

Fast Infeasibility Analysis of Network-constrained Unit Commitment Model

Yanhong Liu, Xinfei Yan, Haiwang Zhong, and Chongqing Kang

Abstract—With the rapid development of the electricity market and the increasingly stringent operational standards, the longstanding infeasible network-constrained unit commitment (NCUC) issue in market operations is calling for effective and systematic methods that can analyze and repair infeasible NCUC models. Studies show that a set of irreducible infeasible subsets (IISs) may lead to infeasibility in a model. The identification and repair of IISs can correct the original infeasible model, while traditional filtering algorithms for IIS location often exhibit computational inefficiency. A fast infeasibility analysis framework of NCUC and a relaxation filtering algorithm are proposed in this paper to analyze infeasible NCUC models more comprehensively and effectively. The infeasibility analysis framework employs subsystem-based IIS analysis method, generating and analyzing simplified models to repair the infeasible NCUC. The relaxation filtering algorithm can achieve greater effectiveness of IIS identification in the holistic NCUC compared with traditional general filtering algorithms. The case studies show that the proposed framework and algorithm can analyze and repair infeasible NCUC models effectively, reducing the analytical time by at least one order of magnitude compared with that of traditional general methods.

Index Terms—Network-constrained unit commitment (NCUC), electricity market, mixed-integer linear programming (MILP), infeasibility, irreducible infeasible subset (IIS), filtering algorithm.

I. INTRODUCTION

UNIT commitment (UC) stands as a pivotal element in the optimal scheduling of power systems and the clearing of electricity markets [1]. Typically, UC is formulated as a mixed-integer linear programming (MILP) problem. The consideration of network constraints transforms UC into network-constrained unit commitment (NCUC). In electricity markets, NCUC is typically utilized for day-ahead market clearing.

Occasionally occurring infeasible NCUC models have been a longstanding issue for market clearing. The evolution of modern power systems has complicated the corresponding NCUC problems, leading to longer computation time [2] and a greater risk of infeasibility.

As the electricity market develops rapidly, operational standards are increasingly stringent. Higher demands are placed on the precision and efficiency of NCUC optimization, while infeasible NCUC models usually hinder the optimization process and compromise the quality of optimization results. Therefore, it is necessary to improve the computational efficiency of the analysis and repair of infeasible NCUC models.

Currently, electricity market operators usually rely on historical operational data to rectify non-convergent or unreasonable clearing outcomes, which leads to problems in both efficiency and precision. In the Pennsylvania-New Jersey-Maryland (PJM) electricity spot market, if a maximum emergency generation or minimum capacity emergency occurs, PJM operators sequentially adopt measures such as adjusting the online generation output, the number of offline units, and the remaining price-sensitive demand to ensure grid security [3]. Similarly, in spot markets such as those in Guangdong [4] and Shanxi [5], China, if an infeasible clearing result arises, the market operators can adjust the operational boundary conditions, modify the unit constraints, or employ other effective measures deemed suitable by the operator and solve NCUC again to achieve a security-compliant clearing result.

The current practices rely heavily on human experience and judgment, resulting in incomplete and inefficient analysis with limited adaptability and universality; this affects the efficiency and precision of the analysis and repair of infeasible NCUC models. Furthermore, since market clearing outcomes directly impact the benefit of various market participants, related adjustments may deviate from the optimal clearing results, leading to concerns about market fairness [6]. Hence, there is an urgent need to conduct deeper research on more efficient, precise, and more systematic analysis and repair methods for infeasible NCUC models. If the conflicted constraints can be precisely located, market operators can make more informed decisions regarding how to modify the conflicted constraints, thereby restoring model feasibility and minimizing market losses. Some studies on infeasible NCUC models have been conducted. The infeasibility of NCUC problem may arise from diverse factors, includ-

Manuscript received: August 13, 2025; revised: November 18, 2025; accepted: January 5, 2026. Date of CrossCheck: January 5, 2026. Date of online publication: February 27, 2026.

The work was supported in part by National Science and Technology Major Project on Smart Grid (2030) (No. 2025ZD0805400) and in part by National Natural Science Foundation of China (No. U24B2077).

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>).

Y. Liu, X. Yan, H. Zhong (corresponding author), and C. Kang are with the State Key Laboratory of Power System Operation and Control, Department of Electrical Engineering, Tsinghua University, Beijing, China, and H. Zhong and C. Kang are also with the Sichuan Energy Internet Research Institute, Tsinghua University, Chengdu, China (e-mail: lyhthu@126.com; yanxinfei1994@126.com; zhonghw@tsinghua.edu.cn; cqkang@tsinghua.edu.cn).

DOI: 10.35833/MPCE.2025.000736



ing inaccurate telemetry data, erroneous data submissions by market participants, improper mandatory start-up or maintenance schedules, extreme load conditions, and overly conservative reserve or security constraint settings [7]. Various methods are employed to repair infeasible NCUC models such as multi-objective optimization [7], heuristics [8], and elastic programming (EP) [9], [10]. These studies focus on the repair of infeasible NCUC models, having limitations in analyzing causes of infeasibility and improving the operation conditions of systems. Regarding infeasibility analysis and repair for general MILP problems, progress has been made in previous studies, which can be applied to infeasible NCUC models.

1) Identification of Infeasible Key Constraints

The overall general process of analysis and repair of infeasible NCUC models is shown in Fig. 1, where x is the vector of continuous decision variables; y is the vector of N -dimensional binary decision variables; $f_{UC}(\cdot)$ is the objective function; and $g(\cdot)$, $h(\cdot)$, and $k(\cdot)$ are the constraint functions. After NCUC is determined to be infeasible by the solver, the infeasibility analysis is conducted to locate the infeasible key constraint set. The infeasibility repair algorithm is applied to the set to restore its feasibility. If NCUC is still infeasible after infeasibility repair, aforementioned processes are performed repeatedly until the NCUC restores its feasibility.

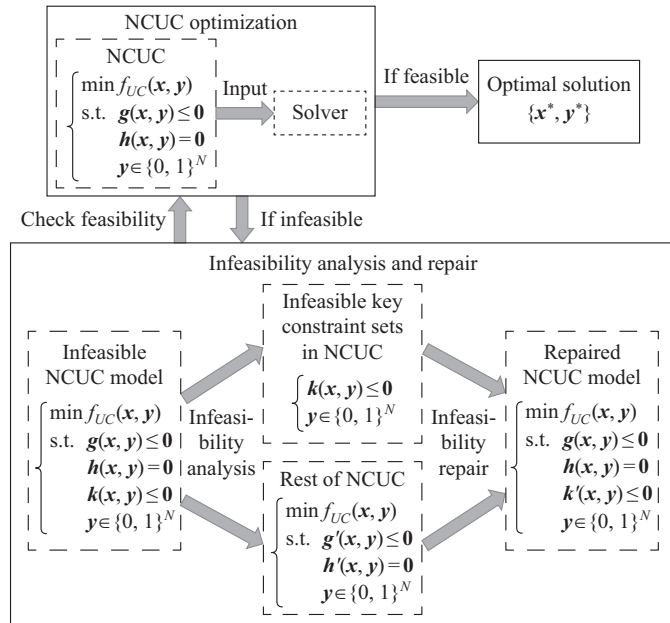


Fig. 1. Overall general process of analysis and repair of infeasible NCUC models.

The identification of infeasible key constraint sets is central to infeasibility analysis and repair. An infeasible key constraint set is the set of constraints that need to be adjusted to restore the feasibility of the model. It is possible for multiple infeasible key constraint sets to coexist in an NCUC model. Currently, there are two primary approaches for identifying infeasible key constraint sets: EP and locating infeasible subsets (ISs) [11].

2) EP

EP assumes that the constraints in the model are not rigid but elastic. By introducing slack variables into the constraints, the model is transformed into an EP model. By solving this model with an objective to minimize the sum of the magnitudes or the squares of slack variables and obtaining the optimal solution, infeasible key constraint sets can be identified, which contain the constraints with non-zero value of slack variables. If the values of slack variables fall within an acceptable range, the feasibility of the model is restored. To adapt to the practical problems, the objective coefficient of slack variables and the constraints that are slack can be modified [9], [12].

