

# A SAT-Based Exact Approach for the Minimum Order Frequency Assignment Problem

Nghia Dao Xuan , Phuong Dang Anh , and Khanh To Van 

**Abstract**—The Minimum Order Frequency Assignment Problem (MO-FAP) is a fundamental NP-hard problem in spectrum management, where the objective is to minimize the number of assigned frequencies while satisfying interference constraints between transmitters. Current metaheuristic methods typically do not provide optimality certificates, while traditional exact methods can struggle on heavily constrained instances. This study proposes an exact framework based on Boolean Satisfiability (SAT) for solving MO-FAP. The key idea is to use standard order variables to reformulate pairwise interference constraints as interval-boundary constraints in a proposed encoding called POSE, thereby reducing the number of explicit conflict clauses relative to a direct SAT encoding. In addition, the framework includes an incremental SAT solving strategy (INCSC) based on Sequential Counter encoding, where the clause construction is designed to support incremental solving and clause reuse across iterations without rebuilding the formula. Experiments on 35 CALMA and DUTest benchmark instances from the FAP database show that the proposed approach certifies optimality or infeasibility for every tested instance within a 600-second time limit. On the tested benchmark set, it certifies more instances than the considered exact baselines and reduces solving time on several instances where the baselines reach the time limit. These results show that the proposed SAT-based framework provides exact certification with competitive runtime on the tested MO-FAP benchmark instances.

Link to graphical and video abstracts, and to code:  
<https://latam.ieeer9.org/index.php/transactions/article/view/10827>

**Index Terms**—Frequency Assignment Problem, MO-FAP, SAT Encoding, Incremental SAT Solving.

## I. INTRODUCTION

THE Frequency Assignment Problem (FAP) is a fundamental NP-hard combinatorial optimization problem [1]–[3] that plays a crucial role in the management of the radio frequency spectrum. The objective of FAP is to assign frequencies to transmitters to satisfy both assignment and interference constraints while optimizing a specific criterion, such as the number of frequencies used, the maximum frequency span, or the minimum interference. Interference constraints are typically modeled as minimum frequency separation requirements between pairs of potentially interfering devices. FAP has numerous practical applications in wireless communication systems, including mobile networks, broadcasting, satellite communications, wireless local area networks, and military communication systems where frequency channels

must be efficiently allocated to minimize interference and optimize spectrum resource utilization [2], [3]. In the context of rapidly developing modern communication systems such as 5G/6G, IoT, and large-scale wireless networks, the demand for spectrum utilization is continuously increasing while spectrum resources remain inherently limited [4]–[6]. Therefore, constructing accurate models and developing efficient solving methods for the FAP is essential to satisfy interference constraints and optimize spectrum efficiency. Furthermore, advancing these optimization methods directly contributes to the United Nations Sustainable Development Goals for Latin America, particularly SDG 9 (Industry, Innovation, and Infrastructure), by fostering resilient, efficient, and sustainable telecommunication infrastructures in the region.

The comprehensive survey by Aardal et al. [2] categorized the FAP into several variants based on different constraints and objective functions. Among these, three prominent variants are: the Minimum Order Frequency Assignment Problem (MO-FAP), aiming to minimize the number of used frequencies; the Minimum Span Frequency Assignment Problem (MS-FAP), aiming to minimize the frequency span (the difference between the maximum and minimum used frequencies); and the Minimum Interference Frequency Assignment Problem (MI-FAP), aiming to minimize the total interference generated across links. These FAP variants have been extensively studied since before the 2000s [2], with many benchmark instances archived in the FAP database (<https://fap.zib.de/>). Despite this long research history, they continue to attract significant attention, with recent advances reported for MO-FAP [7] (2023), MI-FAP [8] (2022), and MS-FAP [9] (2025). This sustained interest highlights the ongoing relevance of FAP, particularly in modern telecommunications, where minimizing frequency usage directly reduces spectrum licensing costs.

Due to the NP-hard nature of the problem, the majority of the research on MO-FAP has concentrated on heuristic and metaheuristic approaches to find high-quality solutions within a reasonable time. Representative methods include Tabu Search [10]–[12], Ant Colony Optimization (ACO) [13], [14], hyper-heuristics [15], as well as neighborhood search strategies and hybrid heuristics [16]. Recent metaheuristic methods, such as Extremal Optimization (EO) [7], achieve high-quality solutions on standard benchmarks. However, their approximate nature prevents them from certifying global optimality. Moreover, many approaches (including EO) rely on iteration-based stopping criteria rather than runtime limits, which may bias evaluations and hinder fair comparisons of solver performance.

To address these limitations, this paper proposes an exact approach based on Boolean Satisfiability (SAT) with three

The associate editor coordinating the review of this manuscript and approving it for publication was Carolina Del-Valle-Soto (*Corresponding author: Nghia Dao Xuan*).

Nghia Dao Xuan, P. D. Anh, and K. T. Van are with the VNU University of Engineering and Technology, Hanoi, Vietnam (e-mails: 25025081@vnu.edu.vn, 24020277@vnu.edu.vn, and khanhtv@vnu.edu.vn).

core contributions. First, we develop the Proposed Order-based SAT Encoding (POSE) for the interference constraints. POSE uses standard order variables but introduces a problem-specific interval-boundary formulation that replaces many explicit pairwise conflict clauses of the direct SAT encoding. Second, we introduce an objective function representation based on Sequential Counter encoding [17], integrated into an incremental SAT optimization strategy (INCSC) to support iterative tightening of the number of assigned frequencies. Finally, we evaluate the proposed approach on 35 CALMA and DUTest benchmark instances and compare it against SAT-encoding variants and other exact baselines. The results show that the proposed framework certifies optimality or infeasibility on the tested instances within the reported time limit.

The remainder of this paper is organized as follows. Section II formulates the MO-FAP and reviews related work. Section III details the proposed SAT encoding method, including the use of order variables for interference constraints and the incremental objective optimization strategy. Section IV analyzes the experimental results. Finally, Section V summarizes the contributions and outlines future research directions.

## II. THE MO-FAP AND RELATED WORK

### A. The MO-FAP

Let  $V = \{1, 2, \dots, |V|\}$  be the set of vertices (transmitters) in the network, where each transmitter  $v \in V$  must be assigned exactly one frequency from a valid pre-licensed frequency domain  $D_v$ . Here,  $D = \bigcup_{v \in V} D_v$  represents the global domain of all available frequencies. The set of edges  $E$  represents the frequency interference constraints and is partitioned into two disjoint subsets:  $E_{eq}$  (bidirectional constraints) and  $E_{ineq}$  (interference constraints). Let  $d_{vw}$  denote the required frequency separation distance between two adjacent transmitters  $v$  and  $w$ . To provide a rigorous mathematical definition of the MO-FAP, we present an Integer Linear Programming (ILP) formulation based on the standard MO-FAP modeling framework in the literature [2]. The formulation follows the MO-FAP benchmark setting adopted by Gomez and Ayala [7], using benchmark instances from the FAP web database.

Let  $x_{vf} \in \{0, 1\}$  be a binary decision variable indicating whether frequency  $f \in D_v$  is assigned to transmitter  $v$ . Similarly, let  $y_f \in \{0, 1\}$  be a binary variable indicating whether frequency  $f \in D$  is used by at least one transmitter. The MO-FAP is formulated by the objective function (1) and constraints (2)–(7):

$$\text{Minimize } \sum_{f \in D} y_f \quad (1)$$

Subject to

$$\sum_{f \in D_v} x_{vf} = 1, \quad \forall v \in V \quad (2)$$

$$x_{vf} \leq y_f, \quad \forall v \in V, f \in D_v \quad (3)$$

$$x_{vf} + \sum_{h \in D_w: |f-h| \neq d_{vw}} x_{wh} \leq 1, \quad \forall vw \in E_{eq}, f \in D_v \quad (4)$$

$$x_{vf} + \sum_{h \in D_w: |f-h| \leq d_{vw}} x_{wh} \leq 1, \quad \forall vw \in E_{ineq}, f \in D_v \quad (5)$$

$$x_{vf} \in \{0, 1\}, \quad \forall v \in V, f \in D_v \quad (6)$$

$$y_f \in \{0, 1\}, \quad \forall f \in D. \quad (7)$$

In this formulation, the objective function (1) minimizes the total number of utilized frequencies. Constraint (2) ensures that exactly one valid frequency is assigned per transmitter; notably, any hardware-level pre-assignments are seamlessly handled by naturally restricting the initial domain  $D_v$  to a single frequency. Constraint (3) logically links the local assignments  $x_{vf}$  to the global usage variables  $y_f$ . Interference is modeled through aggregated packing constraints: for each pair  $vw \in E_{eq}$  or  $vw \in E_{ineq}$ , constraints (4) and (5) ensure that if  $x_{vf} = 1$ , then all conflicting assignments  $x_{wh}$  (based on the required separation) are forbidden by forcing their sum to zero.

### B. Related Work

The MO-FAP and related frequency assignment variants have been studied over several decades. Existing solution approaches can be broadly categorized into two main classes: exact methods and approximation algorithms.

The first category encompasses studies dedicated to formalizing mathematical models and developing exact solution methods. The pioneering work of Baybars [18] formulated the assignment of broadcasting frequencies as a 0-1 integer programming problem. Driven by this foundational conceptualization and the pursuit of global optimality, a diverse array of exact algorithms was actively explored in the following decades. These efforts include branch-and-cut techniques [19], [20], partial constraint satisfaction formulations related to FAP [21], enumerative algorithms [22], and graph-theoretic approaches [23]. A comprehensive synthesis of these milestones, which unified various FAP variants and established the core ILP models, was later provided in the classic survey by Aardal *et al.* [2]. Subsequent exact approaches, including dynamic programming methods [24], still face the combinatorial growth of the search space. As a result, these methods can become computationally demanding on large or heavily constrained instances [2], [24].

To address these computational challenges, many studies have considered approximation algorithms. A notable milestone during this transition is the work of Dib *et al.* [10], who proposed a hybrid optimization method combining constraint propagation, nogood management, and Tabu Search for the frequency assignment problem. Following this, the field witnessed a proliferation of heuristic and metaheuristic methods to handle large-scale networks. This era was marked by extensive contributions from Alrajhi and colleagues, who consistently proposed algorithms based on Tabu Search hybridized with various neighborhood structures [12], dynamic multiple neighborhood structures [16], variations of Ant Colony Optimization (including DACO) [13], [14], and dynamic hyper-heuristics [15]. Recently, Gomez and Ayala [7] applied the Extremal Optimization (EO) algorithm to the MO-FAP variant and compared their results with the benchmark results summarized by Aardal *et al.* [2] and empirical results reported by Alrajhi's group.

Overall, although a rich set of solution methods has been developed for the MO-FAP, several limitations remain. First, different evaluation protocols, such as iteration-based stopping

criteria used in [7] instead of runtime limits, can make solver comparisons less direct. Second, while metaheuristics can quickly reach high-quality solutions on large instances, they cannot certify global optimality. In contrast, traditional exact methods (ILP, CP, DP) can become computationally demanding on difficult instances [2], [3]. Recently, SAT-based approaches have been used for exact solving in combinatorial problems [25], including graph labeling problems [26], [27]. These works show that SAT encodings can provide exact certification while benefiting from modern SAT solvers. Motivated by these advances, this work proposes a SAT-based exact framework for MO-FAP that combines standard order variables for the interference constraints with incremental optimization of the objective function.

### III. PROPOSED EXACT APPROACH

This section details the proposed SAT-based methodology for solving the MO-FAP. First, a domain preprocessing technique is applied to shrink the initial search space (Section III-A). Next, we present a baseline Direct SAT Encoding (DSE) for the assignment, exact-distance, and interference constraints (2), (4), and (5) (Section III-B). To overcome the combinatorial explosion inherent in the direct pairwise interference encoding, the Proposed Order-based SAT Encoding (POSE) is then introduced for the interference constraints (Section III-C). Finally, the min-order objective (1) is handled through active-frequency variables and an incremental SAT strategy (Section III-D).

#### A. Domain Preprocessing

Before constructing the optimization models, the initial frequency domains are first simplified by an arc-consistency preprocessing step [2], [10]. The purpose of this step is to remove a candidate frequency  $f \in D_v$  whenever there exists an adjacent transmitter  $w$  such that no frequency in  $D_w$  can be paired with  $f$  without violating the constraint on edge  $vw$ . In addition, pre-assigned transmitters are handled by setting their domains to  $D_v = \{p_v\}$ , where  $p_v$  is the prescribed frequency of transmitter  $v$ . The output is a reduced set of domains, still denoted by  $D_v$ , which is used by the subsequent formulations and encodings. If this preprocessing step makes some domain  $D_v$  empty, then no feasible assignment can satisfy constraint (2), which requires each transmitter to be assigned exactly one frequency. Therefore, the instance is declared infeasible before any SAT or ILP model is constructed. In the subsequent formulations, each remaining domain  $D_v$  is thus assumed to be nonempty.

