similar results to the remaining methods

Case 4: Comparison of the methods in instances where the KS shows stagnation.

Case 5: Comparison of methods in which KS failed to improve the initial solution

Analysis of Case 1

In this case two instances,uc_141 and uc_146 from the set of large-scale instances, are evaluated in which SM1 was not able to find a feasible solution. As can be seen in Figures 5a and 5b, the proposed solution methods (and their variants) were able to improve the initial solution found by HARDUC, with KS performing the best of all and LB3 showing the worst performance.



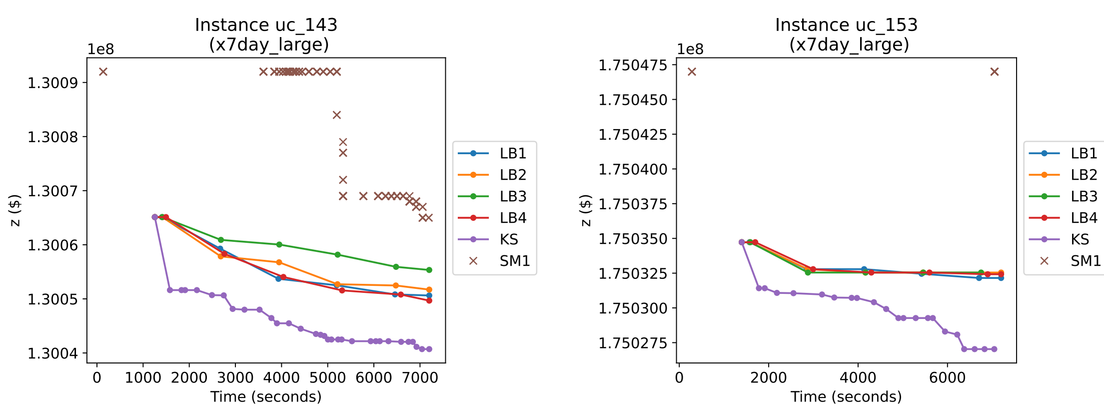
(a) Instance 141.

(b) Instance 146.

Figure 5: Comparison of methods on instances where the solver (SM1) could not solve within the allowed time limit.

Analysis of Case 2

In the second case, two instances (uc_143 and uc_153), depiucted in Figures 6a and 6b, illustrate a scenario where SM1 finds a solution, but its performance and quality are not competitive with the other methods. Here, the initial solution provided by HARDUC plays a crucial role in both the starting point and overall performance of the proposed methods. Although SM1 required a significant amount of time to achieve its first notable improvements, the other methods have taken advantage of the initial solution of HARDUC much earlier, allowing them to use the remaining time more effectively for refinement. Among these methods, KS again demonstrated superior performance and solution quality.



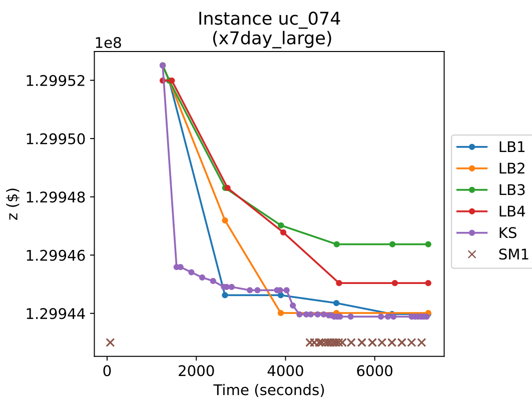
(a) Instance 143.

(b) Instance 153.

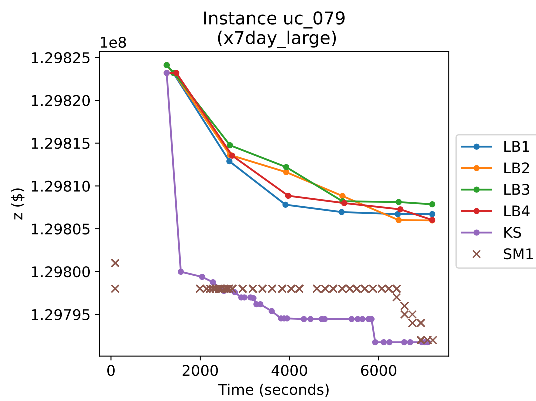
Figure 6: Performance of methods on two instances where SM1 found a solution but performed worse than the remaining methods

Analysis of Case 3

In some isolated cases, SM1 produced competitive or even superior results compared to metaheuristic methods. As shown in Figures 7a and 7b, the initial solution of SM1 was significantly better than the starting point provided by HARDUC. However, while the proposed methods quickly achieved substantial improvements in most cases, they were unable to surpass the solution of SM1 in these specific instances. Only the KS was able to arrive at a similar solution in the given time in one of the cases, being in general one of the proposed methods that showed a better performance. In particular, despite the strong initial solution of SM1, it struggled to make significant further improvements for most of the remaining time.