3) Locating ISs and Irreducible Infeasible Subsets (IISs)

The process of locating ISs focuses on the identification of ISs. The original NCUC model typically contains many constraints, which poses challenges for direct infeasibility analysis and repair. In fact, the cause of model infeasibility lies in the existence of ISs within the model. These ISs are typically much smaller in scale than the original model, which facilitates infeasibility analysis and repair. Furthermore, ISs are easier for model users to comprehend, enabling the provision of more reasonable and more explainable feasibility repair solutions for the model.

Locating ISs in infeasible NCUC models is trivial. However, large-scale ISs are difficult to analyze, so only a few small-scale ISs are useful and can be used to repair the model. Locating the IIS is much more difficult, which can contribute much more to the repair of infeasible NCUC models.

The IIS is a subset such that when the constraints and variable bounds within this subset are isolated from the model, the subset remains infeasible. However, upon removing any single constraint or variable bound from this subset, the subset becomes feasible [13]–[15]. In other words, the IIS itself is infeasible, yet all of its proper subsets are feasible, which indicates that IISs are valid infeasible key constraint sets.

The definition of the IIS indicates that once all the IISs in an infeasible NCUC model are repaired, the feasibility of the model is restored. The analysis and repair of an infeasible NCUC model can be translated into the analysis and repair of IISs in the model. For MILP, the identification of IISs is achieved mainly by general filtering algorithms. The deletion filter [16] and the addition filter [16] are the two most basic filtering algorithms. Based on these two filtering algorithms, other filtering algorithms have been developed, most of which are combinations of filtering algorithms and other algorithms, such as the addition-deletion filter [17], generalized filtering algorithms [18], and the deletion filter with pre-solving algorithms [19]. Focusing on the characteristics of MILP, several specific strategies including safety bounds have been proposed to accelerate the identification of an IIS [17].

Currently, commercial solvers predominantly employ filtering algorithms for the computation of the IIS [20]. Furthermore, fewer constraints in an IIS reveal more insights regarding the cause of conflict, thereby offering greater assistance in repairing the original model [13], [14], [21]. Nevertheless,

computing the minimum IIS is known to be nondeterministic polynomial-time hard (NP-hard) [21]. Drawing upon the elastic filtering algorithm and observed fundamental facts, a polynomial-time heuristic that compares the sum of infeasibilities for minimum IIS in linear programming (LP) is introduced [22], but an efficient algorithm for MILP remains elusive.

4) Limitations of Existing Methods

The existing two types of methods have certain limitations. First, EP is a post-solution infeasibility analysis and repair method. One full NCUC model with slack variables needs to be solved, which is NP-hard and time-consuming. Occasionally, iterations are performed when the solution is impractical, which is more time-consuming.

Second, locating ISs or IISs is a pre-solution infeasibility analysis and repair method, avoiding the solution of the full NCUC model. However, it is time-consuming for MILP because it involves multiple determinations of the feasibility of MILP [17], [21], which is nondeterministic polynomial-time complete (NP-complete) [23]. The computational burden of a direct IIS analysis can sometimes exceed that of solving the original NCUC model itself. Infeasible real-world systems often contain multiple IISs with overlapping constraints, where information from a single IIS is insufficient to repair the original NCUC model.

Additionally, both types of methods are general and have limitations in utilizing the characteristics of the NCUC problem. If the mathematical and physical characteristics of the NCUC can be properly and completely considered, the analysis process and algorithm can be improved, thus increasing the computational efficiency and precision.

To further increase the computational efficiency and precision of analysis for infeasible NCUC models, a fast infeasibility analysis framework and a relaxation filtering algorithm for NCUC are proposed in this paper. The proposed framework and algorithm can rapidly locate IISs, and further improve overall infeasibility analysis and repair process by combining with existing slacking based repair methods. The major contributions of this paper are as follows.

1) A fast infeasibility analysis framework employing a sub-system-based IIS analysis method for NCUC is proposed, enabling more efficient and accurate pre-solution infeasibility analysis compared to conventional methods.

2) Building upon existing filtering algorithms and constraint relaxation, a relaxation filtering algorithm is developed, improving the efficiency of IIS identification for holistic NCUC compared to traditional filtering algorithms.

5) Differences with Infeasible NCUC Solution Analysis

To clarify, this paper focuses on analyzing infeasible NCUC models rather than infeasible NCUC solutions. While previous studies have typically addressed the issue of infeasible NCUC solutions, particularly within the context of optimizing NCUC models through methods such as Lagrangian relaxation [24], heuristics [25], and other strategies [26]–[28] or conducting fast simultaneous feasibility tests (SFTs) [29], these methods operate under the assumption that the underlying model itself is feasible. For instance, given an NCUC solution, the SFT computes branch flows, checks for limit vio-

lations, and returns violations with flow-injection power sensitivity coefficients. Prior research on infeasible NCUC solutions has aimed to enhance the feasibility testing of NCUC models by proposing solutions derived from either single-unit problem or NCUC models with relaxed security constraints. Recently, more attention has been paid to feasibility-guaranteed data-driven methods for NCUC solutions [30], [31].

These methods have three essential distinctions from infeasible model analysis.

1) Most infeasible solution repair methods operate on given solutions, while no solution can be obtained for infeasible model diagnosis usually.

2) Most infeasible solution repair methods modify the given infeasible solutions rather than repair the infeasible models.

3) The models considered by infeasible solution repair methods are feasible whereas infeasible model diagnosis deals with inherently infeasible models. Therefore, these methods become ineffective when the fundamental model is inherently infeasible and no solution exists—a scenario precisely addressed in this paper. Table I illustrates the gaps between infeasible model diagnosis and infeasible solution repair.

TABLE I
GAPS BETWEEN INFEASIBLE MODEL DIAGNOSIS AND INFEASIBLE SOLUTION REPAIR

Task	Model feasibility	Target of modification	Technique
Infeasible model diagnosis	Infeasible	Model parameters	Filtering algorithm
Infeasible solution repair	Feasible	Solutions	SFT and solution polishing

The rest of the paper is organized as follows. In Section II, the formulation of the NCUC is introduced. In Section III, the fast infeasibility analysis framework for NCUC is proposed. In Section IV, the relaxation filtering algorithm for holistic NCUC analysis is introduced. In Section V, case studies are presented to evaluate the computational efficiency and precision of the proposed framework and algorithm. Section VI presents the conclusion.

II. FORMULATION OF NCUC

In this section, a general NCUC model is constructed, which is formulated as follows.

The objective function of NCUC is formulated as (1), which minimizes the total cost of generation, start-up, and shut-down.

$$\min \left(\sum_{i=1}^{N_G} \sum_{t=1}^{N_T} f(p_{i,t}) + \sum_{i=1}^{N_G} \sum_{t=1}^{N_T-1} g(x_{i,t}, y_{i,t}) \right) \quad (1)$$

where $f(\cdot)$ is the unit cost calculation function, which is a quadratic function usually replaced with piecewise linear functions; $g(\cdot)$ is the unit start-up and shut-down cost function (when a unit switches from on to off or from off to on, the start-up or shut-down cost is calculated once); $x_{i,t}$ is the start-up status binary variable, which is equal to 1 if unit i

starts up in period t ; $y_{i,t}$ is the shut-down status binary variable, which is equal to 1 if unit i shuts down during period t ; $p_{i,t}$ is the power output of unit i during period t ; N_T is the number of periods; and N_G is the number of units.