For each edge  $vw$ , a candidate frequency  $f \in D_v$  is retained only when there exists a frequency  $h \in D_w$  that satisfies the corresponding edge constraint. For interference constraints and bidirectional constraints, the corresponding domain filtering rules are defined in (8) and (9), respectively:

$$D_v \leftarrow \{f \in D_v \mid \exists h \in D_w : |f - h| > d_{vw}\}, \quad \forall vw \in E_{ineq} \quad (8)$$

$$D_v \leftarrow \{f \in D_v \mid \exists h \in D_w : |f - h| = d_{vw}\}, \quad \forall vw \in E_{eq}. \quad (9)$$

Removing a frequency by the preprocessing rules does not remove any feasible solution of the original problem. Consider a transmitter-frequency pair  $(v, f)$  with  $f \in D_v$ . If  $f$  does not satisfy the retaining condition in either (8) or (9) for an edge  $vw$ , then no frequency in  $D_w$  can be assigned together with  $f$  while satisfying the constraint on that edge. Therefore,  $f$  cannot be assigned to  $v$  in any feasible solution and can be safely removed from  $D_v$  before model construction.

#### B. Direct SAT Encoding

Unlike the ILP formulation, our SAT encoding relies on the positional order of the candidate frequencies. We define a Boolean variable  $b_{v,j} = \text{True}$  if and only if transmitter  $v$  is assigned the  $j$ -th frequency in its preprocessed domain  $D_v$  (sorted in ascending order). Let  $D_{v,j}$  denote the actual frequency value at index  $j$ . Based on this notation, the baseline feasibility constraints are encoded into three logical formulations, expressed in (10)–(12):

$$\sum_{j=1}^{|D_v|} b_{v,j} = 1, \quad \forall v \in V \quad (10)$$

$$b_{v,j} \rightarrow \bigvee_{j'} b_{w,j'}, \quad \forall vw \in E_{eq}, \forall j \in \{1, \dots, |D_v|\}, \quad (11)$$

$$\begin{aligned} &\forall j' \in \{1, \dots, |D_w|\} : |D_{v,j} - D_{w,j'}| = d_{vw} \\ &\neg b_{v,j} \vee \neg b_{w,j'}, \quad \forall vw \in E_{ineq}, \forall j \in \{1, \dots, |D_v|\}, \\ &\forall j' \in \{1, \dots, |D_w|\} : |D_{v,j} - D_{w,j'}| \leq d_{vw}. \end{aligned} \quad (12)$$

Equation (10) encodes the Exactly-One constraint. Although written in a numerical form for clarity, it can be translated into Conjunctive Normal Form (CNF) using standard cardinality encodings (e.g., At-Most-One and At-Least-One), which are readily supported by modern SAT libraries [28], [29]. Notably, no additional clauses are required to handle the pre-assignment constraint. Since the preprocessed domain of any pre-assigned transmitter naturally contains exactly one element ( $|D_v| = 1$ ), (10) inherently reduces to  $b_{v,1} = \text{True}$ , natively satisfying the condition. Bidirectional constraints are expressed in (11): if transmitter  $v$  is assigned the  $j$ -th frequency, the paired transmitter  $w$  must be assigned a frequency index  $j'$  such that the frequency separation between their actual values is exactly  $d_{vw}$ , and vice versa. Finally, interference constraints (which correspond to the binary packing constraints in the ILP model) are handled in (12) via conflict clauses: if transmitter  $v$  selects the frequency index  $j$ , then the adjacent transmitter  $w$  is strictly prohibited from adopting any frequency index  $j'$  whose separation is less than or equal to the safety threshold  $d_{vw}$ .

Thus, the Boolean variables  $b_{v,j}$  and constraints (10)–(12) ensure that every satisfying assignment of the DSE formula corresponds to a feasible frequency assignment satisfying constraints (2), (4), and (5). Conversely, any frequency assignment satisfying these constraints can be translated into a satisfying assignment of the DSE formula. This correctness property is formalized in Proposition 1.

*Proposition 1:* For the preprocessed domains, every satisfying assignment of the DSE constraints (10)–(12) induces a frequency assignment satisfying constraints (2), (4), and (5). Conversely, every frequency assignment satisfying these constraints induces a satisfying assignment of the DSE constraints.

*Proof:* Consider the direct mapping in which  $b_{v,j}$  is true exactly when transmitter  $v$  is assigned frequency  $D_{v,j}$  ( $x_{v,D_{v,j}} = 1$  in the ILP formulation). Suppose first that the DSE constraints have a satisfying assignment. By (10), exactly one  $b_{v,j}$  is true for each transmitter  $v$ , so the induced frequency assignment selects exactly one frequency for each transmitter and satisfies (2). For every edge  $vw \in E_{eq}$ , constraint (11) states that once  $v$  selects  $D_{v,j}$ , transmitter  $w$  must select one of the frequencies  $D_{w,j'}$  whose distance from  $D_{v,j}$  is exactly  $d_{vw}$ . Therefore, the selected pair on edge  $vw$  satisfies (4). For every edge  $vw \in E_{ineq}$ , constraint (12) adds a conflict clause for each pair with  $|D_{v,j} - D_{w,j'}| \leq d_{vw}$ , preventing both variables in any such pair from being true simultaneously. Hence, every selected pair has distance greater than  $d_{vw}$ , so (5) holds.

Conversely, let a frequency assignment satisfy (2), (4), and (5). For each transmitter  $v$ , let  $j$  be the index such that the assigned frequency of  $v$  is  $D_{v,j}$ . The corresponding assignment of the DSE variables sets  $b_{v,j}$  to true and sets  $b_{v,k}$  to false for all  $k \neq j$ . Since the assignment selects exactly one frequency for each transmitter, the DSE exactly-one constraint (10) is satisfied. For every exact-distance edge  $vw \in E_{eq}$ , constraint (4) forbids all pairs whose distance is not  $d_{vw}$ . Therefore, if a pair is selected by the frequency assignment, its distance must be exactly  $d_{vw}$ , and the corresponding implication in (11) is satisfied. Similarly, for every interference edge  $vw \in E_{ineq}$ , constraint (5) forbids all pairs whose distance is at most  $d_{vw}$ . Therefore, no forbidden pair in (12) is selected simultaneously. Hence, all DSE constraints are satisfied. ■

### C. Proposed Order-Based SAT Encoding

Although DSE provides an intuitive formulation, expressing the interference constraints (12) in this manner may require many clauses, since each forbidden frequency pair is encoded separately. To overcome this limitation, POSE uses order variables following order encoding [30]–[33] to reformulate the MO-FAP interference constraints without encoding each forbidden frequency pair separately. Note that this work does not introduce a new order encoding. The main contribution of POSE is the adaptation of order variables to the interference constraints of MO-FAP. It links these variables with  $b_{v,j}$  and uses the resulting representation to encode forbidden frequency intervals as an alternative to the pairwise formulation used in DSE.

Specifically, while retaining the Boolean variables  $b_{v,j}$  and the preprocessed domain  $D_v$ , POSE introduces a supplementary set of Boolean variables, denoted by  $g_{v,j}$ . The domain  $D_v$  is sorted in increasing order, and  $g_{v,j}$  represents whether the selected frequency index of transmitter  $v$  is greater than or equal to  $j$ . With  $|D_v|$  denoting the number of candidate

frequencies in the domain of transmitter  $v$ , the variables  $b_{v,j}$  and  $g_{v,j}$  are linked by constraints (13)–(16):

$$b_{v,j} \leftrightarrow g_{v,j} \wedge \neg g_{v,j+1}, \forall v \in V, \forall j \in \{1, \dots, |D_v| - 1\} \quad (13)$$

$$b_{v,|D_v|} \leftrightarrow g_{v,|D_v|}, \quad \forall v \in V \quad (14)$$

$$g_{v,1}, \quad \forall v \in V \quad (15)$$

$$g_{v,j} \rightarrow g_{v,j-1}, \quad \forall v \in V, \forall j \in \{2, \dots, |D_v|\}. \quad (16)$$

For  $j < |D_v|$ , constraint (13) makes  $b_{v,j}$  true when  $g_{v,j}$  is true and  $g_{v,j+1}$  is false, indicating that  $j$  is the selected frequency index for transmitter  $v$ . Constraint (14) handles the last index of the domain. Constraints (15) and (16) ensure that the true  $g$  variables form a nonempty prefix. Together, these constraints establish the semantics of the  $g$  variables, with exactly one  $b_{v,j}$  true for each transmitter. Lemma 1 formalizes these properties.

*Lemma 1:* For each transmitter  $v$ , constraints (13)–(16) enforce exactly one true variable among  $\{b_{v,j} \mid j \in \{1, \dots, |D_v|\}\}$ . If  $b_{v,t}$  is this true variable, then  $g_{v,j}$  is true if and only if the selected frequency index  $t$  is greater than or equal to  $j$  in the sorted domain  $D_v$ .

*Proof:* Let  $n = |D_v|$  and consider any assignment satisfying (13)–(16). By (15),  $g_{v,1}$  is true. Constraint (16) implies that if  $g_{v,j}$  is true, then all variables  $g_{v,k}$  with  $k < j$  are also true. Therefore, the true  $g$  variables form a nonempty prefix of the ordered domain.

First, suppose that there exists an index  $r$  such that  $g_{v,r}$  is false, and let  $r$  be the smallest such index. Then  $r > 1$ . Set  $t = r - 1$ . By the choice of  $r$ ,  $g_{v,t}$  is true and  $g_{v,t+1}$  is false. Substituting  $j = t$  into (13) gives  $b_{v,t} \leftrightarrow (g_{v,t} \wedge \neg g_{v,t+1})$ , so  $b_{v,t}$  is true. For any  $k < t$ ,  $g_{v,k+1}$  is true, so (13) makes  $b_{v,k}$  false. For any  $t < k < n$ ,  $g_{v,k}$  is false, so (13) also makes  $b_{v,k}$  false. Since  $g_{v,n}$  is false in this case, (14) makes  $b_{v,n}$  false. Hence, exactly one  $b$  variable is true, namely  $b_{v,t}$ . Moreover,  $g_{v,j}$  is true exactly for the indices  $j \leq t$ , which is equivalent to  $t \geq j$ .

It remains to consider the case in which no  $g_{v,j}$  is false. Then all variables  $g_{v,j}$  are true. By (14),  $b_{v,n}$  is true. For every  $k < n$ ,  $g_{v,k+1}$  is true, so (13) makes  $b_{v,k}$  false. Hence, exactly one  $b$  variable is true, namely  $b_{v,n}$ . In this case the selected index is  $t = n$ , and  $g_{v,j}$  is true for every valid index  $j$ , which is equivalent to  $t \geq j$ . ■

Subsequently, the order variables  $g$  are used to reformulate the interference constraints. When transmitter  $v$  is assigned the frequency  $D_{v,j}$  (i.e.,  $b_{v,j} = \text{True}$ ), it induces the forbidden interval  $[D_{v,j} - d_{vw}, D_{v,j} + d_{vw}]$  for its adjacent transmitter  $w$ . Panels (a)–(f) of Fig. 1 illustrate the relative positions between this forbidden interval and the preprocessed domain  $D_w$  for Cases 1–6, respectively. The domain  $D_w$  is nonempty at this stage. Three situations may then arise. In Cases 1 and 2, the domain  $D_w$  lies entirely outside the forbidden interval, either to its left or to its right. In this situation, every value in  $D_w$  already satisfies the interference constraint, so no additional constraint is required. In Cases 3–5, the domain  $D_w$  overlaps the forbidden interval on the left, on the right, or on both sides. In these cases, the remaining values of  $D_w$  outside the forbidden interval must be characterized explicitly. In Case 6, all values in  $D_w$  fall inside the forbidden interval.

Consequently,  $D_{v,j}$  has no compatible value in  $D_w$  and is therefore removed from  $D_v$  during preprocessing.

For Cases 3–5, in which at least one value of  $D_w$  lies in the forbidden interval, the corresponding clauses are constructed by (17), (18), and (19):

$$\begin{aligned} b_{v,j} &\rightarrow \neg g_{w,L+1}, \\ L &= \arg \max_k \{k | D_{w,k} < D_{v,j} - d_{vw}\}, \\ \forall vw \in E_{ineq}, j \in \{1, \dots, |D_v|\}, \\ \{k | D_{w,k} < D_{v,j} - d_{vw}\} &\neq \emptyset, \\ \{k | D_{v,j} - d_{vw} \leq D_{w,k} \leq D_{v,j} + d_{vw}\} &\neq \emptyset, \\ \{k | D_{w,k} > D_{v,j} + d_{vw}\} &= \emptyset \end{aligned} \quad (17)$$

$$\begin{aligned} b_{v,j} &\rightarrow g_{w,R}, \\ R &= \arg \min_k \{k | D_{w,k} > D_{v,j} + d_{vw}\}, \\ \forall vw \in E_{ineq}, j \in \{1, \dots, |D_v|\}, \\ \{k | D_{w,k} < D_{v,j} - d_{vw}\} &= \emptyset, \\ \{k | D_{v,j} - d_{vw} \leq D_{w,k} \leq D_{v,j} + d_{vw}\} &\neq \emptyset, \\ \{k | D_{w,k} > D_{v,j} + d_{vw}\} &\neq \emptyset \end{aligned} \quad (18)$$

$$\begin{aligned} b_{v,j} &\rightarrow (\neg g_{w,L+1} \vee g_{w,R}), \\ L &= \arg \max_k \{k | D_{w,k} < D_{v,j} - d_{vw}\}, \\ R &= \arg \min_k \{k | D_{w,k} > D_{v,j} + d_{vw}\}, \\ \forall vw \in E_{ineq}, j \in \{1, \dots, |D_v|\}, \\ \{k | D_{w,k} < D_{v,j} - d_{vw}\} &\neq \emptyset, \\ \{k | D_{v,j} - d_{vw} \leq D_{w,k} \leq D_{v,j} + d_{vw}\} &\neq \emptyset, \\ \{k | D_{w,k} > D_{v,j} + d_{vw}\} &\neq \emptyset. \end{aligned} \quad (19)$$

For each considered interference edge  $vw$  and assignment  $D_{v,j}$ , the three conditions in (17)–(19) specify whether  $D_w$  has values to the left of the forbidden interval, inside the forbidden interval, and to the right of the forbidden interval, respectively. These conditions ensure that the boundary indices  $L$  and  $R$  are introduced only when the corresponding side exists and that Cases 3–5 are actual overlap cases.