(a) Instance 074.

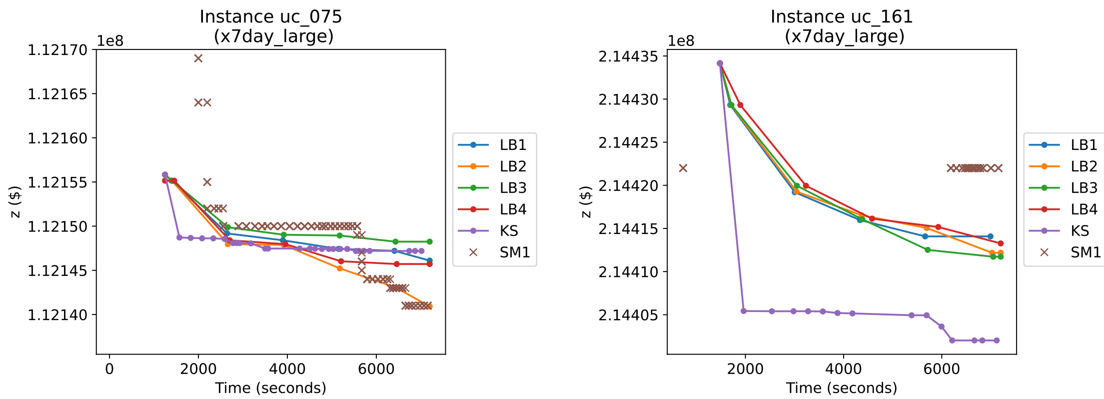


(b) Instance 079.

Figure 7: Performance of methods on two instances for which SM1 obtained similar results to the remaining methods

Analysis of Case 4

In Case 4, two instances show a particular pattern in the KS method. Initially, KS achieved significant improvements, but as iterations progressed, it reached longer periods of stagnation, where further enhancements became minimal. This stagnation prevented the algorithm from making meaningful progress in subsequent iterations. For example, in Figure 8a, the solution improved substantially during the first iteration, but showed little progress over the remaining 5000 seconds. A similar pattern is observed in Figure 8b, where although the solution improved considerably in the final 1000 seconds, it struggled for most of the runtime to escape a local optimum. This behavior may be due to the first buckets containing key variables that provide early improvements, while the remaining buckets, due to their size, failed to capture the ideal combination of variables needed for further optimization. It would be valuable to explore whether evaluating a larger kernel or expanding the number of buckets could improve efficiency and mitigate stagnation.



(a) Instance 075. (b) Instance 161.

Figure 8: Comparison of the methods in instances where the KS shows stagnation

Analysis of Case 5

In this final analysis, there are four cases where KS failed to improve the initial feasible solution provided by HARDUC (see Figures 9a, 9b, 9c, and 9d). In these cases, only the LB method and its variations were able to achieve better solutions, with slightly different performance depending on the stage solved. One possible approach to address this limitation is to analyze alternative kernel and hub sizes to determine whether tuning these parameters could increase the opportunity for further improvements.

Briefly speaking, since the solver did not find a feasible solution in some large-scale instances, we provided an initial starting solution using our HARDUC method. However, even with initial solution, the solver obtained less accurate results than the proposed improvement methods LB1-4, and KS.

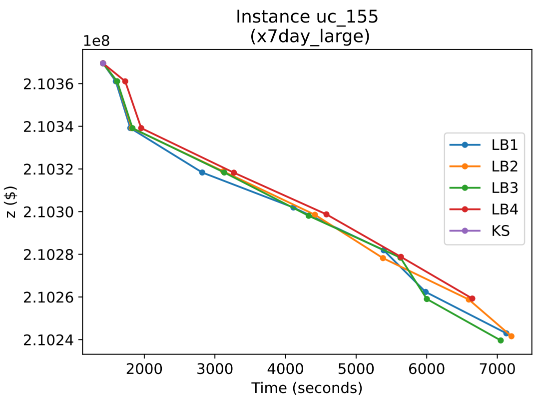
For small-scale instances, when the maximum time limit was set to 4000 s., the solver performed similarly to the LB1-4 and KS methods, with no significant accuracy. However, when the time was extended to 7200 s., the solver outperformed LB3 and SM2, achieving a smaller relative optimality gap.