Constraint (2) represents the system balance. Constraint (3) represents unit power generation limits. The system operation reserve is considered in constraint (4). Constraints (5) and (6) represent the ramp up and ramp down limits, respectively. Constraints (7) and (8) represent the minimum up and down time limits, respectively. Constraints (9) and (10) represent the logical limits among the unit on-off state indicator variable, unit start-up variable, and unit shut-down variable. The network constraints are considered as (11).

$$\sum_{i=1}^{N_G} p_{i,t} = \sum_{j=1}^{N_D} d_{j,t} = D_t \quad \forall t \quad (2)$$

$$\underline{P}_i u_{i,t} \leq p_{i,t} \leq \bar{P}_i u_{i,t} \quad \forall i, \forall t \quad (3)$$

$$\sum_{i=1}^{N_G} (u_{i,t} \bar{P}_i - p_{i,t}) \geq R_t \quad \forall t \quad (4)$$

$$p_{i,t} - p_{i,t-1} \leq RU_i \cdot u_{i,t-1} + \underline{P}_i (u_{i,t} - u_{i,t-1}) + \bar{P}_i (1 - u_{i,t}) \quad \forall i, \forall t \quad (5)$$

$$p_{i,t-1} - p_{i,t} \leq RD_i \cdot u_{i,t} - \underline{P}_i (u_{i,t} - u_{i,t-1}) + \bar{P}_i (1 - u_{i,t-1}) \quad \forall i, \forall t \quad (6)$$

$$u_{i,t+\tau-1} \geq u_{i,t} - u_{i,t-1} \quad \forall i, \forall \tau = 1, 2, \dots, T_{i,on}^{\min}, \forall t \quad (7)$$

$$u_{i,t+\tau-1} \leq 1 - u_{i,t-1} + u_{i,t} \quad \forall i, \forall \tau = 1, 2, \dots, T_{i,off}^{\min}, \forall t \quad (8)$$

$$x_{i,t} + y_{i,t} \leq 1 \quad \forall i, \forall t \quad (9)$$

$$u_{i,t+1} - u_{i,t} = x_{i,t} - y_{i,t} \quad \forall i, \forall t \quad (10)$$

$$-P_{l,\max} \leq \sum_k G_{l,n_k} p_{k,t} - \sum_j G_{l,n_j} d_{j,t} \leq P_{l,\max} \quad \forall l, \forall t \quad (11)$$

where N_D is the number of buses; $d_{j,t}$ is the demand of load j during period t ; D_t is the total demand during period t ; $u_{i,t}$ is the unit on-off state indicator binary variable, which is equal to 1 when unit i is on and to 0 when unit i is off during period t ; $\underline{P}_i$ and $\bar{P}_i$ are the minimum and maximum power outputs of unit i , respectively; R_t is the case-specified minimum spinning reserve requirement during period t ; RU_i and RD_i are the ramp up and ramp down limits of unit i , respectively; $T_{i,on}^{\min}$ and $T_{i,off}^{\min}$ are the minimum up and down time of unit i , respectively; $G_{l,n}$ is the power transfer distribution factor of line l and node n ; n_k is the node of generator k ; n_j is the node of load j ; $p_{k,t}$ is the power output of generator k during period t ; $d_{j,t}$ is the demand of load j during period t ; and $P_{l,\max}$ is the maximum power flow limit of line l .

III. FAST INFEASIBILITY ANALYSIS FRAMEWORK FOR NCUC

This paper establishes a fast infeasibility analysis framework for analyzing the infeasibility of NCUC, which analyzes and repairs infeasible NCUC model based on IISs, as shown in Fig. 2, where GC is short for Gurobi computeIIS.

For infeasible MILP problems, general analytical tools include filtering algorithms and EP. In the case of EP, the initial step involves identifying which constraints will be elasticized.

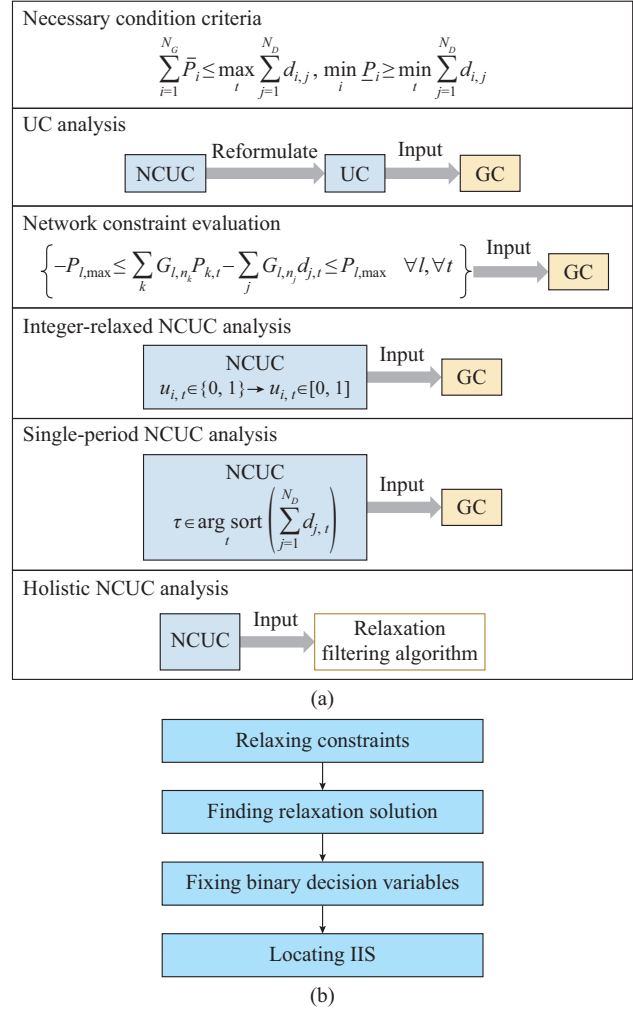


Fig. 2. Overall structures of proposed framework and relaxation filtering algorithm. (a) Proposed framework. (b) Relaxation filtering algorithm.

Subsequently, penalty factors for the slack variables in each elasticized constraint are determined. Finally, by solving the EP problem, an optimal slack-feasible solution is obtained under the prespecified penalty factor parameters, thereby facilitating the analysis and repair of model infeasibility.

The IIS-based method proceeds by directly applying the IIS location algorithm to the entire model to identify the corresponding IIS. Following this, the IIS is manually analyzed, and constraints are relaxed or removed to render the IIS feasible. If the model regains feasibility after IIS repair, the analysis and repair of the infeasible model are completed. Otherwise, the IIS location algorithm is reapplied to the entire model to identify a new IIS, and the process is repeated until the model becomes feasible.

One significant issue with the aforementioned general methods is that the NCUC problem is considered as a whole throughout the analysis process, which is inefficient. Furthermore, the ability of these methods to analyze the problem is relatively limited and uniform. The analysis processes corresponding to these general methods do not fully exploit the unique structural characteristics of different optimization problem models, as they are always considered as a whole.

While the generality of these methods is an advantage, it is also a limitation.

A. Subsystem-based IIS Analysis Method

The proposed framework employs a subsystem-based IIS analysis method shown in Fig. 3 for IIS identification in infeasible NCUC model. By removing constraints from the original model, simplified subsystems are generated. Crucially, the feasible regions of these subsystems strictly contain the feasible region of the original model. This mathematical relationship establishes a fundamental principle: the feasibility of the original model is a necessary condition for the feasibility of its subsystems. The IISs of individual subsystems constitute IISs of the original model.

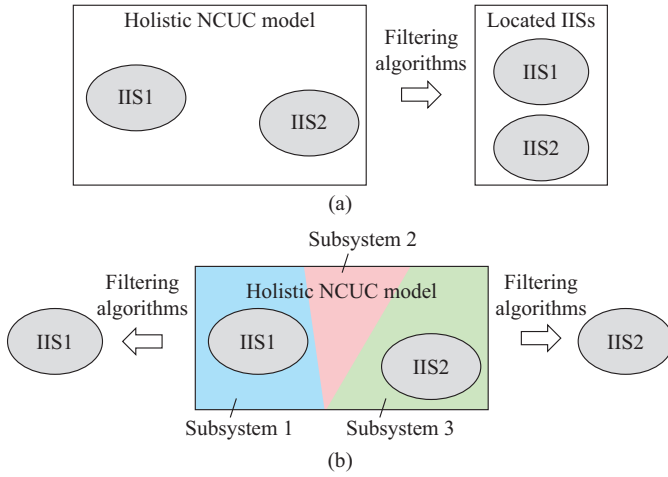


Fig. 3. General analysis method and subsystem-based IIS analysis method. (a) General analysis method. (b) Subsystem-based IIS analysis method.

Furthermore, the effectiveness of the subsystem-based IIS analysis method hinges on two critical factors.

- 1) Constraint screening strategy: how to identify and remove constraints for generating meaningful subsystems.
- 2) Redundancy preservation: how to ensure the simplified models retain essential structural properties of the original problem.