In Case 3, the forbidden interval overlaps the right part of  $D_w$ , so the remaining values of  $D_w$  lie entirely to the left of the forbidden interval, and the clause is given by (17). In Case 4, the forbidden interval overlaps the left part of  $D_w$ , so the remaining values of  $D_w$  lie entirely to the right of the forbidden interval, and the clause is given by (18). In Case 5, the forbidden interval lies strictly inside  $D_w$ , so the remaining values of  $D_w$  appear on both sides of the forbidden interval, and the clause is given by (19). Notably,  $L$  is used only in (17) and (19), whereas  $R$  is used only in (18) and (19). Therefore, each boundary index is defined only in the cases where the corresponding side of  $D_w$  exists.

Case 6 occurs when every value in  $D_w$  lies in the forbidden interval induced by  $D_{v,j}$ . In the current formulation, this case is eliminated by preprocessing, whereas without preprocessing it can be encoded directly by the unary clause  $\neg b_{v,j}$ .

The formulas also provide a simple estimate of the clause counts for the DSE and POSE interference encodings. From (17), (18), and (19), each fixed assignment  $b_{v,j}$  and adjacent transmitter  $w$  generates at most one boundary implication, because the forbidden interval falls into exactly one of the

relevant relative-position cases. Therefore, over all  $|D_v|$  candidate assignments of transmitter  $v$ , one oriented interference edge from  $v$  to  $w$  generates at most  $\mathcal{O}(|D_v|)$  POSE boundary implications. In contrast, DSE checks every pair of candidate frequencies  $(D_{v,j}, D_{w,j'})$  and adds a conflict clause whenever  $|D_{v,j} - D_{w,j'}| \leq d_{vw}$ . In the worst case, all  $|D_v||D_w|$  pairs may be forbidden, yielding  $\mathcal{O}(|D_v||D_w|)$  pairwise conflict clauses. Proposition 2 establishes that the POSE boundary implications preserve the same feasible assignments as the DSE pairwise interference clauses.

**Proposition 2:** For the preprocessed domains, POSE ensures that each transmitter is assigned exactly one frequency. More precisely, constraints (13)–(16) are equivalent to the DSE constraint (10). Under these constraints, (17) in Case 3, (18) in Case 4, and (19) in Case 5 are satisfied if and only if the assignment to the variables  $b$  satisfies the DSE interference constraints (12). POSE uses the same exact-distance constraints (11) as DSE. Therefore, Proposition 1 implies that POSE represents exactly the frequency assignments satisfying constraints (2), (4), and (5).

*Proof:* By Lemma 1, every assignment satisfying constraints (13)–(16) has exactly one true variable  $b_{v,t}$  for each transmitter  $v$ . Conversely, suppose that (10) holds and  $b_{v,t}$  is the unique true variable for transmitter  $v$ . Set  $g_{v,k}$  to true if and only if  $t \geq k$ . This assignment satisfies constraints (13)–(16). Hence, these constraints are equivalent to (10).

Consider an interference edge  $vw \in E_{ineq}$  and an index  $j \in \{1, \dots, |D_v|\}$  such that  $b_{v,j}$  is true. Let  $b_{w,t}$  be the unique true variable for transmitter  $w$ . By Lemma 1,  $g_{w,k}$  is true if and only if  $t \geq k$ . The indices used below are well defined by the corresponding cases described above.

In Case 3, constraint (17) requires  $\neg g_{w,L+1}$  because  $b_{v,j}$  is true. By Lemma 1,  $g_{w,L+1}$  is true if and only if  $t \geq L+1$ . Therefore,  $\neg g_{w,L+1}$  is equivalent to  $t \leq L$ . Since  $D_w$  is sorted and  $L$  is the largest index satisfying  $D_{w,L} < D_{v,j} - d_{vw}$ , the selected frequency  $D_{w,t}$  lies to the left of the forbidden interval. For each  $k \in \{L+1, \dots, |D_w|\}$ , DSE includes the clause  $\neg b_{v,j} \vee \neg b_{w,k}$  in (12). Since  $b_{v,j}$  is true, these clauses force  $b_{w,L+1}, \dots, b_{w,|D_w|}$  to be false. Hence, the unique true variable  $b_{w,t}$  must satisfy  $t \leq L$ . DSE therefore imposes the same restriction as (17).

In Case 4, constraint (18) requires  $g_{w,R}$  because  $b_{v,j}$  is true. By Lemma 1, this condition is equivalent to  $t \geq R$ . Since  $D_w$  is sorted and  $R$  is the smallest index satisfying  $D_{w,R} > D_{v,j} + d_{vw}$ , the selected frequency  $D_{w,t}$  lies to the right of the forbidden interval. For each  $k \in \{1, \dots, R-1\}$ , DSE includes the clause  $\neg b_{v,j} \vee \neg b_{w,k}$  in (12). Since  $b_{v,j}$  is true, these clauses force  $b_{w,1}, \dots, b_{w,R-1}$  to be false. Hence, the unique true variable  $b_{w,t}$  must satisfy  $t \geq R$ . DSE therefore imposes the same restriction as (18).

Case 5 combines the two possibilities established in Cases 3 and 4. When  $b_{v,j}$  is true, constraint (19) requires  $\neg g_{w,L+1} \vee g_{w,R}$ , which is equivalent to  $t \leq L$  or  $t \geq R$ . DSE includes the clause  $\neg b_{v,j} \vee \neg b_{w,k}$  for each  $k \in \{L+1, \dots, R-1\}$  in (12). Since  $b_{v,j}$  is true, these clauses force  $b_{w,L+1}, \dots, b_{w,R-1}$  to be false. Hence, the unique true variable  $b_{w,t}$  must satisfy either  $t \leq L$  or  $t \geq R$ . DSE therefore imposes the same restriction as (19).

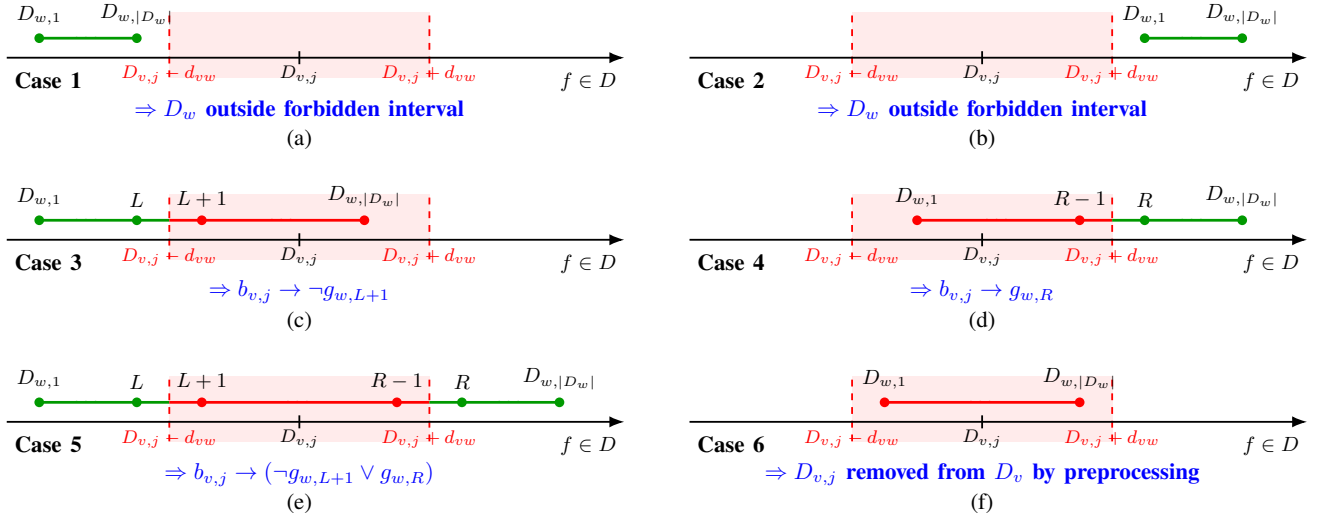


Fig. 1. Relative positions between the forbidden interval  $[D_{v,j} - d_{vw}, D_{v,j} + d_{vw}]$  and  $D_w$ : (a)–(b) no overlap; (c)–(d) one-sided overlap; (e) two-sided overlap; and (f) complete coverage, causing  $D_{v,j}$  to be removed by preprocessing. Red and green segments denote forbidden and remaining values, respectively.

In each case, the POSE constraint and the corresponding DSE clauses are equivalent to the same condition on the selected index  $t$ . Therefore, the equivalence holds in both directions. Cases 1 and 2 require no additional constraints, while Case 6 cannot occur in the preprocessed domains. If  $b_{v,j}$  is false, frequency  $D_{v,j}$  is not selected, and neither encoding restricts transmitter  $w$  based on this frequency. Therefore, over every interference edge  $vw \in E_{ineq}$  and every index  $j \in \{1, \dots, |D_v|\}$ , constraints (17), (18), and (19) exclude exactly the same assignments as (12). Together with the equivalence between constraints (13)–(16) and (10), and the unchanged exact-distance constraints (11), this establishes the result by Proposition 1. ■

#### D. Objective Function Optimization via Incremental SAT

After the feasibility constraints (2), (4), and (5) have been encoded, the remaining task is to minimize the total number of distinct frequencies utilized. Let  $D_m$  denote the frequency value at the  $m$ -th position of the sorted global domain  $D$ , where  $1 \leq m \leq N = |D|$ . We introduce Boolean variables  $A_m$ , where  $A_m$  is forced to be True whenever the frequency  $D_m$  is used by at least one transmitter. The connection between the local assignment variables  $b_{v,j}$  and the global frequency-usage variables  $A_m$  is expressed by (20):

$$b_{v,j} \rightarrow A_m, \quad \forall v \in V, \forall j, m : D_{v,j} = D_m. \quad (20)$$

As illustrated in Fig. 2, each selected local assignment  $b_{v,j}$  activates the corresponding global usage variable  $A_m$ , and the objective encoding minimizes the number of active  $A_m$  variables.

The optimization of the objective function is performed by a linear search over the number of assigned frequencies. The solver first finds a feasible assignment and uses the number of assigned frequencies as the initial upper bound  $UB$ . It then searches for a better solution satisfying an At-Most- $k$

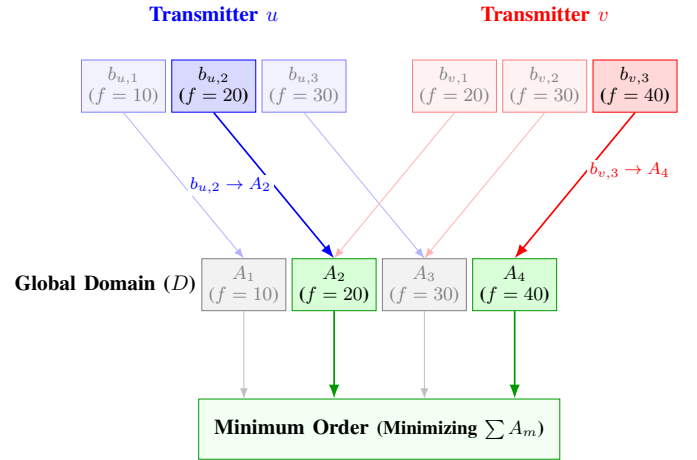


Fig. 2. Mapping from local assignment variables ( $b_{v,j}$ ) to global frequency-usage variables ( $A_m$ ).

constraint with  $k = UB - 1$ . Whenever such a solution is found,  $UB$  is updated and the bound is tightened again.

These search iterations could be executed as independent SAT calls, where each new value of  $k$  requires constructing a new At-Most- $k$  constraint and solving a fresh formula. However, this strategy discards the learned clauses obtained in previous iterations. Therefore, we use the incremental solving mechanism supported by PySAT [28], [29]. The baseline incremental strategy, denoted as INC, handles the At-Most- $k$  constraints by `ITotalizer` [34], [35]. The proposed strategy, denoted as INCSC, uses a Sequential Counter encoding instead. The same optimization pipeline is used by both INC and INCSC. They differ only in the objective encoding and the assumption literal used to enforce each At-Most- $k$  bound. The complete pipeline is summarized in Table I.

In INC, the `ITotalizer` construction adds the required cardinality clauses to the solver and returns a sequence of output literals, denoted by `rhs` in PySAT. Each output literal

represents a cardinality threshold, so the assumption  $\neg \text{rhs}[k]$  enforces the At-Most- $k$  constraint. During Step 5, the solver searches for a better feasible assignment using at most  $k = UB - 1$  frequencies. This bound is imposed by passing the assumption  $\neg \text{rhs}[k]$  to the solver, which forbids assignments using more than  $k$  frequencies. If the call returns SAT, the returned assignment becomes the new *best*,  $UB$  is updated to its number of assigned frequencies, and the next iteration repeats the same procedure with the updated value of  $UB$ .