For medium-scale instances with a 4000-second time limit, the KS outperformed the remaining methods. In addition, LB1-4 performed better than SM1 and SM2.

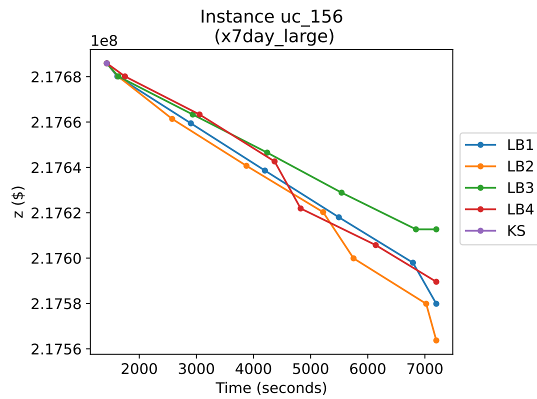
For large-scale instances, in general, KS was again the most efficient method for both time limits. In addition, LB1 performed better than SM1, and SM2 outperformed SM1.

It was observed that all LB methods outperformed SM2. However, when the time limit was set to 7200 s., no significant differences were found between LB1-4 and SM1 methods. It is important to note that the solver did not successfully solve all medium- and large-scale instances. Furthermore, if an initial solution is provided to the solver, the performance is noticeably lower compared when it is performed without it.

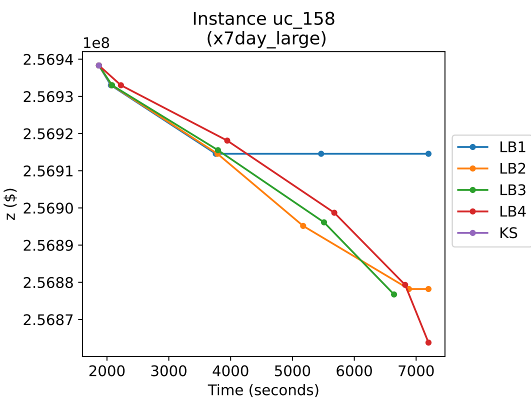
In general, among the methods, KS achieved the smallest average optimality gap. On the other hand, LB3, which closely follows the original method proposed by Fischetti and Lodi [14], has



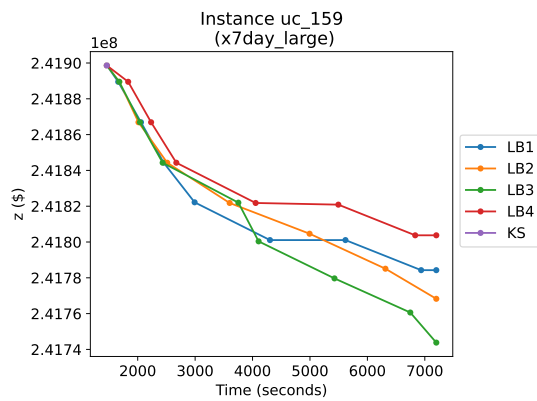
(a) Instance 155.



(b) Instance 156.



(c) Instance 158.



(d) Instance 159.

Figure 9: Comparison of methods on instances where the KS remained in the initial solution without improvement.

shown the lowest performance among the proposed LB methods.

Although KS generally provided better results in terms of relative optimality gap, it did not improve the initial HARDUC solution in approximately 4% of the instances. In contrast with the LB methods, which successfully improved the initial solution.

6 Conclusions

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

Among the methods based on LB, LB3 is the closest version to the original local branching, unlike LB1, LB2, and LB4, which are variants that implement the soft-fixing concept and a restricted candidate list. 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 assessed by solving three groups of instances classified into small-, medium-, and large-scale according to the number of generators they contained. A time limit of 4000 seconds and 7200 seconds was set for solving the instances using the LB1, LB2, LB3, LB3, LB4, and KS improvement methods, including the commercial solver. We also tried out feeding the commercial solver with a feasible solution found by the constuction heuristic.