Based on this, three basic principles for subsystem division are proposed, including physical principle, mathematical principle, and data-driven principle.

1) Physical principle: for optimization problems derived from real world with strong physical meanings, their physical structures and explanation to infeasibility can be employed for subsystem division.

2) Mathematical principle: common mathematical relaxation and division methods for optimization can be employed for subsystem division.

3) Data-driven principle: for general optimization lacking knowledge about physical or mathematical structures, historical data involving IISs can be employed to predict subsystems covering more IISs.

In comparison with methods that directly predict IISs, the subsystem-based IIS analysis method can ensure the property of IISs and explore the hidden structure of models.

Considering the mathematical and physical meanings of NCUC, our methodology addresses these challenges through

a hierarchical analysis framework based on physical principle and mathematical principle, which balances computational efficiency with analysis accuracy. In practical scenarios, there are many causes of infeasibility of NCUC problems. To conduct an efficient and comprehensive analysis of infeasible NCUC models, it is necessary to examine the infeasibility of NCUC model from multiple perspectives, considering both its physical and mathematical characteristics. Furthermore, from a mathematical perspective, NCUC involves a significant number of integer variables, allowing for analysis of the integer-relaxed model. An important physical characteristic of NCUC is the coupling of units and time periods, which can be considered in decoupling model analysis. The models derived from relaxing the NCUC constraints are utilized for these analyses. The infeasibility analysis and repair conducted on these relaxed models can assist in resolving the infeasibility of the original NCUC problem. Based on this observation, the proposed framework considers several simplified NCUC models rather than only holistic NCUC model and conducts analysis from multiple perspectives. Given the exponentially increasing complexity of NCUC problems, such a decomposition strategy proves particularly effective. Reducing the problem scale significantly decreases the computational time required for IIS identification. In many cases, the IIS identified within a subsystem is identical to that derived from the holistic model, yet the former achieves this with substantially lower computational time.

B. Analysis Process of Proposed Framework

The analysis process of the proposed framework follows the basic pattern of the general process shown in Fig. 1, including the iterations of the identification and repair of infeasible key constraints, which is generally an IIS for the proposed framework.

1) Identification of Infeasible Key Constraints

Overall, the proposed framework is segmented into several major submodules: necessary condition criteria determination, network constraint evaluation, single-period NCUC model analysis, integer-relaxed NCUC model analysis, UC model analysis, and holistic NCUC model analysis, as shown in Fig. 4.

Among them, the necessary condition criteria determination is used in the highest-priority analysis method, primarily for examining the feasibility of NCUC model through efficient operations such as checking the satisfaction of inequalities like (12) and (13).

Formula (12) ensures the total available generation capacity $\sum_{i=1}^{N_G} \bar{P}_i$ meets the peak demand of the system $\max_t \sum_{j=1}^{N_D} d_{j,t}$, acting as a hard lower bound for feasibility. Formula (13) guarantees that the minimum generation capability $\min_i \bar{P}_i$ exceeds the minimum demand of the system $\min_t \sum_{j=1}^{N_D} d_{j,t}$.

$$\sum_{i=1}^{N_G} \bar{P}_i \leq \max_t \sum_{j=1}^{N_D} d_{j,t} \quad (12)$$

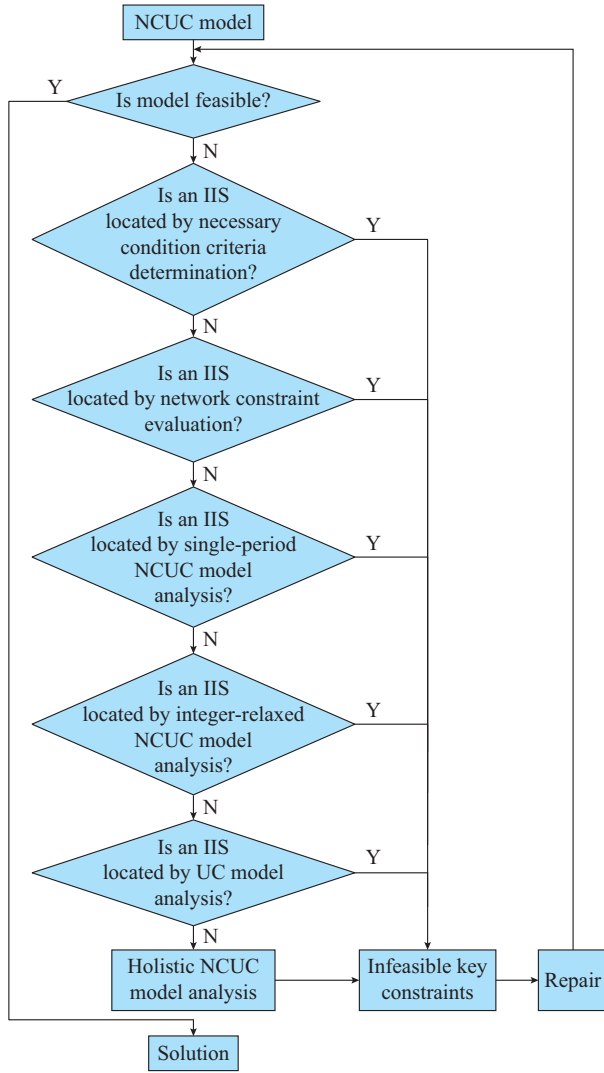


Fig. 4. Workflow of proposed framework for NCUC.

$$\min_i P_i \geq \min_t \sum_{j=1}^{N_D} d_{j,t} \quad (13)$$

Then, a series of operations are performed to simplify the original NCUC model, and the IIS is identified within the simplified models. The proposed framework includes four types of simplified models, prioritized in order of analysis: a UC model (2)-(10), a network feasibility model (11), an integer-relaxed NCUC model that relaxes the integer decision variables, and a single-period NCUC model (14) with a transition period τ from peak load to valley load conditions.

$$\min \sum_{i=1}^{N_G} f(p_{i,\tau}) \quad (14a)$$

s.t.

$$\sum_{i=1}^{N_G} p_{i,\tau} = \sum_{j=1}^{N_D} d_{j,\tau} = D_\tau \quad (14b)$$

$$P_i u_{i,\tau} \leq p_{i,\tau} \leq \bar{P}_i u_{i,\tau} \quad \forall i \quad (14c)$$

$$\sum_{i=1}^{N_G} u_{i,\tau} (\bar{P}_i - p_{i,\tau}) \leq R_\tau \quad (14d)$$

$$-P_{l,\max} \leq \sum_k G_{l,n_k} p_{k,\tau} - \sum_j G_{l,n_j} d_{j,\tau} \leq P_{l,\max} \quad \forall l \quad (14e)$$

$$\tau \in \arg \text{sort} (D_l) \quad (14f)$$

where $\arg \text{sort}(\cdot)$ is the function that returns the indices t that would sort D_l .

The process priorities of the proposed framework are established according to two fundamental principles. First, model simplicity takes precedence, implying that subsystems with minimal complexity are prioritized in the analysis sequence. Second, specialized models are given higher priority compared to general-purpose models. This ensures early detection of low-complexity conflicts before progressing to more intricate models. Based on these principles, the network constraint evaluation and single-period NCUC model analysis are designated as first-order subsystems for analysis, with the former taking precedence due to its LP structure, smaller problem scale, and superior computational efficiency. In comparison to the UC model, the integer-relaxed NCUC model features more linear constraints ($2N_L N_T$) but fewer integer constraints ($3N_G N_T$). Although $2N_L N_T > 3N_G N_T$ typically holds for NCUC problems, the integer-relaxed NCUC model is analyzed following the single-period NCUC model analysis due to the higher computational complexity associated with integer constraints. Finally, the UC model is analyzed last in the sequence.

By analyzing these simplified models via various algorithms, infeasible key constraints can be identified more efficiently and precisely. Most of these simplified models are LP problems or simple enough, which means that commercial solvers can be directly applied to get an IIS effectively.

If all simplified model analyses fail to identify the cause of infeasibility, the relaxation filtering algorithm proposed in Section IV is applied to locate the IIS within the holistic NCUC model, which can be computationally intensive in terms of time and space. However, the relaxation filtering algorithm offers significant advantages in speed and accuracy in comparison with general IIS analysis methods embedded in commercial solvers.