Although INC supports learned-clause reuse through assumptions, its Totalizer-based objective encoding introduces more auxiliary structure than the Sequential Counter encoding. To reduce this overhead, INCSC keeps the same assumptions-based optimization loop as INC but replaces the Totalizer with a Sequential Counter encoding [17]. For the variables  $A_1, \dots, A_N$ , indexed by the sorted global domain  $D$ , INCSC introduces auxiliary variables  $S_{i,c}$ , where  $1 \leq i \leq N$  and  $1 \leq c \leq \min(i, K)$ . The variable  $S_{i,c}$  is true if and only if at least  $c$  variables among  $A_1, \dots, A_i$  are true. Equivalently, each prefix state  $S_i$  is a vector of  $\min(i, K)$  bits,  $S_i = (S_{i,1}, \dots, S_{i,\min(i,K)})$ :  $S_1$  has one bit,  $S_2$  has two bits, and for  $i \geq K$ ,  $S_i$  has  $K$  bits. This semantics is enforced by the constraints in (21)–(27):

$$\bigwedge_{i=1}^N (A_i \rightarrow S_{i,1}) \quad (21)$$

$$\bigwedge_{i=2}^N \bigwedge_{c=1}^{\min(i-1,K)} (S_{i-1,c} \rightarrow S_{i,c}) \quad (22)$$

$$\bigwedge_{i=2}^N \bigwedge_{c=2}^{\min(i,K)} (A_i \wedge S_{i-1,c-1} \rightarrow S_{i,c}) \quad (23)$$

$$\bigwedge_{i=2}^N \bigwedge_{c=1}^{\min(i-1,K)} (\neg A_i \wedge \neg S_{i-1,c} \rightarrow \neg S_{i,c}) \quad (24)$$

$$\bigwedge_{i=1}^K (\neg A_i \rightarrow \neg S_{i,i}) \quad (25)$$

$$\bigwedge_{i=2}^N \bigwedge_{c=2}^{\min(i,K)} (\neg S_{i-1,c-1} \rightarrow \neg S_{i,c}) \quad (26)$$

$$\bigwedge_{i=K+1}^N (A_i \rightarrow \neg S_{i-1,K}). \quad (27)$$

The Sequential Counter encoding used by INCSC contains  $\mathcal{O}(NK)$  auxiliary variables and clauses for the At-Most- $K$  constraint built in Step 4. This gives an explicit objective encoding specified by the proposed formulation, rather than relying on the library-level `ITotalizer` construction used in INC.

In Step 4 of Table I, the Sequential Counter encoding in (21)–(27) is built once for the initial bound  $K = UB - 1$ . In Step 5, any tighter At-Most- $k$  bound with  $k < K$  is enforced by the single assumption  $\neg S_{N,k+1}$ . Since  $S_{N,k+1}$  represents that at least  $k+1$  frequencies are active, assuming  $\neg S_{N,k+1}$  forces the solution to use at most  $k$  frequencies. Thus, each iteration changes only the assumption literal, while the Sequential Counter encoding remains unchanged and the

TABLE I  
INCREMENTAL SAT PIPELINE FOR MO-FAP OPTIMIZATION

| Input  | Preprocessed MO-FAP instance; selected hard-constraint encoding; selected objective encoding.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Step 1 | Construct the hard formula $F$ for the assignment, exact-distance, and interference constraints using DSE (10)–(12) or POSE (13)–(16), (17), (18), and (19).                                                                                                                                                                                                                                                                                                                                                                                            |
| Step 2 | Create the global frequency-usage variables $A_m$ for $m = 1, \dots, N$ .<br>Add the usage-linking clauses in (20) to $F$ .                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Step 3 | Solve $F$ .<br><b>If</b> $F$ is UNSAT, <b>return</b> INF.<br><b>If</b> $F$ is SAT, set $best \leftarrow$ the returned feasible assignment.<br>Set $UB \leftarrow$ the number of assigned frequencies in $best$ .<br><b>If</b> $UB = 1$ , <b>return</b> OPT with $best$ and $UB$ .                                                                                                                                                                                                                                                                       |
| Step 4 | Set $K \leftarrow UB - 1$ and build the selected objective encoding for the At-Most- $K$ constraint using <code>ITotalizer</code> for INC or Sequential Counter (21)–(27) for INCSC.                                                                                                                                                                                                                                                                                                                                                                    |
| Step 5 | <b>While</b> $UB > 1$ <b>do</b> :<br>Set $k \leftarrow UB - 1$ and solve $F$ with the corresponding bound. For INC, use the <code>ITotalizer</code> assumption $\neg \text{rhs}[k]$ to enforce At-Most- $k$ . For INCSC, use the assumption $\neg S_{N,k+1}$ when $k < K$ .<br><b>If</b> the call is UNSAT, <b>return</b> OPT with $best$ and $UB$ .<br>Set $best \leftarrow$ the returned feasible assignment.<br>Set $UB \leftarrow$ the number of assigned frequencies in $best$ .<br><b>End while</b> .<br><b>Return</b> OPT with $best$ and $UB$ . |
| Output | INF, or OPT with $best$ as the best feasible assignment and $UB$ as its number of assigned frequencies.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

solver can reuse learned clauses. For both INC and INCSC, once the At-Most- $k$  constraint is enforced by the corresponding assumption, the same stopping condition applies. Its correctness is formalized in Proposition 3.

*Proposition 3:* Let  $UB > 1$  denote the number of distinct frequencies used by the current best feasible assignment. If the next SAT call enforces the At-Most- $k$  constraint with  $k = UB - 1$  over the global frequency-usage variables  $A_m$  and returns UNSAT, then the current value  $UB$  is globally optimal.

*Proof:* By the definition of  $b_{v,j}$  and by (20), if transmitter  $v$  is assigned frequency  $D_{v,j} = D_m$ , then  $b_{v,j}$  is true and the corresponding variable  $A_m$  is forced to be true. Thus, every used frequency has its corresponding variable  $A_m$  set to true. Therefore, any model satisfying an At-Most- $k$  constraint over the variables  $A_m$  uses at most  $k$  distinct frequencies. Any feasible assignment using at most  $k$  distinct frequencies can set at most  $k$  variables  $A_m$  to true, namely those corresponding to the used frequencies. Therefore, it satisfies both (20) and the At-Most- $k$  constraint, so the bounded formula does not incorrectly exclude such an assignment. Consequently, if the SAT call enforcing the At-Most- $k$  constraint with  $k = UB - 1$  returns UNSAT, then no feasible assignment exists that uses at most  $UB - 1$  distinct frequencies. This means that every feasible assignment uses at least  $UB$  distinct frequencies. On the other hand, the current best assignment is feasible and uses exactly  $UB$  distinct frequencies. Therefore,  $UB$  is the minimum possible number of distinct frequencies and is globally optimal. ■

## IV. EXPERIMENTS

## A. Experimental Setup

All experiments were conducted on a workstation with an Intel Core i7-10850H processor and 16 GB RAM running Ubuntu 24.04.3 LTS. The SAT-based methods were implemented with CaDiCaL (v1.9.5) through PySAT [28], [29]. Each run was executed under `runlim` with a time limit of 600 seconds and a memory limit of 14,000 MB.

The evaluation was performed on the standard CALMA and DUTtest benchmarks from the FAP web database <https://fap.zib.de/>. In addition to the native MO-FAP instances, we also reformulated CELAR and GRAPH instances into the MO-FAP objective. For these reformulated instances, the original constraints are kept unchanged, and only the objective is set to Minimum Order. Table II summarizes the benchmark instances used in the experiments, where  $|V|$  and  $|E|$  denote the number of transmitters and distance constraints, respectively.

TABLE II  
BENCHMARK INSTANCES USED IN THE EXPERIMENTS

| Instance | Original Name | Original Objective   | $ V $ | $ E $ |
|----------|---------------|----------------------|-------|-------|
| C01      | CELAR 01      | Minimum Order        | 916   | 5548  |
| C02      | CELAR 02      | Minimum Order        | 200   | 1235  |
| C03      | CELAR 03      | Minimum Order        | 400   | 2760  |
| C04      | CELAR 04      | Minimum Order        | 680   | 3967  |
| C05      | CELAR 05      | Minimum Span         | 400   | 2598  |
| C06      | CELAR 06      | Minimum Interference | 200   | 1322  |
| C07      | CELAR 07      | Minimum Interference | 400   | 2865  |
| C08      | CELAR 08      | Minimum Interference | 916   | 5744  |
| C09      | CELAR 09      | Minimum Interference | 680   | 4103  |
| C10      | CELAR 10      | Minimum Interference | 680   | 4103  |
| C11      | CELAR 11      | Minimum Order        | 680   | 4103  |
| G01      | GRAPH 01      | Minimum Order        | 200   | 1134  |
| G02      | GRAPH 02      | Minimum Order        | 400   | 2245  |
| G03      | GRAPH 03      | Minimum Span         | 200   | 1134  |
| G04      | GRAPH 04      | Minimum Span         | 400   | 2244  |
| G05      | GRAPH 05      | Minimum Interference | 200   | 1134  |
| G06      | GRAPH 06      | Minimum Interference | 400   | 2170  |
| G07      | GRAPH 07      | Minimum Interference | 400   | 2170  |
| G08      | GRAPH 08      | Minimum Order        | 680   | 3757  |
| G09      | GRAPH 09      | Minimum Order        | 916   | 5246  |
| G10      | GRAPH 10      | Minimum Span         | 680   | 3907  |
| G11      | GRAPH 11      | Minimum Interference | 680   | 3757  |
| G12      | GRAPH 12      | Minimum Interference | 680   | 4017  |
| G13      | GRAPH 13      | Minimum Interference | 916   | 5273  |
| G14      | GRAPH 14      | Minimum Order        | 916   | 4638  |
| T2.1     | TUD200.1      | Minimum Order        | 200   | 1171  |
| T2.2     | TUD200.2      | Minimum Order        | 200   | 1143  |
| T2.3     | TUD200.3      | Minimum Order        | 200   | 1160  |
| T2.4     | TUD200.4      | Minimum Order/Span   | 200   | 1142  |
| T2.5     | TUD200.5      | Minimum Interference | 200   | 1125  |
| T9.1     | TUD916.1      | Minimum Order        | 916   | 5177  |
| T9.2     | TUD916.2      | Minimum Order        | 916   | 5173  |
| T9.3     | TUD916.3      | Minimum Order        | 916   | 5262  |
| T9.4     | TUD916.4      | Minimum Order/Span   | 916   | 5183  |
| T9.5     | TUD916.5      | Minimum Interference | 916   | 5213  |

Table III summarizes the method names used in the experimental results. The SAT-based methods differ in terms of how the feasibility constraints and the objective function are handled. For the feasibility constraints, DSE and POSE correspond to the encodings described in Sections III-B and III-C, respectively. In addition to DSE and POSE, CARD denotes cardinality-based SAT encodings that encode (10) as

TABLE III  
METHODS COMPARED IN THE EXPERIMENTS

| Method                                             | Description                                                                                                                               |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <i>SAT encodings and objective strategies</i>      |                                                                                                                                           |
| DSE                                                | Direct SAT encoding with pairwise interference clauses.                                                                                   |
| CARD-Seq                                           | Cardinality-based SAT encoding for (10) and (12) using PySAT Sequential Counter encodings.                                                |
| POSE                                               | Proposed order-based SAT encoding.                                                                                                        |
| INC                                                | Incremental SAT optimization using <code>ITotalizer</code> .                                                                              |
| INCSC                                              | Incremental SAT optimization using Sequential Counter.                                                                                    |
| MaxSAT-RC2                                         | MaxSAT optimization using PySAT RC2 with hard feasibility clauses and unit soft clauses over the global frequency-usage variables $A_m$ . |
| <i>Main SAT configuration</i>                      |                                                                                                                                           |
| POSE+INCSC                                         | POSE combined with INCSC.                                                                                                                 |
| <i>Feasibility-encoding comparison with INCSC</i>  |                                                                                                                                           |
| POSE+INCSC                                         | POSE combined with INCSC.                                                                                                                 |
| DSE+INCSC                                          | DSE combined with INCSC.                                                                                                                  |
| CARD-Seq+INCSC                                     | CARD-Seq combined with INCSC.                                                                                                             |
| <i>Objective-optimization comparison with POSE</i> |                                                                                                                                           |
| POSE+INCSC                                         | POSE combined with INCSC.                                                                                                                 |
| POSE+INC                                           | POSE combined with INC.                                                                                                                   |
| POSE+MaxSAT-RC2                                    | POSE combined with MaxSAT-RC2.                                                                                                            |
| <i>Baseline methods</i>                            |                                                                                                                                           |
| Gurobi                                             | Gurobi MIP solver (v12.0.3).                                                                                                              |
| CPLEX-CP                                           | CP model in CPLEX Optimization Studio (v12.1.1).                                                                                          |
| CPLEX-MIP                                          | MIP model in CPLEX Optimization Studio (v12.1.1).                                                                                         |
| CP-SAT                                             | CP-SAT solver in Google OR-Tools (v9.15.6755).                                                                                            |

an Exactly-One constraint and encode (12) through At-Most-One constraints using PySAT. Among the evaluated CARD variants, including Totalizer (CARD-Totalizer) and Cardinality Network (CARD-CardNet) encodings, Sequential Counter (CARD-Seq) is selected as the representative CARD baseline in the main results. The detailed results for CARD-Totalizer and CARD-CardNet are reported in Table VII.