The solver without an initial solution only solved approximately 79% of medium-sized instances and 63% of large-scale instances. In contrast, our methods, LB1, LB2, LB3, LB4, and KS, always had feasible initial solutions provided by our HARDUC method in the constructive phase.

In terms of solution quality, statistical tests showed no significant differences among the LB1, LB2, LB4, LB3, and KS methods and the solver on small-scale instances. However, for testing medium-scale instances within a maximum allowed running time of 4000 seconds, the KS performed best. The other methods outperformed the solver with an initial solution and matched the results of the solver without an initial solution. For tests with a maximum allowed running time of 7200 s., the solver performed in the same way as 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 in solution quality for the set of large-scale instances. Again, the KS performed best.

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-scale instances. According to the results, the proposed adaptations of local branching, including soft-fixing and RCL components, helped find better solutions in medium-scale instances instead of the original version.

Although 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. However, local branching methods offer a consistent improvement and have the capacity 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 are able to "dive" 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 matheuristic methods and the solver simultaneously on different computers. This strategy of diversifying solution methods always allows us to obtain feasible and high-quality solutions.

Future research: For future research, it would interesting the developing of a hybrid algorithm combining KS and the Local Branching Constraint (LBC) (40) to benefit from KS's rapid convergence in initial iterations and LBC's ability to escape local optima when KS stagnates. Another line of work could be designing a branch-and-cut method that includes local branching constraints, using callback functions to introduce cuts into the solver's solution tree, saving time and enhancing efficiency during problem solving. We think that a study to assess the proposed approach with various UCP models, considering variations such as prohibited zones, non-linear elements, elastic demand, interdependent generators, and network flow limits, offers the potential for evaluating its effectiveness across various UCP formulations. Another line of work, within the context of KS, consists of further studies about strategies for determining values for the kernel and bucket list size. Naturally, many of the ideas developed here can be extended to handle stochastic versions of the problem. Many of those solution algorithms rely on decomposition or scenario-based schemes where some subproblems might have similar structures.

Acknowledgments: Uriel Lezama-Lope acknowledges a scholarship for doctoral studies from the Mexican Council for Humanities, Science, and Technology (CONAHCYT). The research of Roger Ríos-Mercado was funded by CONAHCYT under grant CF-2023-I-880. The research of Diana Huerta-Muñoz has been supported by CONAHCYT through a postdoctoral fellowship.

Data Availability: The repository https://github.com/urieliram/tc_uc contains the data and source code for the methods that have been implemented.

References

[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] E. Angelelli, R. Mansini, and M. G. Speranza. Kernel search: A general heuristic for the multi-dimensional knapsack problem. *Computers & Operations Research*, 37(11):2017–2026, 2010.

[3] 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.

[4] 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.

[5] P. Bendotti, P. Fouilhoux, and C. Rottner. On the complexity of the unit commitment problem. *Annals of Operations Research*, 274(1):119–130, 2019.

[6] M. A. Boschetti, A. N. Letchford, and V. Maniezzo. Matheuristics: Survey and synthesis. *International Transactions in Operational Research*, 30(6):2840–2866, 2023.

[7] E. Danna, E. Rothberg, and C.L. Pape. Exploring relaxation induced neighborhoods to improve mip solutions. *Mathematical Programming*, 102(1):71–90, 2005.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98(1):23–47, 2003.

[15] M. Fischetti, F. Glover, and A. Lodi. The feasibility pump. *Mathematical Programming*, 104 (1):91–104, 2019.

[16] 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.

[17] 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.

[18] M. Håberg. Fundamentals and recent developments in stochastic unit commitment. *International Journal of Electrical Power & Energy Systems*, 109:39–48, 2019.

[19] I. Harjunkoski, 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.

[20] 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.

[21] 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.

[22] 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.

[23] V. Maniezzo, M. A. Boschetti, and T. Stützle. *Matheuristics: Algorithms and Implementations.* EURO Advanced Tutorials on Operational Research. Springer, Cham, Switzerland, 2021.

[24] L. Montero, A. Bello, and J. Reneses. A review on the unit commitment problem: Approaches, techniques, and resolution methods. *Energies*, 15(4):1296, 2022.

[25] G. Morales-España. *Unit Commitment: Computational Performance, System Representation and Wind Uncertainty Management.* PhD thesis, Universidad Pontificia Comillas, Madrid, Spain, October 2014.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] E. Nycander, G. Morales-España, and L. Söder. Security constrained unit commitment with continuous time-varying reserves. *Electric Power Systems Research*, 199:107276, 2021.