2) Repair of Infeasible Key Constraints and Iterations

After it is located, the infeasible key constraint set, which is usually an IIS, is repaired by removing or adjusting the constraints to restore feasibility. For the former method, removing constraints may affect the accuracy of NCUC modeling. For the latter, the adjustment is usually performed through EP.

Considering an infeasible key constraint set such as $\mathbf{a}_i^T \mathbf{x} \leq b_i, \forall i$, an elastic program (15) is solved to obtain the adjusting scheme, where $\mathbf{a}_i$ is the coefficient of the i^{th} constraint; b_i is the right-hand-side value; $s_i \geq 0$ is the slack variable introduced to relax the i^{th} constraint; and w_i is the corresponding penalty weight. The solver is then reapplied to the repaired model. If the model is feasible, the proposed framework has successfully analyzed and repaired the model; otherwise, the model will be analyzed repeatedly following the introduced steps.

$$\min \sum w_i s_i \quad (15a)$$

s.t.

$$a_i^T x \leq b_i + s_i \quad \forall i \quad (15b)$$

IV. RELAXATION FILTERING ALGORITHM FOR HOLISTIC NCUC MODEL ANALYSIS

To improve the efficiency of IIS identification in the holistic NCUC model, a relaxation filtering algorithm inspired by dimensionality reduction techniques in optimization acceleration frameworks is proposed.

Model simplification through unit/period clustering can reduce the number of binary variables and redundant constraints, improving NCUC optimization efficiency. These strategies are extended to IIS location to address the core challenge of long time consumption in filtering caused by the combinatorial complexity of the MILP formulation of NCUC.

The key of the relaxation filtering algorithm is to reduce the number of binary variables and redundant constraints to locate IISs more quickly. This algorithm is a combination of preprocessing algorithms and general filtering algorithms. The following steps are included in the relaxation filtering algorithm.

- 1) Exclude network constraints in the NCUC to obtain a relaxed NCUC.
- 2) Find a relaxation solution for the relaxed NCUC.
- 3) Replace the binary variables with the values of the solution found in the relaxed NCUC, obtaining a fixed NCUC.
- 4) Locate IISs in the fixed NCUC.

Each step is described below.

A. Obtaining Relaxed NCUC

The relaxation filtering algorithm first applies constraint relaxation to achieve preliminary model repair, enabling the derivation of a preliminary approximation for the repaired NCUC model.

To ensure the quality of the solution, the network constraints are removed to form a UC model, which can guarantee the availability of a feasible solution in most cases. In cases where this relaxation proves insufficient to achieve feasibility, further consideration may be given to removing and/or relaxing other constraints such as unit ramping rate constraints. In this study, the relaxation filtering algorithm is applied within the proposed framework for holistic NCUC model analysis, with UC model analysis ensuring UC model feasibility, thereby validating the effectiveness of the relaxation filtering algorithm in feasibility assurance from the given conditions.

The relaxed NCUC is obtained after constraint relaxation, which is an MILP problem.

B. Finding Relaxation Solution

Following the formulation of the relaxed NCUC, a commercial optimization solver (e.g., Gurobi or Cplex) is employed to compute a relaxation solution $\{x_r^*, y_r^*\}$, where x_r^* denotes the vector of values for all continuous variables; and y_r^* denotes the vector of values for all binary variables within the relaxed NCUC.

This relaxation solution serves as a proximate solution for

the repaired NCUC formulation. By leveraging this proximate solution as an initial approximation, the original NCUC problem can be systematically repaired.

This process outperforms solving model with slack variables in terms of efficiency, despite their similarities in relaxing constraints. The former focuses only on finding an approximate solution of the reduced model, which is used to fix the binary decision variables. The latter searches for a precise optimal solution to repair the model, which means that the former is computationally faster than the latter.

C. Obtaining Fixed NCUC

The vector of binary decision variables y in the original NCUC model is substituted with the relaxation solution y_r^* , which is achieved by adding binary variable fixing constraints $y = y_r^*$. The resulting model after this substitution process is called the fixed NCUC, which is expressed as:

$$\begin{cases} Ax + By \leq b \\ y = y_r^* \\ x \geq 0 \end{cases} \quad (16)$$

where A and B are the coefficient matrices corresponding to continuous and binary variables, respectively; and b is the vector of constant parameters.

Notably, unlike the original NCUC and UC, the fixed NCUC is an LP problem, which means that it can be handled efficiently by general filtering algorithms.

D. Locating IIS

Finally, filtering algorithms implemented in commercial solvers are applied to the fixed NCUC to locate the IIS. If the detected IIS includes binary variable fixing constraints $y = y_r^*$, these constraints are systematically removed from the fixed NCUC problem, followed by re-identification of the IIS in the revised formulation. This iterative process continues until an IIS free of binary variable fixing constraints is obtained. Notably, such an IIS is guaranteed to correspond to an IIS of the primal NCUC model, for its constraints are all from the primal NCUC model. The IIS contributes to the core conflict in the original model. By leveraging the identified IIS, the original NCUC formulation can be partially repaired through targeted constraint relaxation.

E. Overview and Time Analysis of Relaxation Filtering Algorithm

The basic idea and process of the relaxation filtering algorithm are demonstrated through a set of algorithmic principle diagrams, as shown in Fig. 5. Considering the original NCUC shown in part (1) of Fig. 5, the model has no feasible solution for some inconsistent constraint subsets. By relaxing some of the constraints, a relaxed NCUC is obtained, as shown in part (2) of Fig. 5, in which all four constraints A, B, C, and D in the model have undergone relaxation operations. There is a feasible solution for the relaxed NCUC. By solving this relaxed NCUC, a relaxation solution is obtained, as shown in part (3) of Fig. 5. Based on this relaxation solution, the integer variables in the original NCUC are replaced, yielding a corresponding fixed NCUC, as shown in part (4) of Fig. 5. In the fixed NCUC, the general IIS location algorithm is employed to find the IIS, as shown

in part (5) of Fig. 5, where the IIS is identified as the set of constraints B, C, D. Finally, by repairing based on the located IIS, the feasibility of the original NCUC can be restored, as illustrated in part (6) of Fig. 5.

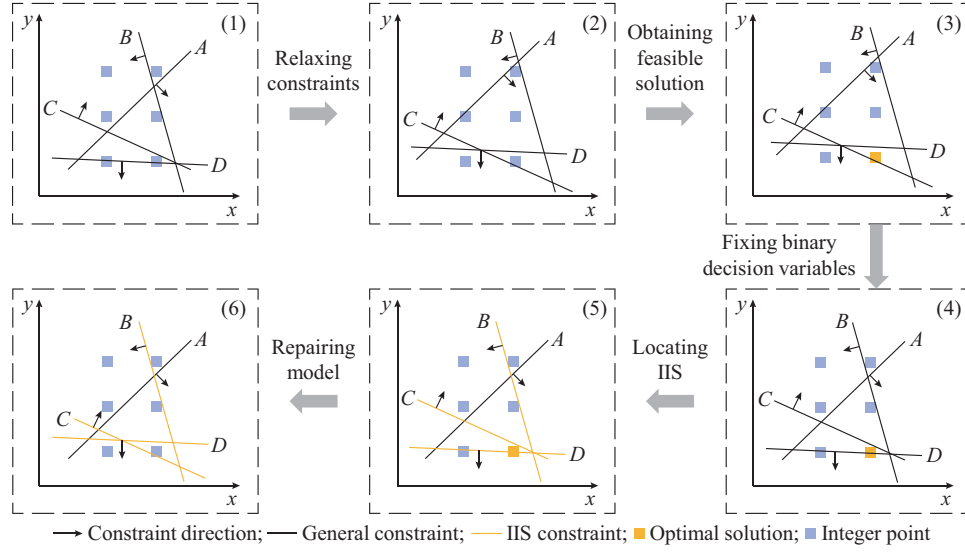


Fig. 5. Basic idea and process of relaxation filtering algorithm.

Based on the approximate UC solution, the relaxation filtering algorithm reduces binary variables in the primal model, thereby significantly increasing the efficiency of locating the IIS.