For the objective function, INC and INCSC correspond to the incremental SAT strategies described in Section III-D. MaxSAT-RC2 is included in the main results as an objective-optimization baseline. In this MaxSAT-RC2 formulation, the feasibility constraints and the usage-linking clauses (20) are added as hard clauses, while each global frequency-usage variable  $A_m$  contributes a unit soft clause  $\neg A_m$  with weight one. Results for an additional MaxSAT solver, EvalMaxSAT (POSE+EvalMaxSAT), are reported in Table VII.

The proposed framework is further compared against several commonly used optimization solvers, including Gurobi, CPLEX-CP, CPLEX-MIP, and OR-Tools CP-SAT. All methods were also run under the same external limits of 600 seconds and 14,000 MB memory using `runlim`. Gurobi, CPLEX-CP, and OR-Tools CP-SAT used their default solver configurations. CPLEX-MIP also used its standard solver configuration, except that zero MIP gap was set to require an exact optimality certificate and its built-in presolve routine was enabled.

The outcome of each evaluated method on each benchmark instance is reported using one of three final statuses. OPT

denotes that optimality is certified, INF denotes that the instance is proven infeasible, and TO denotes that the run reaches the 600-second time limit without a certificate. When cumulative solving time is reported, each TO run is counted as 600 seconds. Details on the benchmark data, experimental results, and source code are provided in the Data and Code Availability statement.

### B. Evaluation of Domain Preprocessing

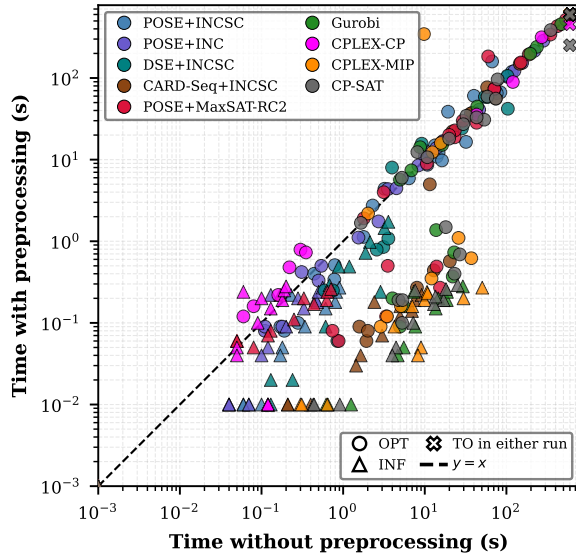


Fig. 3. Solving time with and without domain preprocessing on 35 benchmark instances.

This experiment examines whether domain preprocessing reduces solving time for the evaluated methods shown in Fig. 3. Overall, preprocessing improves solving time for most of the evaluated methods on a substantial subset of instances, with the clearest effect observed on infeasible (INF) instances. For POSE+INCSC, enabling preprocessing reduces solving time on 28 of the 35 instances. In Fig. 3, points close to the  $y = x$  line correspond to runs where preprocessing has limited effect, whereas points below the diagonal indicate faster solving after preprocessing. The largest reductions are mainly associated with infeasible instances, where inconsistent domain values can be removed before the main solving process.

### C. Evaluation of the Proposed Order-Based SAT Encoding

This experiment evaluates the effect of the SAT encoding used for the assignment, exact-distance, and interference constraints while keeping the objective-function strategy fixed. We compare POSE, DSE, and CARD-Seq, each combined with INCSC and domain preprocessing. The comparison considers three aspects: the number of variables and clauses generated by these constraint encodings, the reported objective values and certification statuses, and the solving time. The complete per-instance results are provided in Table VII, where *Sol.* denotes the reported number of assigned frequencies, *Time*

TABLE IV  
NUMBER OF VARIABLES AND CLAUSES IN THE SAT ENCODINGS OF THE ASSIGNMENT, EXACT-DISTANCE, AND INTERFERENCE CONSTRAINTS FOR INSTANCES NOT ELIMINATED BY PREPROCESSING

| Inst. | Variables     |        |     |                | Clauses       |                |     |                 |
|-------|---------------|--------|-----|----------------|---------------|----------------|-----|-----------------|
|       | DSE           | POSE   | P/D | CARD C/D       | DSE           | POSE           | P/D | CARD C/D        |
| C01   | <b>36,200</b> | 72,400 | 200 | 692,321 1912   | 1,446,348     | <b>355,326</b> | 25  | 1,982,888 137   |
| C02   | <b>8,004</b>  | 16,008 | 200 | 181,777 2271   | 349,519       | <b>82,133</b>  | 23  | 522,669 150     |
| C03   | <b>15,892</b> | 31,784 | 200 | 362,247 2279   | 702,647       | <b>173,390</b> | 25  | 1,047,531 149   |
| C04   | <b>1,960</b>  | 3,920  | 200 | 5,783 295      | <b>10,472</b> | 11,930         | 114 | 12,729 122      |
| C05   | <b>3,722</b>  | 7,444  | 200 | 73,726 1981    | 107,224       | <b>26,565</b>  | 25  | 205,588 192     |
| C11   | <b>26,856</b> | 53,712 | 200 | 548,487 2042   | 1,105,353     | <b>268,996</b> | 24  | 1,573,495 142   |
| G01   | <b>6,920</b>  | 13,840 | 200 | 71,739 1037    | 206,786       | <b>62,950</b>  | 30  | 196,572 95      |
| G02   | <b>14,624</b> | 29,248 | 200 | 172,559 1180   | 475,829       | <b>139,391</b> | 29  | 478,208 100     |
| G03   | <b>7,480</b>  | 14,960 | 200 | 463,875 6202   | 598,543       | <b>71,704</b>  | 12  | 1,341,243 224   |
| G04   | <b>14,816</b> | 29,632 | 200 | 904,601 6106   | 1,168,001     | <b>140,881</b> | 12  | 2,615,266 224   |
| G08   | <b>25,628</b> | 51,256 | 200 | 389,284 1519   | 906,033       | <b>236,932</b> | 26  | 1,079,747 119   |
| G09   | <b>36,092</b> | 72,184 | 200 | 573,735 1590   | 1,299,985     | <b>352,970</b> | 27  | 1,599,202 123   |
| G10   | <b>26,594</b> | 53,188 | 200 | 1,781,951 6701 | 2,273,085     | <b>256,868</b> | 11  | 5,163,511 227   |
| G14   | <b>36,716</b> | 73,432 | 200 | 462,509 1260   | 1,204,487     | <b>334,704</b> | 28  | 1,269,682 105   |
| T2.1  | <b>7,986</b>  | 15,972 | 200 | 173,596 2174   | 339,265       | <b>78,025</b>  | 23  | 493,586 145     |
| T2.2  | <b>200</b>    | 400    | 200 | <b>200</b> 100 | <b>300</b>    | 1,286          | 429 | <b>300</b> 100  |
| T2.3  | <b>7,066</b>  | 14,132 | 200 | 153,407 2171   | 288,705       | <b>66,557</b>  | 23  | 433,731 150     |
| T2.4  | <b>624</b>    | 1,248  | 200 | 5,275 845      | 9,049         | <b>4,050</b>   | 45  | 13,974 154      |
| T9.1  | <b>33,572</b> | 67,144 | 200 | 705,399 2101   | 1,380,601     | <b>306,313</b> | 22  | 2,000,264 145   |
| T9.2  | <b>1,180</b>  | 2,360  | 200 | 1,454 123      | 2,490         | 7,147          | 287 | <b>2,312</b> 93 |
| T9.3  | <b>34,386</b> | 68,772 | 200 | 756,304 2199   | 1,443,466     | <b>329,553</b> | 23  | 2,144,292 149   |
| T9.4  | <b>2,886</b>  | 5,772  | 200 | 9,100 315      | <b>15,888</b> | 16,813         | 106 | 20,280 128      |

P/D and C/D are percentages relative to DSE.

TABLE V  
CERTIFICATION OUTCOMES AND TOTAL SOLVING TIME OF SAT ENCODINGS AFTER PREPROCESSING.

| Method         | #OPT | #INF | #TO | #Certified | Total time (s) |
|----------------|------|------|-----|------------|----------------|
| POSE+INCSC     | 22   | 13   | 0   | 35/35      | 599.94         |
| DSE+INCSC      | 13   | 13   | 9   | 26/35      | 5723.34        |
| CARD-Seq+INCSC | 13   | 13   | 9   | 26/35      | 5786.33        |

is the solving time in seconds, and *Status* is one of OPT, INF, or TO.

Table IV reports the number of variables and clauses generated by DSE, POSE, and CARD-Seq when encoding the assignment, exact-distance, and interference constraints after preprocessing. The columns P/D and C/D report the percentage ratios of POSE to DSE and CARD-Seq to DSE, respectively. POSE introduces one order variable per domain value in addition to the assignment variables, so its number of variables is about twice that of DSE. In return, each selected frequency is handled through boundary clauses over the ordered domain of the adjacent transmitter, rather than by enumerating all conflicting frequency pairs. As a result, POSE generates fewer clauses than DSE on 18 of the 22 instances not eliminated by preprocessing. The four exceptions correspond to instances where the interference constraints generate only a small number of incompatible frequency pairs for DSE after preprocessing, while POSE still introduces the order variables and their linking/propagation clauses. CARD-Seq reduces the need to list every incompatible pair of frequencies explicitly, but it encodes each Exactly-One and At-Most-One constraint separately using Sequential Counter encodings. Since the auxiliary variables introduced for one cardinality constraint are not reused by other constraints, CARD-Seq produces many additional variables and clauses, which explains its much larger variable counts in Table IV.

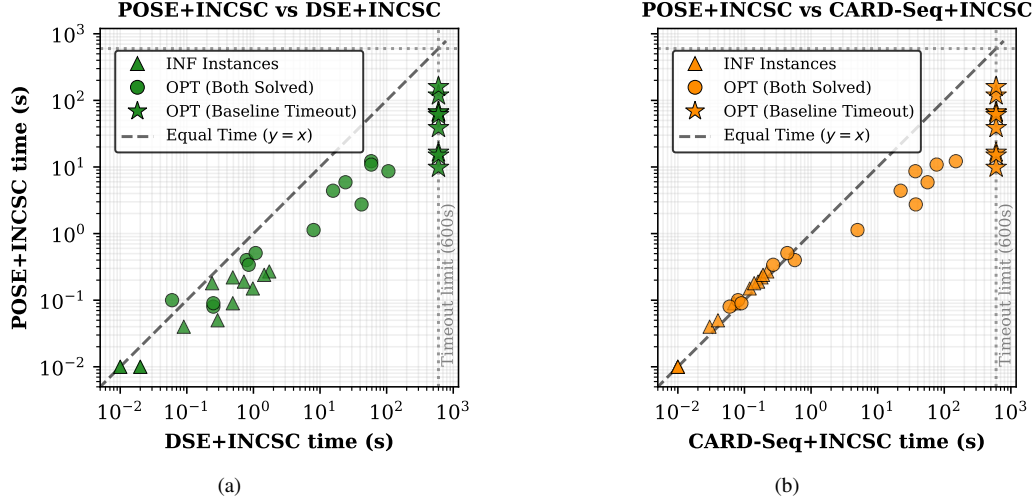


Fig. 4. Comparison of solving time on 35 benchmark instances: (a) POSE+INCSC versus DSE+INCSC; (b) POSE+INCSC versus CARD-Seq+INCSC.

In the summary tables, #OPT, #INF, and #TO denote the numbers of instances with the corresponding final statuses, and total time denotes the cumulative solving time over all benchmark instances, including timeout runs. Using this notation, across all instances for which a feasible value is reported, POSE+INCSC, DSE+INCSC, and CARD-Seq+INCSC return the same objective value. The main difference is therefore not the reported value, but the ability to certify it. POSE+INCSC certifies all 35 instances, whereas DSE+INCSC and CARD-Seq+INCSC each certify 26 instances and time out on 9 instances. On those 9 timeout instances, POSE+INCSC proves optimality within the time limit.

Fig. 4(a) compares POSE+INCSC with DSE+INCSC, while Fig. 4(b) compares POSE+INCSC with CARD-Seq+INCSC. Points below the diagonal indicate that POSE+INCSC is faster. When comparing solving time, most infeasible instances in the reported results should be interpreted separately because they are detected during domain preprocessing. Therefore, the run-time differences among the encodings are more informative on feasible instances. In addition to certifying 9 more instances as optimal, as discussed above, POSE+INCSC is faster than both DSE+INCSC and CARD-Seq+INCSC on several instances whose solving time exceeds one second. For sub-second instances, this advantage is less clear, and CARD-Seq+INCSC is faster in several cases. However, these points remain close to the  $y = x$  line, indicating that the absolute differences are small. This trend is also reflected in Table V, where the total time is 599.94 seconds for POSE+INCSC, compared with 5723.34 seconds for DSE+INCSC and 5786.33 seconds for CARD-Seq+INCSC. This gap should be interpreted under the reported timeout setting, because the totals include 9 timeout runs for DSE+INCSC and CARD-Seq+INCSC. The detailed per-instance results in Table VII show larger speedups over CARD-Seq+INCSC on instances such as G03, G04, G14, T2.1, T2.3, T9.1, and T9.3.

