Relaxation-Free Min-k-Partition
for PCI Assignment in 5G Networks

Yeqing Qiu, Chengpiao Huang, Ye Xue, Zhipeng Jiang, Qingjiang Shi, Dong Zhang, and Zhi-Quan Luo

Abstract—Physical Cell Identity (PCI) is a critical parameter in 5G networks. Efficient and accurate PCI assignment is essential for mitigating mod-3 interference, mod-30 interference, collisions, and confusions among cells, which directly affect network reliability and user experience. In this paper, we propose a novel framework for PCI assignment by decomposing the problem into Min-3-Partition, Min-10-Partition, and a graph coloring problem, leveraging the Chinese Remainder Theorem (CRT). Furthermore, we develop a relaxation-free approach to the general Min-k-Partition problem by reformulating it as a quadratic program with a norm-equality constraint and solving it using a penalized mirror descent (PMD) algorithm. The proposed method demonstrates superior computational efficiency and scalability, significantly reducing interference while eliminating collisions and confusions in large-scale 5G networks. Numerical evaluations on real-world datasets show that our approach reduces computational time by up to 20 times compared to state-of-the-art methods, making it highly practical for real-time PCI optimization in large-scale networks. These results highlight the potential of our method to improve network performance and reduce deployment costs in modern 5G systems.

Index Terms—Network Optimization, PCI Assignment, Graph Partition, Relaxation-Free, Mirror Descent

I. INTRODUCTION

The rollout of the fifth-generation mobile networks has shown considerable promise to revolutionize the telecommunications industry, offering unprecedented speed, minimal latency, and extensive connectivity [1–3]. Despite the initial excitement surrounding the deployment of 5G in 2019 [4],

(Corresponding author: Ye Xue, Qingjiang Shi)
Yeqing Qiu is with the School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China, and Shenzhen Research Institute of Big Data, Shenzhen, China (email: yeqingqiu@link.cuhk.edu.cn).
Chengpiao Huang is with the Department of Industrial Engineering and Operations Research, Columbia University, New York, USA (email: chengpiao.huang@columbia.edu).
Ye Xue is with Shenzhen Research Institute of Big Data, Shenzhen, China, and School of Data Science, The Chinese University of Hong Kong, Shenzhen, China (email: xueye@cuhk.edu.cn).
Zhipeng Jiang is with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China (email: jiangzhipeng@ucas.ac.cn).
Qingjiang Shi is with the School of Software Engineering, Tongji University, Shanghai, China, and Shenzhen Research Institute of Big Data, Shenzhen, China (email: shiqj@tongji.edu.cn).
Dong Zhang is with Huawei Technologies, Shenzhen, China (email: zhangdong48@huawei.com)
Zhi-Quan Luo is with The Chinese University of Hong Kong, Shenzhen, China, and Shenzhen Research Institute of Big Data, Shenzhen, China (email: luozq@cuhk.edu.cn).

certain regions have witnessed a decline in 5G speeds and user satisfaction [5]. This decline is attributed to the fact that the real-world wireless network performance depends not only on hardware capabilities, such as base stations and mobile devices but also on the configuration of network parameters [6, 7]. Therefore, to improve 5G network performance, it is crucial to optimize key system parameters to match the local radio propagation environment and user distribution.

In this paper, we study the configuration of an important 5G network parameter known as the *Physical Cell Identity* (PCI). It serves as a distinctive identifier for cells and enables user equipment (UE) to search for cells to connect to, identify neighbor cells for cell re-selection, and facilitate cell-to-cell handovers. A good PCI configuration can effectively reduce co-frequency interference among cells [8], and lead to better communication quality and user experience.