The computationally intensive components of the relaxation filtering algorithm primarily involve two aspects: optimization of the relaxation solution and IIS identification within the fixed NCUC. While the optimization of the UC formulation is generally effective in generating the relaxation solution, further improvements in computational efficiency can be achieved by integrating UC solution acceleration algorithms. This integration reduces the time required to obtain the relaxation solution, thereby enhancing the overall performance of the relaxation filtering algorithm.

For the IIS identification component, the fixed NCUC typically allows iteration-free identification of the IIS in most problem instances. The IIS location algorithm for LP problems exhibits high computational efficiency, with theoretical complexity analysis indicating that it requires solving an LP problem whose scale is comparable to that of the dual problem of the original LP model. In scenarios where the initially identified IIS includes binary variable fixing constraints, a revised fixed NCUC formulation is generated. While this process introduces additional binary variables into the model, the cardinality of binary variables remains significantly lower than that in the primal NCUC formulation. Consequently, the IIS identification procedure in the revised fixed NCUC retains its computational advantages, enabling faster IIS location compared to the original problem.

V. CASE STUDIES

The effectiveness of the proposed framework and algorithm is evaluated individually in a modified IEEE 118-bus system, a South-Carolina 500-bus system, and a European 1354-bus system, which are abbreviated as IEEE118, ACTIVSg500, and PEGASE1354, respectively. Detailed infor-

mation about these cases are shown in Table II. The interval of the time slots is set to be 15 min. The solver is Gurobi 11.0.0. The gap parameter of the mixed-integer programming (MIP) is set to be 0.1% and the solution time limit is set to be 20000 s. All the computations are performed on a PC with 192 GB of RAM and an Intel Core i9-14900KS CPU @ 3.20 GHz.

TABLE II
DETAILED INFORMATION OF DIFFERENT CASES

Case	Number of buses	Number of branches	Number of units
IEEE118	118	186	53
ACTIVSg500	410	500	90
PEGASE1354	1354	1991	193

To obtain sufficient infeasible models for analysis, the peak rates of the feasible cases are modified to obtain infeasible models. The peak rate p is the rate of the peak load and the sum of the maximum outputs of the units, expressed as (17). The detailed modification method is described in (18). The targeted peak rate p_s is first set. Then, the ratio of the original peak rate p_o to the targeted peak rate p_s is calculated. Finally, all the load data $d_{i,t}$ are multiplied with the calculated ratio to obtain the targeted load data $d'_{i,t}$.

$$p = \frac{\max_t \sum_i d_{i,t}}{\sum_i \bar{P}_i} = \frac{\max_t D_t}{\sum_i \bar{P}_i} \quad (17)$$

$$d'_{i,t} = d_{i,t} \frac{p_s}{p_o} = d_{i,t} \frac{\max_t D_t^s}{\max_t D_t^o} = d_{i,t} \frac{\max_t D_t^s}{\sum_i \bar{P}_i} \quad (18)$$

where D_t^o is the total demand during period t in the original case; and D_t^s is the total demand during period t in the targeted case.

The constraints in NCUC models exhibit complex coupling. An increasing peak rate can directly cause infeasibility in single-period constraints and inherently triggers intricate temporal interdependencies across the entire model. As the peak rate increases, the system load and the unit output increase, causing conflicts in the constraints such as congestion in lines or power imbalance. The increased system load can yield substantially different UC schedules, potentially violating unit minimum on/off time constraints. Additionally, the rate of change of the system load also increases as the peak rate increases, which can lead to lack of ramping ability.

A comparison of the analysis and repair efficiency and accuracy for infeasible NCUC models between the proposed framework and other methods is conducted first, followed by the ablation study of the proposed framework. Finally, a comparison of the proposed algorithm and the IIS location algorithm provided by the commercial solver is conducted.

A. Comparison of Analysis Efficiency for Infeasible NCUC Models

To conduct a comparative analysis with a focus on the efficiency of identifying the IIS and the efficiency of repairing infeasible NCUC models, we compare the proposed framework with the following two representative methods. Notably, conventional filtering methods, including deletion and addition filters, are seldom used for domain-specific, large-scale NCUC problems due to limited efficacy.

1) EP: slack variables are incorporated into the network constraints of the NCUC model to facilitate the solution process [8].

2) GC: the computeIIS function is provided by Gurobi for IIS location. All the related parameters are the default values.

A comparison of the analysis efficiency for infeasible NCUC models in different cases is listed in Table III. EP fails to perform the analysis in a reasonable amount of time in two cases. GC and the proposed framework analyze all three cases successfully, while the computational time of GC is at least 10 times longer than that of the proposed framework. As is clear from Table III, the efficiency of the proposed framework is the highest, at least one order of magnitude greater than that of other two methods.

TABLE III

COMPARISON OF ANALYSIS EFFICIENCY FOR INFEASIBLE NCUC MODELS IN DIFFERENT CASES

Case	Time (s)		
	EP	GC	Proposed framework
IEEE118	>20000.00	248.73	28.65
ACTIVSg500	7526.18	980.45	81.81
PEGASE1354	>30000.00	2330.66	374.00

B. Comparison of Repair Efficiency for Infeasible NCUC Models

Tests are also conducted to evaluate the repair efficiency

of different methods. EP is considered a proper repair method, which means that solving the EP model achieves analysis and repair of infeasibility simultaneously.

For GC and the proposed framework, analysis and repair are carried out iteratively, and different repair methods are applied to the IIS located in each iteration. The repair methods for the involved IIS include removing constraints directly and adjusting constraints by solving a small EP problem.

The comparison of repair efficiency for infeasible NCUC models in different cases is listed in Tables IV and V.

EP fails to repair the IEEE118, but takes a reasonable time in the ACTIVSg500, which is the same with that in Table III, as EP achieves analysis and repair of infeasibility simultaneously. For GC and the proposed framework, the required time varies with the repair method used. Adjusting the constraints costs much more time than just removing the constraints. GC performs well in the IEEE118, whereas it fails to repair the ACTIVSg500 when adjusting constraints is used as the repair method. The proposed framework successfully repairs all the cases with both repair methods in a reasonable time. The repair efficiency of the proposed framework remains the highest, achieving at least one order of magnitude improvement in repair efficiency compared with the EP and GC. This finding indicates that the proposed framework has a significant advantage in terms of repairing infeasible NCUC models.

TABLE IV
COMPARISON OF REPAIR EFFICIENCY FOR INFEASIBLE NCUC MODELS IN IEEE118

Repair method	Time (s)		
	EP	GC	Proposed framework
Removing constraints	>20000	1909.62	50.35
Adjusting constraints	>20000	6150.21	73.38

TABLE V
COMPARISON OF REPAIR EFFICIENCY FOR INFEASIBLE NCUC MODELS IN ACTIVSg500

Repair method	Time (s)		
	EP	GC	Proposed framework
Removing constraints	7526.18	17213.85	267.02
Adjusting constraints	7526.18	>20000.00	1053.68

To better investigate the difference between the proposed framework and GC, the detailed information on each iteration in repairing infeasible NCUC models is compared. In the IEEE118, the proposed framework and GC both use 2 iterations. A comparison of the analysis efficiency per single iteration for infeasible NCUC models in the IEEE118 is listed in Table VI. In the ACTIVSg500, the proposed framework uses 6 iterations, whereas GC fails to repair the model in 62 iterations. A comparison of the analysis efficiency of the 5 most time-consuming iterations for infeasible NCUC models in the ACTIVSg500 is listed in Table VII.

The efficiency of the proposed framework is higher than that of GC when they are compared in a single iteration.

TABLE VI
COMPARISON OF ANALYSIS EFFICIENCY PER ITERATION FOR INFEASIBLE NCUC MODELS IN IEEE118

Number of iterations	Time (s)		IIS size	
	GC	Proposed framework	GC	Proposed framework
1	248.40	30.09	5-1	3-0
2	5276.05	43.29	39-3	42-5

Note: m - n represents that there are m linear constraints and n bound constraints in the IIS, which also applies to the following tables.