#### D. Evaluation of the Objective Function Optimization

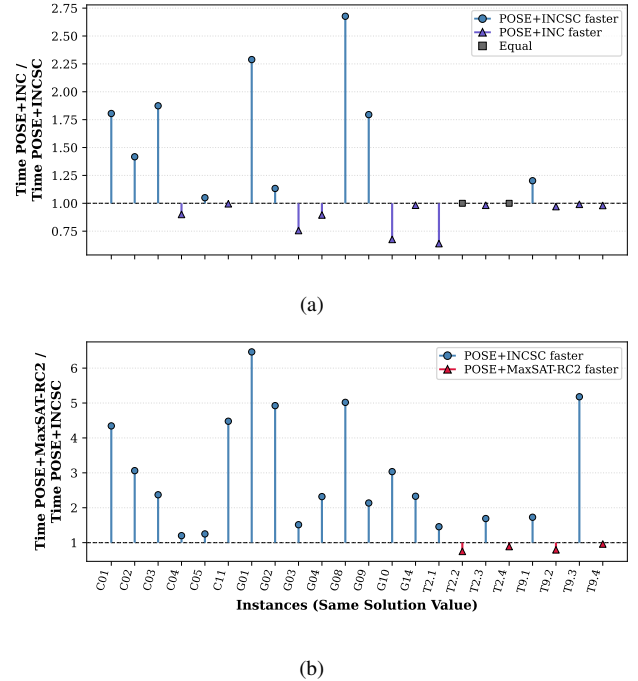


Fig. 5. Comparison of solving-time ratios on 22 feasible instances: (a) POSE+INC relative to POSE+INCSC; (b) POSE+MaxSAT-RC2 relative to POSE+INCSC.

This experiment compares INCSC with INC and MaxSAT-RC2 for objective-function optimization while keeping the POSE encoding and domain preprocessing fixed. All three configurations certify the same objective value on the 22 feasible instances. Therefore, Fig. 5(a) shows the time ratio of POSE+INC relative to POSE+INCSC, while Fig. 5(b) shows that of POSE+MaxSAT-RC2 relative to POSE+INCSC. Relative to INC, the largest reductions occur on C01, C03, G01, G08, and G09, where INCSC reduces solving time by 44% to

63%. Relative to MaxSAT-RC2, INCSC is faster on 18 of the 22 feasible instances. Under the reported protocol for all 35 benchmark instances, the cumulative solving times are 599.94 seconds for INCSC, 1015.19 seconds for INC, and 2107.76 seconds for MaxSAT-RC2, corresponding to reductions of 40.90% and 71.54%, respectively.

### E. Comparison with Exact Solver Baselines

Finally, POSE+INCSC is compared with exact solver baselines, including Gurobi, CPLEX-CP, CPLEX-MIP, and CP-SAT. Table VIII reports the detailed per-instance results and Best Known Solution (BKS) values for these exact solver baselines, while Figs. 6 and 7 highlight the differences in solving time and solution quality.

TABLE VI

SUMMARY OF EXACT SOLVER BASELINE PERFORMANCE AFTER PREPROCESSING.

| Method     | #OPT | #INF | #TO | #Certified | Total time (s) |
|------------|------|------|-----|------------|----------------|
| POSE+INCSC | 22   | 13   | 0   | 35/35      | <b>599.94</b>  |
| CP-SAT     | 20   | 13   | 2   | 33/35      | 2228.96        |
| Gurobi     | 14   | 13   | 8   | 27/35      | 5787.40        |
| CPLEX-MIP  | 10   | 13   | 12  | 23/35      | 7579.80        |
| CPLEX-CP   | 10   | 13   | 12  | 23/35      | 8100.18        |

As summarized in Table VI, POSE+INCSC solves and certifies all 35 instances within the 600-second limit, with the longest runtime of approximately 160 seconds (G09). Among the baselines, CP-SAT certifies 33 instances and times out on 2, Gurobi certifies 27 instances and times out on 8, whereas CPLEX-CP and CPLEX-MIP each certify 23 instances and time out on 12. This gap is also visible in Fig. 6, where the curve of POSE+INCSC extends further to the right than those of the baseline solvers. As shown in Table VIII, for the 22 feasible instances, the objective values certified by POSE+INCSC are equal to the BKS values reported in [7], and POSE+INCSC further proves their optimality. The remaining 13 instances are proven infeasible.

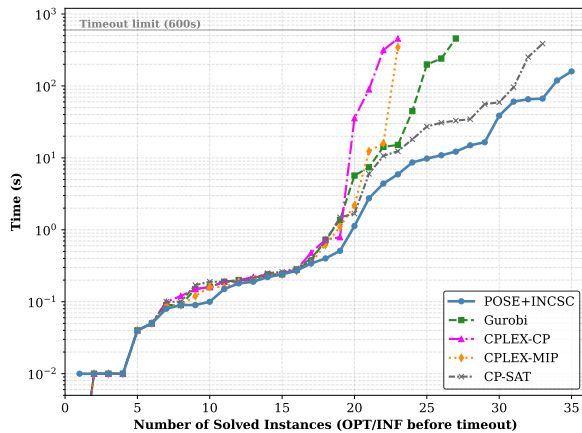


Fig. 6. Cactus plot comparing the solving time of POSE+INCSC, Gurobi, CPLEX-CP, CPLEX-MIP, and CP-SAT on the 35 benchmark instances.

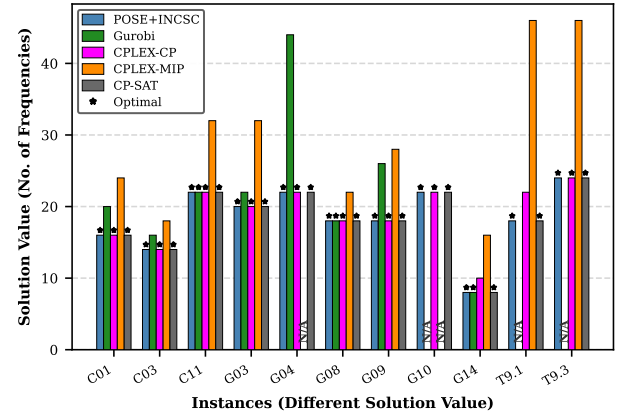


Fig. 7. Solution quality comparison on selected hard instances where at least one baseline solver exceeds the time limit or reports a different solution value. Lower values indicate better solutions.

In terms of execution efficiency, POSE+INCSC requires 599.94 seconds of cumulative solving time across all 35 instances, counting each timeout run as 600 seconds. The corresponding values are 2228.96 seconds for CP-SAT, 5787.40 seconds for Gurobi, 7579.80 seconds for CPLEX-MIP, and 8100.18 seconds for CPLEX-CP. On difficult cases such as T9.1 and T9.3, POSE+INCSC proves optimality in 10.87s and 4.40s, whereas Gurobi, CPLEX-CP, and CPLEX-MIP hit the 600-second limit. CP-SAT is more competitive on these cases but still does not certify all satisfiable instances. On infeasible instances, POSE+INCSC is typically below 0.3s.

The difference in reported solution quality is most visible on the hard instances where the baseline solvers do not certify optimality. CP-SAT often reports the same solution value as POSE+INCSC, but it still times out on two feasible instances. CPLEX-CP reports the optimal value on several timeout instances, such as C01, C03, and T9.3, but does not certify optimality within the time limit. In contrast, CPLEX-MIP and Gurobi often report weaker solution values or no feasible solution before timeout. For example, on T9.1, CPLEX-MIP reports 46 while the certified optimum is 18, and Gurobi reports no feasible solution on instances such as G10, T9.1, and T9.3.

## V. CONCLUSION

This paper presents an exact SAT-based framework for solving and certifying optimality in the MO-FAP. The proposed approach is built upon two main components. First, the order-based SAT encoding (POSE) uses standard order variables together with a problem-specific interval-boundary formulation to replace many pairwise interference conflicts of the direct SAT encoding. Second, the incremental SAT solving strategy (INCSC), based on Sequential Counter encoding, enables efficient reuse of learned clauses across iterations without reconstructing the formula. Empirical evaluations on the 35 tested benchmark instances show that the proposed SAT-based framework certifies optimality or infeasibility for all instances within a 600-second time limit. Compared with the direct SAT encoding, POSE+INCSC reduces the cumulative solving time from 5723.34 seconds to 599.94 seconds

TABLE VII  
DETAILED COMPARISON OF THE MAIN SAT CONFIGURATIONS WITH DOMAIN PREPROCESSING.