[32] 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.

[33] 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, June 2005.

[34] 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.

[35] 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.

[36] D. W. Scott. Sturges' rule. *WIREs Computational Statistics*, 1(3):303–306, 2009.

[37] 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.

[38] 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.

[39] 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.

[40] L. A. Wolsey. *Integer Programming*. Wiley, New York, 2nd edition, 2021.

[41] 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.

[42] 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.

A Results for Construction Heuristics

The distributions of the relative optimality gap results for each method are shown in Figures A1, A2, and A3. 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.

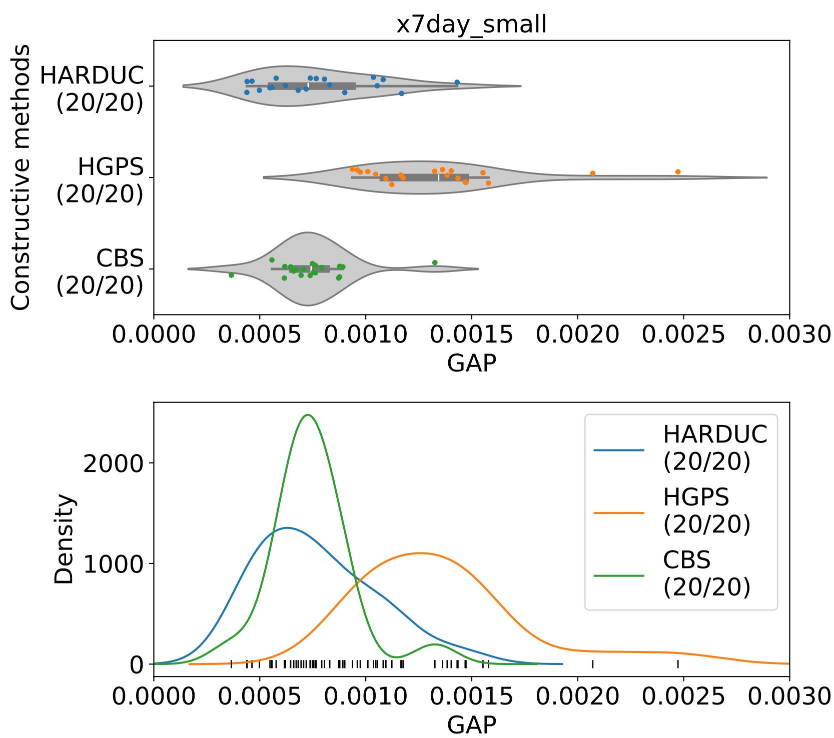


Figure A1: Relative optimality gap distributions of constructive methods in the instances group x7day_small.

The following is an analysis of variance, and mean difference 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 results of the analysis of variance applied to the results of the constructive methods are shown in Table A1. In the three groups of instances, the null hypothesis H_o of equality of means