TABLE VII
COMPARISON OF ANALYSIS EFFICIENCY OF 5 MOST TIME-CONSUMING ITERATIONS FOR INFEASIBLE NCUC MODELS IN ACTIVSG500

Number of iterations		Time (s)		IIS size	
GC	Proposed framework	GC	Proposed framework	GC	Proposed framework
62	2	82264.55	643.45	Fail to analysis	3-0
61	1	8857.42	134.52	32-18	15-0
2	4	3492.68	105.99	26-12	39-15
1	5	1389.80	95.17	3-0	29-2
60	3	171.91	74.54	15-50	39-11

The IISs located by the proposed framework do not exhibit significant advantages in terms of size in comparison with those located by GC, yet they remain reasonable and acceptable.

C. Comparison of Repair Accuracy for Infeasible NCUC Models

It is crucial to evaluate the accuracy of repairing infeasible NCUC models. Evaluating such accuracy, however, poses a challenge since an infeasible NCUC model lacks a feasible solution until it undergoes repair. Consequently, we regard EP outcomes as ideal benchmarks for repair, as they concurrently account for both the primary objectives and the penalty costs associated with constraint violations. Nevertheless, determining appropriate penalty coefficients for slack variables presents a significant challenge, as previously noted. The question of how to set these coefficients to accurately represent the true economic costs incurred by constraint violations requires further research. Therefore, the comparative analysis presented in this subsection should be viewed as preliminary and indicative rather than definitive, where we introduce a large penalty coefficient M for the slack variables in the objective function to minimize constraint modifications. This penalty term is then combined with the original objective of the NCUC model for optimization. Based on the solution, the values of the slack variables are fixed, and the model is re-optimized to obtain the final results.

The comparison of repair accuracy for infeasible NCUC models in the ACTIVSG500 is listed in Table VIII.

When removing constraints is used as the repair method, both GC and the proposed framework attain a substantially lower objective value compared to EP. This suggests that while simply removing constraints achieves high computational efficiency, this method remains impractical due to its significant error.

TABLE VIII
COMPARISON OF REPAIR ACCURACY FOR INFEASIBLE NCUC MODELS IN ACTIVSG500

Repair method	Objective value of EP	Relative objective value of GC to EP	Relative objective value of proposed framework to EP
Removing constraints	1.000	0.622	0.989
Adjusting constraints	1.000	Fail to repair	0.995

In contrast, when adjusting constraints is used as the repair method, the proposed framework achieves a reasonable objective value close to that of EP, demonstrating the accuracy of the proposed framework. As previously stated, GC fails to repair the model within 20000 s when constraint adjustment is used as the repair method, and thus its accuracy is not compared here. In conclusion, considering repair accuracy, adjusting constraints is the appropriate repair method for real-world applications, and the proposed framework achieves high repair accuracy in this context.

D. Ablation Study of Proposed Framework

In this subsection, we investigate the effects of the major submodules and the relaxation filtering algorithm within the proposed framework by systematically removing and evaluating each submodule and the relaxation filtering algorithm. These submodules are skipped for removal. For the relaxation filtering algorithm, it is replaced with the computeIIS function of Gurobi for removal. The experiment is conducted in the ACTIVSG500, with adjusting constraints chosen as the repair method. The complete framework serves as the baseline for this ablation study. The results are listed in Table IX.

TABLE IX
ABLATION STUDY OF PROPOSED FRAMEWORK

Removal	Relative analysis time (%)	Relative objective value (%)
Necessary condition criteria	100.00	100.00
Network constraint evaluation	124.58	100.04
Single-period NCUC model analysis	233.94	100.00
Integer-relaxed NCUC model analysis	170.32	100.00
UC model analysis	140.49	100.00
Relaxation filtering algorithm and single-period NCUC model analysis	298.14	100.00

Except for the necessary condition criteria, the removal of other submodules results in a significant increase in the analysis time. These removed submodules play a crucial role in improving the efficiency of analyzing infeasible NCUC models. However, the necessary condition criteria does not function as intended because it is specifically designed to identify severely infeasible models, whereas the ACTIVSG500 represents a complex case that causes the necessary condition criteria to fail.

The performance of the relaxation filtering algorithm is rather modest in comparison to other submodule removal, which highlights the significance for avoiding holistic infeasibility analysis of large-scale optimization models. Regarding analysis accuracy, the submodules all have minimal impact on the objective of repairing infeasible NCUC models. This implies that compared to performing a comprehensive NCUC infeasibility analysis, the proposed framework significantly enhances the efficiency of the analysis process while maintaining a high level of accuracy.

E. Test of Relaxation Filtering Algorithm

In this subsection, the efficiency of the relaxation filtering

algorithm (algorithm 1) is tested and compared with the computeIIS function provided by Gurobi (algorithm 2).

The test results are shown in Table X. Algorithm 1 demonstrates a notable efficiency advantage over algorithm 2 even without incorporating acceleration algorithms. Specifically, compared to algorithm 2, algorithm 1 reduces the computation time by 79.22% in Scenario 1 and 86.85% in Scenario 2 (with higher peak rate) for the IEEE118, achieves a 68.40% reduction for the ACTIVSg500, and achieves a 68.77% reduction for the PEGASE1354. This underscores the significant acceleration achieved by algorithm 1 across different cases.

TABLE X
COMPUTATION TIME AND IIS SIZE OF DIFFERENT ALGORITHMS

Case	Computation time (s)		Computation time reduced by algorithm 1 compared to algorithm 2 (%)	IIS size	
	Algorithm 1	Algorithm 2		Algorithm 1	Algorithm 2
IEEE118 (Scenario 1)	51.37	247.26	79.22	8-3	5-1
IEEE118 (Scenario 2)	57.56	437.62	86.85	21-32	3-0
ACTIVSg500	306.08	968.69	68.40	25-50	3-0
PEGASE1354	732.98	2347.20	68.77	4-0	4-0

Regarding the scale of the located IIS, the scale of the IIS located by algorithm 1 remains small and within an acceptable range, indicating its effectiveness in IIS location.

VI. CONCLUSION

A fast infeasibility analysis framework for NCUC and a relaxation filtering algorithm are proposed in the paper. The framework can perform efficient and precise pre-solution infeasibility analysis based on the subsystem-based IIS analysis method. Taking the mathematical and physical characteristics of NCUC into consideration, several simplified models of the infeasible NCUC model are generated. By analyzing the simplified models, the speed of IIS location is significantly improved. For the relaxation filtering algorithm, based on the relaxation solution, the holistic NCUC model is transformed and filtered to locate the IIS, significantly improving the efficiency of infeasibility analysis. The case studies show that the proposed framework and algorithm can analyze and repair infeasible NCUC models effectively, reducing the analytical time by at least one order of magnitude over that of traditional methods.

The proposed framework and algorithm address deterministic NCUC under fixed forecasts but require extension to handle stochastic uncertainties from renewable energy sources [32], [33]. Future studies should focus on infeasibility diagnosis for stochastic optimization in power systems.

Though the relaxation filtering algorithm is theoretically applicable to all MILP problems, its performance highly depends on the quality of the approximation solution, which may limit its further application. The relationship between solution quality and IIS identification requires deeper investigation. For the proposed framework, the subsystem-based IIS analysis method is applicable to other MILP problems in which the constraint set can be decoupled based on physical knowledge or mathematical principle. Nonetheless, the pro-

posed framework can still be improved by introducing more precise subsystem division methods based on advanced knowledge. Moreover, the causes of infeasibility can be different across cases while current framework is sequential and determined. Therefore, future work can focus on generating heterogeneous subsystems and developing dynamic, context-aware selection mechanisms tailored to case-specific characteristics. Furthermore, data-driven methods can be implemented to uncover abstract, yet critical constraint interaction patterns.

Additionally, current infeasibility repair methods rely either on introducing full slack variables (EP-based methods) or iteratively adding local slack variables (IIS-based methods), yet little research has focused on accelerating the optimization of relaxed models or optimizing penalty coefficients for slack variables. This gap underscores the urgent need for more efficient, adaptive, and computationally scalable repair mechanisms.