| Inst.      | POSE+INCSC |               |        | POSE+INC |              |        | POSE+MaxSAT-RC2 |             |        | POSE+EvalMaxSAT |               |        | DSE+INCSC |             |        | CARD-Seq+INCSC |               |        | CARD-Totalizer+INCSC |               |        | CARD-CardNet+INCSC |               |        |
|------------|------------|---------------|--------|----------|--------------|--------|-----------------|-------------|--------|-----------------|---------------|--------|-----------|-------------|--------|----------------|---------------|--------|----------------------|---------------|--------|--------------------|---------------|--------|
|            | Sol.       | Time          | Status | Sol.     | Time         | Status | Sol.            | Time        | Status | Sol.            | Time          | Status | Sol.      | Time        | Status | Sol.           | Time          | Status | Sol.                 | Time          | Status | Sol.               | Time          | Status |
| C01        | 16         | <b>119.14</b> | OPT    | 16       | 215.06       | OPT    | 16              | 517.67      | OPT    | 16              | 483.31        | OPT    | 16        | 600         | TO     | 16             | 600           | TO     | 16                   | 600           | TO     | 16                 | 600           | TO     |
| C02        | 14         | <b>9.78</b>   | OPT    | 14       | 13.86        | OPT    | 14              | 29.95       | OPT    | 14              | 99.96         | OPT    | 14        | 600         | TO     | 14             | 600           | TO     | 14                   | 600           | TO     | 14                 | 600           | TO     |
| C03        | 14         | <b>65.31</b>  | OPT    | 14       | 122.41       | OPT    | 14              | 154.87      | OPT    | -               | 600           | TO     | 14        | 600         | TO     | 14             | 600           | TO     | 14                   | 600           | TO     | 14                 | 600           | TO     |
| C04        | 46         | 0.10          | OPT    | 46       | 0.09         | OPT    | 46              | 0.12        | OPT    | 46              | 0.11          | OPT    | 46        | <b>0.06</b> | OPT    | 46             | 0.08          | OPT    | 46                   | 0.10          | OPT    | 46                 | 0.09          | OPT    |
| C05        | 40         | 0.40          | OPT    | 40       | 0.42         | OPT    | 40              | 0.50        | OPT    | 40              | <b>0.37</b>   | OPT    | 40        | 0.79        | OPT    | 40             | 1.18          | OPT    | 40                   | 1.18          | OPT    | 40                 | 0.93          | OPT    |
| C06        | -          | 0.22          | INF    | -        | 0.22         | INF    | -               | 0.22        | INF    | -               | <b>0.17</b>   | INF    | -         | 0.49        | INF    | -              | 0.18          | INF    | -                    | 0.18          | INF    | -                  | 0.18          | INF    |
| C07        | -          | 0.19          | INF    | -        | 0.20         | INF    | -               | 0.20        | INF    | -               | <b>0.16</b>   | INF    | -         | 0.72        | INF    | -              | <b>0.16</b>   | INF    | -                    | <b>0.16</b>   | INF    | -                  | <b>0.16</b>   | INF    |
| C08        | -          | 0.27          | INF    | -        | 0.27         | INF    | -               | 0.26        | INF    | -               | <b>0.21</b>   | INF    | -         | 1.73        | INF    | -              | 0.22          | INF    | -                    | 0.22          | INF    | -                  | 0.22          | INF    |
| C09        | -          | <b>0.01</b>   | INF    | -        | <b>0.01</b>  | INF    | -               | 0.06        | INF    | -               | <b>0.01</b>   | INF    | -         | 0.02        | INF    | -              | <b>0.01</b>   | INF    | -                    | <b>0.01</b>   | INF    | -                  | <b>0.01</b>   | INF    |
| C10        | -          | <b>0.01</b>   | INF    | -        | <b>0.01</b>  | INF    | -               | 0.06        | INF    | -               | <b>0.01</b>   | INF    | -         | <b>0.01</b> | INF    | -              | <b>0.01</b>   | INF    | -                    | <b>0.01</b>   | INF    | -                  | <b>0.01</b>   | INF    |
| C11        | 22         | 16.49         | OPT    | 22       | <b>16.42</b> | OPT    | 22              | 73.85       | OPT    | 22              | 94.04         | OPT    | 22        | 600         | TO     | 22             | 600           | TO     | 22                   | 600           | TO     | 22                 | 600           | TO     |
| G01        | 18         | <b>66.73</b>  | OPT    | 18       | 152.72       | OPT    | 18              | 431.42      | OPT    | -               | 600           | TO     | 18        | 600         | TO     | 18             | 600           | TO     | 18                   | 600           | TO     | 18                 | 600           | TO     |
| G02        | 14         | <b>14.96</b>  | OPT    | 14       | 16.95        | OPT    | 14              | 73.68       | OPT    | 14              | 122.89        | OPT    | 14        | 600         | TO     | 14             | 600           | TO     | 14                   | 600           | TO     | 14                 | 600           | TO     |
| G03        | 20         | 5.90          | OPT    | 20       | <b>4.46</b>  | OPT    | 20              | 8.93        | OPT    | 20              | 53.55         | OPT    | 20        | 24.10       | OPT    | 20             | 56.14         | OPT    | 20                   | 118.41        | OPT    | 20                 | 64.51         | OPT    |
| G04        | 22         | 12.21         | OPT    | 22       | <b>10.93</b> | OPT    | 22              | 28.31       | OPT    | 22              | 49.22         | OPT    | 22        | 58.64       | OPT    | 22             | 149.20        | OPT    | 22                   | 241.80        | OPT    | 22                 | 164.72        | OPT    |
| G05        | -          | 0.04          | INF    | -        | 0.04         | INF    | -               | 0.07        | INF    | -               | 0.04          | INF    | -         | 0.09        | INF    | -              | <b>0.03</b>   | INF    | -                    | <b>0.03</b>   | INF    | -                  | <b>0.03</b>   | INF    |
| G06        | -          | 0.09          | INF    | -        | 0.09         | INF    | -               | 0.11        | INF    | -               | <b>0.07</b>   | INF    | -         | 0.49        | INF    | -              | <b>0.07</b>   | INF    | -                    | <b>0.07</b>   | INF    | -                  | <b>0.07</b>   | INF    |
| G07        | -          | 0.01          | INF    | -        | 0.01         | INF    | -               | 0.001       | INF    | -               | <b>0.0003</b> | INF    | -         | 0.01        | INF    | -              | <b>0.0003</b> | INF    | -                    | <b>0.0003</b> | INF    | -                  | <b>0.0003</b> | INF    |
| G08        | 18         | <b>38.64</b>  | OPT    | 18       | 103.44       | OPT    | 18              | 193.93      | OPT    | 18              | 83.03         | OPT    | 18        | 600         | TO     | 18             | 600           | TO     | 18                   | 600           | TO     | 18                 | 600           | TO     |
| G09        | 18         | <b>159.46</b> | OPT    | 18       | 286.23       | OPT    | 18              | 340.59      | OPT    | -               | 600           | TO     | 18        | 600         | TO     | 18             | 600           | TO     | 18                   | 600           | TO     | 18                 | 600           | TO     |
| G10        | 22         | 60.55         | OPT    | 22       | <b>40.94</b> | OPT    | 22              | 183.74      | OPT    | 22              | 82.87         | OPT    | 22        | 600         | TO     | 22             | 600           | TO     | 22                   | 600           | TO     | 22                 | 600           | TO     |
| G11        | -          | 0.15          | INF    | -        | 0.15         | INF    | -               | 0.17        | INF    | -               | 0.13          | INF    | -         | 0.98        | INF    | -              | <b>0.12</b>   | INF    | -                    | <b>0.12</b>   | INF    | -                  | <b>0.12</b>   | INF    |
| G12        | -          | <b>0.01</b>   | INF    | -        | <b>0.01</b>  | INF    | -               | 0.05        | INF    | -               | <b>0.01</b>   | INF    | -         | 0.02        | INF    | -              | <b>0.01</b>   | INF    | -                    | <b>0.01</b>   | INF    | -                  | <b>0.01</b>   | INF    |
| G13        | -          | 0.24          | INF    | -        | 0.24         | INF    | -               | 0.24        | INF    | -               | 0.20          | INF    | -         | 1.45        | INF    | -              | <b>0.19</b>   | INF    | -                    | <b>0.19</b>   | INF    | -                  | <b>0.19</b>   | INF    |
| G14        | 8          | 8.64          | OPT    | 8        | <b>8.49</b>  | OPT    | 8               | 20.12       | OPT    | 8               | 34.79         | OPT    | 8         | 106.29      | OPT    | 8              | 36.85         | OPT    | 8                    | 52.42         | OPT    | 8                  | 33.39         | OPT    |
| T2.1       | 12         | 2.74          | OPT    | 12       | <b>1.75</b>  | OPT    | 12              | 3.99        | OPT    | 12              | 16.10         | OPT    | 12        | 42.02       | OPT    | 12             | 37.41         | OPT    | 12                   | 29.43         | OPT    | 12                 | 46.56         | OPT    |
| T2.2       | 46         | 0.08          | OPT    | 46       | 0.08         | OPT    | 46              | <b>0.06</b> | OPT    | 46              | 0.07          | OPT    | 46        | 0.25        | OPT    | 46             | <b>0.06</b>   | OPT    | 46                   | <b>0.06</b>   | OPT    | 46                 | 0.07          | OPT    |
| T2.3       | 14         | 1.13          | OPT    | 14       | <b>1.11</b>  | OPT    | 14              | 1.91        | OPT    | 14              | 10.81         | OPT    | 14        | 8           | OPT    | 14             | 4.98          | OPT    | 14                   | 8.20          | OPT    | 14                 | 8.01          | OPT    |
| T2.4       | 20         | 0.09          | OPT    | 20       | 0.09         | OPT    | 20              | <b>0.08</b> | OPT    | 20              | <b>0.08</b>   | OPT    | 20        | 0.25        | OPT    | 20             | 0.09          | OPT    | 20                   | 0.12          | OPT    | 20                 | 0.10          | OPT    |
| T2.5       | -          | 0.05          | INF    | -        | 0.05         | INF    | -               | 0.08        | INF    | -               | <b>0.04</b>   | INF    | -         | 0.29        | INF    | -              | <b>0.04</b>   | INF    | -                    | <b>0.04</b>   | INF    | -                  | <b>0.04</b>   | INF    |
| T9.1       | 18         | <b>10.87</b>  | OPT    | 18       | 13.07        | OPT    | 18              | 18.78       | OPT    | 18              | 38.66         | OPT    | 18        | 58.76       | OPT    | 18             | 76.92         | OPT    | 18                   | 110.03        | OPT    | 18                 | 67.73         | OPT    |
| T9.2       | 48         | 0.34          | OPT    | 48       | 0.33         | OPT    | 48              | <b>0.27</b> | OPT    | 48              | 0.29          | OPT    | 48        | 0.85        | OPT    | 48             | <b>0.27</b>   | OPT    | 48                   | 0.28          | OPT    | 48                 | 0.28          | OPT    |
| T9.3       | 24         | 4.40          | OPT    | 24       | <b>4.36</b>  | OPT    | 24              | 22.79       | OPT    | 24              | 21.78         | OPT    | 24        | 15.68       | OPT    | 24             | 22.14         | OPT    | 24                   | 51.12         | OPT    | 24                 | 30.80         | OPT    |
| T9.4       | 26         | 0.51          | OPT    | 26       | 0.50         | OPT    | 26              | 0.49        | OPT    | 26              | <b>0.44</b>   | OPT    | 26        | 1.08        | OPT    | 26             | <b>0.44</b>   | OPT    | 26                   | 0.47          | OPT    | 26                 | 0.47          | OPT    |
| T9.5       | -          | 0.18          | INF    | -        | 0.18         | INF    | -               | 0.19        | INF    | -               | <b>0.14</b>   | INF    | -         | 0.24        | INF    | -              | <b>0.14</b>   | INF    | -                    | 0.15          | INF    | -                  | 0.15          | INF    |
| Total Time | -          | <b>599.94</b> | -      | -        | 1015.19      | -      | -               | 2107.76     | -      | -               | 2993.56       | -      | -         | 5723.31     | -      | -              | 5786.33       | -      | -                    | 6014.81       | -      | -                  | 5818.85       | -      |
| #OPT       | -          | -             | 22     | -        | -            | 22     | -               | 22          | -      | -               | 19            | -      | -         | 13          | -      | -              | 13            | -      | -                    | 13            | -      | -                  | 13            | -      |
| #INF       | -          | -             | 13     | -        | -            | 13     | -               | 13          | -      | -               | 13            | -      | -         | 13          | -      | -              | 13            | -      | -                    | 13            | -      | -                  | 13            | -      |
| #TO        | -          | -             | 0      | -        | -            | 0      | -               | 0           | -      | -               | 3             | -      | -         | 9           | -      | -              | 9             | -      | -                    | 9             | -      | -                  | 9             | -      |

OPT: optimal, INF: infeasible, TO: timeout under the 600-second limit.

Time values are reported in seconds. Bold values indicate the fastest non-timeout method for each instance.

under the reported timeout-counting protocol. Furthermore, on this benchmark set and under the reported settings, the proposed framework certifies more instances and achieves lower cumulative runtime than the considered exact solver baselines. These results support the effectiveness of the proposed SAT-based architecture as an exact method for the tested MO-FAP benchmarks. Future work will extend the proposed framework to other FAP variants and real-world MO-FAP datasets. We also plan to investigate hybrid approaches that combine SAT-based exact solving with metaheuristic methods to improve scalability on large-scale communication networks.

#### DATA AND CODE AVAILABILITY

The source code, benchmark data, experimental results, and solver logs used in this study are publicly available at <https://github.com/dev2points/Min-Order-Frequency-Assignment-Problem>.

#### REFERENCES

- [1] R. A. Murphey, P. M. Pardalos, and M. G. C. Resende, *Frequency assignment problems*. Boston, MA: Springer US, 1999, pp. 295–377, doi: 10.1007/978-1-4757-3023-4\_6.
- [2] K. I. Aardal, S. P. M. van Hoese, A. M. C. A. Koster, C. Mannino, and A. Sassano, “Models and solution techniques for frequency assignment problems,” *Annals of Operations Research*, vol. 153, no. 1, pp. 79–129, 2007, doi: 10.1007/s10479-007-0178-0.
- [3] A. Mahmoudi, Y. Mahmoudi, K. Benatchba, A. Rezgui, and N. Zioui, “Towards optimal frequency plans: A survey of frequency assignment strategies, models, and methods,” *ENP Engineering Science Journal*, vol. 3, pp. 41–54, Dec. 2023, doi: 10.53907/enpesj.v3i2.249.
- [4] S. Mondal, M. P. Dutta, and S. K. Chakraborty, “Dynamic spectrum management and spectrum sharing techniques in 5G networks: A survey,” *Journal of Mobile Multimedia*, vol. 20, no. 5, pp. 997–1038, 2024, doi: 10.13052/jmm1550-4646.2052.
- [5] D. Cuellar, M. Sallal, and C. Williams, “BSM-6G: Blockchain-based dynamic spectrum management for 6G networks: Addressing interoperability and scalability,” *IEEE Access*, vol. 12, pp. 59 643–59 664, 2024, doi: 10.1109/ACCESS.2024.3393288.
- [6] N. Alhussien and T. A. Gulliver, “Toward AI-enabled green 6G networks: A resource management perspective,” *IEEE Access*, vol. 12, pp. 132 972–132 995, Sep. 2024, doi: 10.1109/ACCESS.2024.3460656.
- [7] P. Gomez-Meneses and H. Ayala, “Extremal optimization applied to the minimum order frequency assignment problem,” *IEEE Latin America Transactions*, vol. 21, pp. 466–474, Mar. 2023, doi: 10.1109/TLA.2023.10068851.
- [8] S. Ceschia, L. Di Gasparo, R. M. Rosati, and A. Schaerf, “Multi-neighborhood simulated annealing for the minimum interference frequency assignment problem,” *EURO Journal on Computational Optimization*, vol. 10, p. 100024, 2022, doi: 10.1016/j.ejco.2021.100024.
- [9] K.-H. Lee, H.-W. Lee, and B. C. Jung, “Minimum span frequency allocation problem: A vector bin packing approach,” *AIMS Mathematics*, vol. 10, pp. 13 278–13 295, Jan. 2025, doi: 10.3934/math.2025595.
- [10] M. Dib, A. Caminada, and H. Mabel, “A fast algorithm to solve the frequency assignment problem,” in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, ser. Lecture Notes in Computer Science, vol. 5547. Springer, May 2009, pp. 316–318, doi: 10.1007/978-3-642-01929-6\_26.
- [11] K. Alrajhi and W. Padungwech, “A dynamic tabu search approach for solving the static frequency assignment problem,” in *Advances in Computational Intelligence Systems*, ser. Advances in Intelligent Systems and Computing, vol. 513. Springer, Jan. 2017, pp. 59–70, doi: 10.1007/978-3-319-46562-3\_5.
- [12] K. Alrajhi, J. Thompson, and W. Padungwech, “Tabu search hybridized with multiple neighborhood structures for the frequency assignment problem,” in *Hybrid Metaheuristics*, ser. Lecture Notes in Computer Science, vol. 9668. Springer, Jun. 2016, pp. 157–170, doi: 10.1007/97