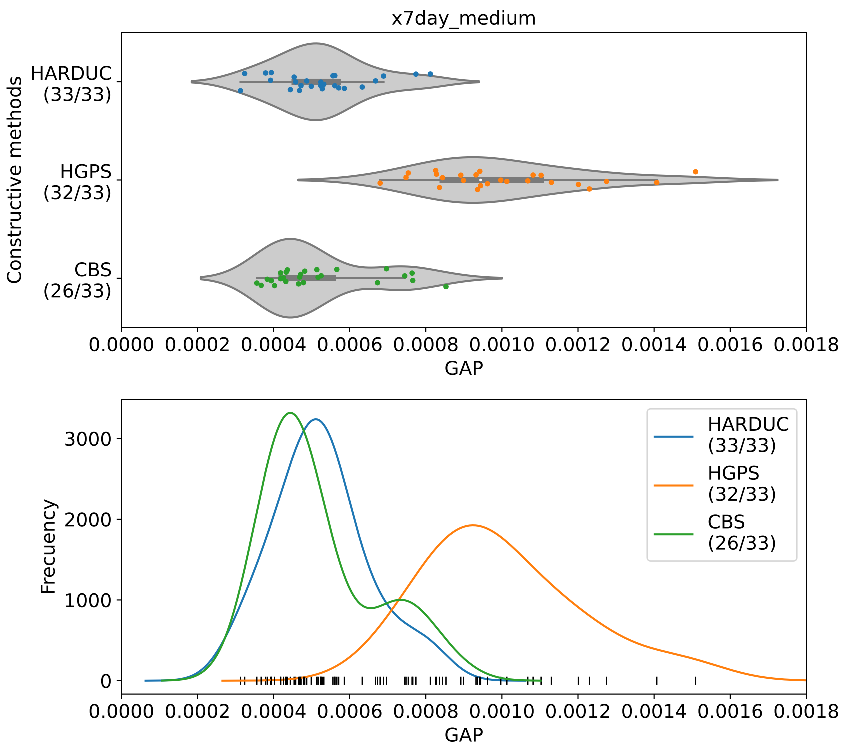


Figure A2: Relative optimality gap distributions of constructive methods in the instances group x7day_medium.

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 C1. The HARDUC method has a significantly lower average in the 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.

Table A1: 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

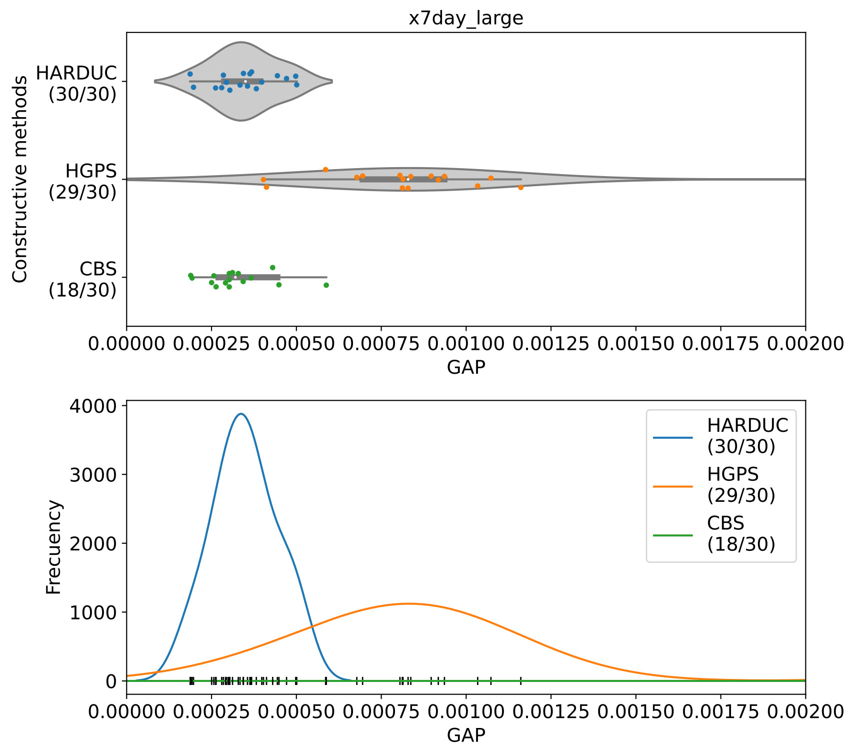


Figure A3: Relative optimality gap distributions of constructive methods in the instances group x7day_large.

Table A2: 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

B Results for Local Branching and Kernel Search

We perform statistical tests by analyzing variance and comparing means to determine whether the improvements are significant. The summary of the results of the analysis of variance for the three groups of instances can be found in Table B1.

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

Table B1: 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

and for the three groups of instances are presented in Tables C2, C4, and C6.

We carry out statistical testing to determine whether the improvements are significant by analyzing variance and comparing means. The summary of the results of the analysis of variance for the three groups of instances can be found in Table B2.

Table B2: 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

C Statistical Tests

Table C1: 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 C2: 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_o
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_o
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_o
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_o
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_o
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_o
LB2_1h-KS_1h: There is no difference between the two population means	Mann-Whitney	0.5484	We fail to reject H_o
LB2_1h-LB1_1h: There is no difference between the two population means	Mann-Whitney	0.4302	We fail to reject H_o
LB2_1h-LB4_1h: There is no difference between the two population means	Mann-Whitney	0.2714	We fail to reject H_o
LB2_1h-LB3_1h: There is no difference between the two population means	Mann-Whitney	0.2367	We fail to reject H_o
LB2_1h-SM2_1h: There is no difference between the two population means	Mann-Whitney	0.1617	We fail to reject H_o
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_o
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_o
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_o
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_o
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_o
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_o
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_o
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_o
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_o
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_o

* Significance level 0.05

Table C3: 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 C4: 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 C5: 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 C6: 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 C7: 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