REFERENCES

- [1] Y. Chen, F. Pan, F. Qiu *et al.*, "Security-constrained unit commitment for electricity market: modeling, solution methods, and future challenges," *IEEE Transactions on Power Systems*, vol. 37, no. 2, pp. 1119-1132, Mar. 2022.
- [2] Z. Ma, H. Zhong, T. Cheng *et al.*, "Redundant and nonbinding transmission constraints identification method combining physical and economic insights of unit commitment," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 3487-3495, Sept. 2021.
- [3] PJM Interconnection. (2022, Mar.). PJM manual 11: energy & ancillary services market operations. [Online]. Available: <https://www.pjm.com>
- [4] South China Energy Regulatory Bureau of National Energy Administration of the People's Republic of China. (2022, Jan.). Detailed rules for the implementation of spot electricity energy trading in the Guangdong electricity market (trial version 2022). [Online]. Available: <https://nfdj.nea.gov.cn/nfdjdoc/aa9c1ba4-c055-4a6a-8b33-72e5203d42bf.pdf>
- [5] Shanxi Power Exchange Center. (2022, Jun.). Compilation of rules for the Shanxi electricity market (pilot operation V12.0). [Online]. Available: <https://pmos.sx.sgcc.com.cn/pxf-settlement-outnetpub/#/pxf-settle->

- ment-outnetpub/unstructuredInfo? flag=1&fileId=nfad9d33d5b2244c0b9455fdf95bdcda9
- [6] 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*, vol. 23, no. 2, pp. 344-352, May 2008.
 - [7] C. He, Q. Chen, H. Zhong *et al.*, "A homogenized-overload model applied for infeasible security-constrained unit commitment (SCUC) problem," in *Proceedings of 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering*, Halifax, Canada, May 2015, pp. 242-247.
 - [8] L. D. Secchin, G. M. Ramalho, C. Sagastizábal *et al.* (2022, Jan.). Analysis of infeasible unit-commitment solutions arising in energy optimization. [Online]. Available: <https://arxiv.org/abs/2201.07163>
 - [9] Y. Sahraoui, P. Bendotti, and C. D'Ambrosio, "Real-world hydro-power unit-commitment: dealing with numerical errors and feasibility issues," *Energy*, vol. 184, pp. 91-104, Oct. 2019.
 - [10] Z. Yang, G. Tang, and L. Li, "Application of relaxed constraints model on generation schedule optimization," *Automation of Electric Power Systems*, vol. 34, no. 14, pp. 53-57, Jul. 2010.
 - [11] N. Chakravarti, "Some results concerning post-infeasibility analysis," *European Journal of Operational Research*, vol. 73, no. 1, pp. 139-143, Feb. 1994.
 - [12] Y. Sahraoui, "Short-term hydropower production scheduling: feasibility and modeling," Ph.D. dissertation, Université Paris Saclay, Paris, France, 2016.
 - [13] J. W. Chinneck, "Finding a useful subset of constraints for analysis in an infeasible linear program," *INFORMS Journal on Computing*, vol. 9, no. 2, pp. 164-174, May 1997.
 - [14] E. Amaldi, M. E. Pfetsch, and L. E. Trotter, "On the maximum feasible subsystem problem, IISs and IIS-hypergraphs," *Mathematical Programming*, vol. 95, no. 3, pp. 533-554, Mar. 2003.
 - [15] M. Parker and J. Ryan, "Finding the minimum weight IIS cover of an infeasible system of linear inequalities," *Annals of Mathematics and Artificial Intelligence*, vol. 17, no. 1-2, pp. 107-126, Mar. 1996.
 - [16] J. W. Chinneck and E. W. Dravnieks, "Locating minimal infeasible constraint sets in linear programs," *ORSA Journal on Computing*, vol. 3, no. 2, pp. 157-168, May 1991.
 - [17] O. Guieu and J. W. Chinneck, "Analyzing infeasible mixed-integer and integer linear programs," *INFORMS Journal on Computing*, vol. 11, no. 1, pp. 63-77, Feb. 1999.
 - [18] M. K. Atlihan and L. Schrage, "Generalized filtering algorithms for infeasibility analysis," *Computers & Operations Research*, vol. 35, no. 5, pp. 1446-1464, May 2008.
 - [19] Y. Puranik and N. V. Sahinidis, "Deletion presolve for accelerating infeasibility diagnosis in optimization models," *INFORMS Journal on Computing*, vol. 29, no. 4, pp. 754-766, Nov. 2017.
 - [20] Gurobi Optimization. (2023, Jul.). Gurobi optimizer reference manual. [Online]. Available: <https://www.gurobi.com/documentation/>
 - [21] T. Achterberg, "Conflict analysis in mixed integer programming," *Discrete Optimization*, vol. 4, no. 1, pp. 4-20, Mar. 2007.
 - [22] Y. Guan, K. Pan, and K. Zhou, "Polynomial time algorithms and extended formulations for unit commitment problems," *IIEE Transactions*, vol. 50, no. 8, pp. 735-751, Aug. 2018.
 - [23] T. Achterberg, R. E. Bixby, Z. Gu *et al.*, "Presolve reductions in mixed integer programming," *INFORMS Journal on Computing*, vol. 32, no. 2, pp. 473-506, Apr. 2020.
 - [24] X. Guan, S. Guo, and Q. Zhai, "The conditions for obtaining feasible solutions to security-constrained unit commitment problems," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1746-1756, Nov. 2005.
 - [25] M. Zhang, Z. Yang, W. Lin *et al.*, "Enhancing economics of power systems through fast unit commitment with high time resolution," *Applied Energy*, vol. 281, p. 116051, Jan. 2021.
 - [26] L. Wei, X. Ai, J. Fang *et al.*, "Data-augmentation acceleration framework by graph neural network for near-optimal unit commitment," *Applied Energy*, vol. 377, p. 124332, Jan. 2025.
 - [27] M. Zhang, Y. Zhou, M. Xiang *et al.*, "A methodology for building generation trajectories to balance continuous-time load profiles," *IEEE Transactions on Power Systems*, vol. 40, no. 2, pp. 1609-1622, Mar. 2025.
 - [28] L. Tziiovani, L. Hadjidemetriou, and S. Timotheou, "Successive convexification algorithms for optimizing power systems with energy storage models," *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 1807-1820, Mar. 2024.
 - [29] J. T. Holzer, Y. Chen, Z. Wu *et al.*, "Fast simultaneous feasibility test for security constrained unit commitment," *IEEE Transactions on Power Systems*, vol. 38, no. 4, pp. 3178-3187, Jul. 2023.
 - [30] A. R. Sayed, X. Zhang, G. Wang *et al.*, "Deep reinforcement learning-assisted convex programming for AC unit commitment and its variants," *IEEE Transactions on Power Systems*, vol. 39, no. 4, pp. 5561-5574, Jul. 2024.
 - [31] A. V. Ramesh and X. Li, "Feasibility layer aided machine learning approach for day-ahead operations," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1582-1593, Jan. 2024.
 - [32] X. Liu and A. J. Conejo, "Quasi-deterministic proxy for network-constrained stochastic unit commitment," *Journal of Modern Power Systems and Clean Energy*, vol. 13, no. 4, pp. 1167-1175, Jul. 2025.
 - [33] D. Xu, Z. Cai, Q. Cheng *et al.*, "Security-constrained transmission maintenance optimization considering generation and operational risk costs," *Journal of Modern Power Systems and Clean Energy*, vol. 12, no. 3, pp. 767-781, May 2024.
- Yanhong Liu** received the B.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2024, where he is currently working toward the Ph.D. degree in electrical engineering. His research interests focus on power system operation.
- Xinfei Yan** received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2019 and 2025, respectively. He is currently a Post-doctor in Tsinghua University. His research interests include distributed optimization algorithm and power system operation.
- Haiwang Zhong** received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2008 and 2013, respectively. He is currently an Associate Professor with the Department of Electrical Engineering, Tsinghua University. He is also the Director of Energy Internet Trading and Operation Research Department of Sichuan Energy Internet Research Institute, Tsinghua University, Chengdu, China. He was a recipient of the ProSPER.Net Young Scientist Award and the Outstanding Postdoctoral Fellow of Tsinghua University. He currently serves as the Convenor of CIGRE C1.54 Working Group on assessment of system reserves and flexibility needs in the power systems of the future. His research interests include power system operation and optimization and electricity market.
- Chongqing Kang** received the Ph.D. degree from the Department of Electrical Engineering, Tsinghua University, Beijing, China, in 1997. He is currently a Professor with Tsinghua University. His research interests include load forecasting, electricity market, power system planning, and generation scheduling optimization.