TABLE VIII  
DETAILED COMPARISON OF POSE+INCSC, BKS VALUES, AND EXACT SOLVER BASELINES WITH DOMAIN  
PREPROCESSING

| Inst.      | BKS POSE+INCSC |      |        |      | Gurobi |         | CPLEX/MIP |      | CPLEX/CP |      | CP-SAT |         |     |
|------------|----------------|------|--------|------|--------|---------|-----------|------|----------|------|--------|---------|-----|
|            | Sol.           | Time | Status | Sol. | Time   | Status  | Sol.      | Time | Status   | Sol. | Time   | Status  |     |
| C01        | 16             | 16   | 119.14 | OPT  | 20     | 600     | TO        | 24   | 600      | TO   | 16     | 600     | TO  |
| C02        | 14             | 14   | 9.78   | OPT  | 14     | 7.39    | OPT       | 14   | 600      | TO   | 14     | 600     | TO  |
| C03        | 14             | 14   | 65.31  | OPT  | 16     | 600     | TO        | 18   | 600      | TO   | 14     | 600     | TO  |
| C04        | 46             | 46   | 0.10   | OPT  | 46     | 0.16    | OPT       | 46   | 0.12     | OPT  | 46     | 0.22    | OPT |
| C05        | 40             | 40   | 0.40   | OPT  | 40     | 1.37    | OPT       | 40   | 1.10     | OPT  | 40     | 0.73    | OPT |
| C06        | -              | -    | 0.22   | INF  | -      | 0.24    | INF       | -    | 0.23     | INF  | -      | 0.24    | INF |
| C07        | -              | -    | 0.19   | INF  | -      | 0.20    | INF       | -    | 0.20     | INF  | -      | 0.20    | INF |
| C08        | -              | -    | 0.27   | INF  | -      | 0.28    | INF       | -    | 0.27     | INF  | -      | 0.28    | INF |
| C09        | -              | -    | 0.01   | INF  | -      | 0.01    | INF       | -    | 0.01     | INF  | -      | 0.01    | INF |
| C10        | -              | -    | 0.01   | INF  | -      | 0.01    | INF       | -    | 0.01     | INF  | -      | 0.01    | INF |
| C11        | 22             | 22   | 16.49  | OPT  | 22     | 198.57  | OPT       | 32   | 600      | TO   | 22     | 600     | TO  |
| G01        | 18             | 18   | 66.73  | OPT  | 18     | 5.68    | OPT       | 18   | 2.19     | OPT  | 18     | 600     | TO  |
| G02        | 14             | 14   | 14.96  | OPT  | 14     | 44.70   | OPT       | 14   | 15.97    | OPT  | 14     | 600     | TO  |
| G03        | 20             | 20   | 5.90   | OPT  | 22     | 600     | TO        | 32   | 600      | TO   | 20     | 89.41   | OPT |
| G04        | 22             | 22   | 12.21  | OPT  | 44     | 600     | TO        | -    | 600      | TO   | 22     | 315.61  | OPT |
| G05        | -              | -    | 0.04   | INF  | -      | 0.04    | INF       | -    | 0.04     | INF  | -      | 0.04    | INF |
| G06        | -              | -    | 0.09   | INF  | -      | 0.09    | INF       | -    | 0.09     | INF  | -      | 0.10    | INF |
| G07        | -              | -    | 0.01   | INF  | -      | 0.0003  | INF       | -    | 0.0003   | INF  | -      | 0.0003  | INF |
| G08        | 18             | 18   | 38.64  | OPT  | 18     | 240.57  | OPT       | 22   | 600      | TO   | 18     | 600     | TO  |
| G09        | 18             | 18   | 159.46 | OPT  | 26     | 600     | TO        | 28   | 600      | TO   | 18     | 600     | TO  |
| G10        | 22             | 22   | 60.55  | OPT  | -      | 600     | TO        | -    | 600      | TO   | 22     | 600     | TO  |
| G11        | -              | -    | 0.15   | INF  | -      | 0.15    | INF       | -    | 0.16     | INF  | -      | 0.15    | INF |
| G12        | -              | -    | 0.01   | INF  | -      | 0.01    | INF       | -    | 0.01     | INF  | -      | 0.01    | INF |
| G13        | -              | -    | 0.24   | INF  | -      | 0.24    | INF       | -    | 0.24     | INF  | -      | 0.25    | INF |
| G14        | 8              | 8    | 8.64   | OPT  | 8      | 456.71  | OPT       | 16   | 600      | TO   | 10     | 600     | TO  |
| T2.1       | 12             | 12   | 2.74   | OPT  | 12     | 14.22   | OPT       | 12   | 12.34    | OPT  | 12     | 600     | TO  |
| T2.2       | 46             | 46   | 0.08   | OPT  | 46     | 0.09    | OPT       | 46   | 0.09     | OPT  | 46     | 0.12    | OPT |
| T2.3       | 14             | 14   | 1.13   | OPT  | 14     | 15.13   | OPT       | 14   | 345.36   | OPT  | 14     | 455.48  | OPT |
| T2.4       | 20             | 20   | 0.09   | OPT  | 20     | 0.20    | OPT       | 20   | 0.16     | OPT  | 20     | 0.16    | OPT |
| T2.5       | -              | -    | 0.05   | INF  | -      | 0.05    | INF       | -    | 0.05     | INF  | -      | 0.05    | INF |
| T9.1       | 18             | 18   | 10.87  | OPT  | -      | 600     | TO        | 46   | 600      | TO   | 22     | 600     | TO  |
| T9.2       | 48             | 48   | 0.34   | OPT  | 48     | 0.37    | OPT       | 48   | 0.35     | OPT  | 48     | 0.48    | OPT |
| T9.3       | 24             | 24   | 4.40   | OPT  | -      | 600     | TO        | 46   | 600      | TO   | 24     | 35.65   | OPT |
| T9.4       | 26             | 26   | 0.51   | OPT  | 26     | 0.73    | OPT       | 26   | 0.62     | OPT  | 26     | 0.79    | OPT |
| T9.5       | -              | -    | 0.18   | INF  | -      | 0.19    | INF       | -    | 0.19     | INF  | -      | 0.19    | INF |
| Total Time | -              | -    | 599.94 | -    | -      | 5787.40 | -         | -    | 7579.80  | -    | -      | 8100.18 | -   |
| #OPT       | -              | -    | -      | 22   | -      | -       | 14        | -    | -        | 10   | -      | -       | 20  |
| #INF       | -              | -    | -      | 13   | -      | -       | 13        | -    | -        | 13   | -      | -       | 13  |
| #TO        | -              | -    | -      | 0    | -      | -       | 8         | -    | -        | 12   | -      | -       | 2   |

BKS: Best Known Solution value reported in [7].

OPT: optimal, INF: infeasible, TO: timeout under the 600-second limit.

Time values are reported in seconds. Bold values indicate the fastest non-timeout method for each instance.

8-3-319-39636-1\_12.

- [13] K. Alrajhi, "Ant metaheuristic algorithm for the frequency assignment problem," *International Journal of Computational and Applied Mathematics & Computer Science*, vol. 3, pp. 98–103, 11 2023, doi: 10.37394/232028.2023.3.11.
- [14] K. Alrajhi, "A dynamic ant colony optimization for solving the static frequency assignment problem," *International Journal of Applied Sciences & Development*, vol. 2, pp. 169–176, 2023, doi: 10.37394/23202 9.2023.2.18.
- [15] K. Alrajhi, "A dynamic hyper heuristic approach for solving the static frequency assignment problem," in *Advances in Computational Intelligence Systems*, ser. Advances in Intelligent Systems and Computing (AISC), vol. 1453. Springer Nature Switzerland, 2024, pp. 336–347, doi: 10.1007/978-3-031-47508-5\_27.
- [16] K. Alrajhi, "Dynamic multiple neighborhood structures for the static frequency assignment problem," *International Journal of Applied Mathematics, Computational Science and Systems Engineering*, vol. 5, pp. 185–193, Nov. 2023, doi: 10.37394/232026.2023.5.17.
- [17] C. Sinz, "Towards an optimal CNF encoding of boolean cardinality constraints," in *Principles and Practice of Constraint Programming - CP 2005*, P. van Beek, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 827–831, doi: 10.1007/11564751\_73.
- [18] I. Baybars, "Optimal assignment of broadcasting frequencies," *European Journal of Operational Research*, vol. 9, no. 3, pp. 257–263, 1982, doi: 10.1016/0377-2217(82)90033-9.
- [19] B. Jansen, K. Aardal, C. P. M. van Hoesel, and A. Hipolito, "A branch-and-cut algorithm for the frequency assignment problem," METEOR, Maastricht University School of Business and Economics, Tech. Rep. 011, Jan. 1996, doi: 10.26481/umamet.1996011.
- [20] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, and E. Toto, "Frequency assignment in mobile radio systems using branch-and-cut techniques," *European Journal of Operational Research*, vol. 123, no. 2, pp. 241–255, 2000, doi: 10.1016/S0377-2217(99)00254-4.
- [21] A. M. C. A. Koster, S. P. M. van Hoesel, and A. W. J. Kolen, "The partial constraint satisfaction problem: Facets and lifting theorems," *Operations Research Letters*, vol. 23, no. 3-5, pp. 89–97, 1998, doi: 10.1016/S016 7-6377(98)00043-1.
- [22] C. Mannino and A. Sassano, "An enumerative algorithm for the frequency assignment problem," *Discrete Applied Mathematics*, vol. 129, no. 1, pp. 155–169, 2003, doi: 10.1016/S0166-218X(02)00239-1.
- [23] D. Král', "An exact algorithm for the channel assignment problem," *Discrete Applied Mathematics*, vol. 145, no. 2, pp. 326–331, 2005, doi: 10.1016/j.dam.2004.01.020.
- [24] G. S. Uyanik, M. J. Abdel-Rahman, and M. Krunz, "Optimal channel assignment with aggregation in multi-channel systems: A resilient approach to adjacent-channel interference," *Ad Hoc Networks*, vol. 20, pp. 64–76, 2014, doi: 10.1016/j.adhoc.2014.03.007.
- [25] J. K. Fichte, D. L. Berre, M. Hecher, and S. Szeider, "The silent (r) evolution of sat," *Communications of the ACM*, vol. 66, no. 6, pp. 64–72, 2023, doi: 10.1145/3560469.
- [26] G. Glorian, J.-M. Lagniez, V. Montmirail, and N. Szczechanski, "An incremental sat-based approach to the graph colouring problem," in *International Conference on Principles and Practice of Constraint Programming*. Springer, 2019, pp. 213–231, doi: 10.1007/978-3-0

- 30-30048-7\_13.
- [27] D. Faber, A. Jabrayilov, and P. Mutzel, “Sat encoding of partial ordering models for graph coloring problems,” in *27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024, pp. 12:1–12:20, doi: 10.4230/LIPIcs.SAT.2024.12.
  - [28] A. Ignatiev, A. Morgado, and J. Marques-Silva, “PySAT: A Python toolkit for prototyping with SAT oracles,” in *SAT*, 2018, pp. 428–437, doi: 10.1007/978-3-319-94144-8\_26.
  - [29] A. Ignatiev, Z. L. Tan, and C. Karamanos, “Towards universally accessible SAT technology,” in *SAT*, 2024, pp. 16:1–16:11, doi: 10.4230/LIPIcs.SAT.2024.16.
  - [30] J. M. Crawford and A. B. Baker, “Experimental results on the application of satisfiability algorithms to scheduling problems,” in *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, ser. AAAI’94. AAAI Press, 1994, pp. 1092–1097, <https://dl.acm.org/doi/10.5555/2891730.2891899>. [Online]. Available: <https://dl.acm.org/doi/10.5555/2891730.2891899>
  - [31] N. Tamura, A. Taga, S. Kitagawa, and M. Banbara, “Compiling finite linear CSP into SAT,” *Constraints*, vol. 14, no. 2, pp. 254–272, 2009, doi: 10.1007/s10601-008-9061-0.
  - [32] J. Petke and P. Jeavons, “The order encoding: From tractable CSP to tractable SAT,” in *Theory and Applications of Satisfiability Testing - SAT 2011*, vol. 6695, Jun. 2011, pp. 371–372, doi: 10.1007/978-3-642-21581-0\_34.
  - [33] I. Abío and P. J. Stuckey, “Encoding linear constraints into SAT,” in *Principles and Practice of Constraint Programming*, B. O’Sullivan, Ed. Cham: Springer International Publishing, 2014, pp. 75–91, doi: 10.1007/978-3-319-10428-7\_9.
  - [34] R. Martins, S. Joshi, V. Manquinho, and I. Lynce, “Incremental cardinality constraints for MaxSAT,” in *Principles and Practice of Constraint Programming*, B. O’Sullivan, Ed. Cham: Springer International Publishing, 2014, pp. 531–548, doi: 10.1007/978-3-319-10428-7\_39.
  - [35] O. Bailleux and Y. Bouffkhad, “Efficient cnf encoding of boolean cardinality constraints,” in *Principles and Practice of Constraint Programming – CP 2003*, vol. 2833, 09 2003, pp. 108–122, doi: 10.1007/978-3-540-45193-8\_8.



**Nghia Dao Xuan** received his B.Sc. degree in computer science from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, where he is currently pursuing the M.Sc. degree. His research interests include combinatorial optimization, SAT, and telecommunication resource allocation.



**Phuong Dang Anh** is an undergraduate student in information technology at the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam. His research interests include combinatorial optimization and SAT-based exact methods.



**Khanh To Van** received his B.Sc. and M.Sc. degrees from the University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, in 2004 and 2007, respectively. He earned the Ph.D. degree in Information Science from the Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2013, where his doctoral research focused on SAT and SMT solving. His research interests include formal methods in software engineering and the development of efficient logical encodings for combinatorial optimization and operations research problems, with a particular emphasis on SAT solving.

ations research problems, with a particular emphasis on SAT solving.