Specifically, the *PCI assignment* problem involves allocating a PCI value to each cell in a 5G network. According to the 5G standard, there are 1008 PCI values in total, which are integers from 0 to 1007 [9]. Since there are typically thousands of cells in a 5G network, multiple cells may share the same PCI value. There are multiple objectives and constraints in the 5G PCI assignment problem: minimizing mod-3 interference and mod-30 interference while eliminating collisions and confusions. Roughly speaking, interference, collisions, and confusions arise when certain pairs of cells share the same PCIs, or their PCIs have the same mod value. Mathematically, PCI assignment is an NP-hard constrained combinatorial optimization problem [10], which becomes increasingly challenging to solve as the density of base stations increases in the continuing build-out of 5G mobile communication networks. Specifically, the solution space size is in the order of 1008^N, where N is the number of cells in the 5G network and is usually of the scale $10^5 \sim 10^6$ in practice. The prohibitively large problem size calls for the design of efficient PCI assignment algorithms.

A. Related Work

Existing methods for PCI assignment can be mainly divided into satisfiability methods and optimization methods.

1) Satisfiability Methods: Satisfiability methods mainly focus on eliminating collisions and confusions between cells. To this end, they model the wireless network as a graph, where each vertex is a cell and each edge representing a collision or a confusion, and then perform graph coloring. Bandh et al. [11] developed a greedy graph coloring (GGC) approach. Based on the techniques from GGC, Pratap et al. [12] proposed a randomized graph coloring approach with

improved computational efficiency. However, when it comes to minimizing mod-3 and mod-30 interference, graph coloring methods are not directly applicable due to the presence of mod-3 and mod-30 operations.

2) Optimization Methods: To further address mod-3 and mod-30 interference, some heuristic algorithms have been proposed, primarily based on genetic algorithms. Li et al. [13] and Shen et al. [14] both proposed genetic algorithm-based methods for PCI assignment, incorporating mod-3 interference into their respective fitness objectives. Yang et al. [10] formulated PCI assignment as an interference self-coordination problem and proposed adaptive heuristics to minimize mod-3 interference. Andrade et al. [15] modeled modular interference with two cell neighboring relations and introduced a memetic algorithm that combines multiparent biased random-key genetic search with local refinement. However, these heuristic approaches do not have theoretical guarantees for either solution quality or computational complexity.

Beyond heuristics, several works formulate PCI assignment as a mathematical optimization problem and design algorithms to solve it. Since PCI assignment is inherently a difficult NP-hard combinatorial optimization problem, most algorithms encounter significant challenges in finding high-quality solutions. Consequently, simpler algorithms relying on greedy strategies and search methods are often employed to solve this problem, leading to suboptimal performance. Gui et al. [8] proposed a greedy algorithm to tackle the NP-hard binary quadratic programming (BQP) model for PCI assignment. Fairbrother et al. [16] modeled the PCI assignment problem as a two-level graph partitioning and presented a mixed-integer programming (MIP) model to solve it.

All existing optimization methods either incorporate weighted objectives to prioritize minimizing mod-3 interference, mod-30 interference, collisions, and confusions, or target a single arbitrary mod-k interference component. As a result, they fail to account for the structural dependence between multiple modular interferences, which often leads to a trade-off between solution quality and computational efficiency. This limitation becomes especially problematic when scaling to large networks.

B. Contributions

In this paper, we propose a novel approach for PCI assignment that jointly addresses mod-3 interference, mod-30 interference, collisions, and confusions. Our key contributions are summarized as follows.

- **Decomposition of PCI assignment into multi-stage graph partitioning.** Based on the Chinese Remainder Theorem (CRT), we break down PCI assignment into sequentially solving the Min-3-Partition, Min-10-Partition, and graph coloring problems. These sub-problems decouple the objectives and constraints in PCI assignment, significantly reducing the size of the solution space and enabling the scalability of the proposed method to large-scale networks.
- **A relaxation-free reformulation of Min-k-Partition.** To tackle the general NP-hard Min-k-Partition problem, we

propose an exact reformulation as a quadratic program with a norm-equality constraint over the probability simplex. To solve this reformulation, we develop a penalized mirror descent (PMD) algorithm, which is both efficient and theoretically grounded. We provide rigorous convergence guarantees that characterize the convergence rate in terms of key system parameters.
- **Numerical validation on real-world data.** We conduct extensive numerical experiments on real-world datasets from dense urban 5G deployments to evaluate the performance of the proposed method. The results demonstrate that our approach outperforms state-of-the-art baselines in terms of interference minimization, collision and confusion elimination, and computational efficiency. These findings highlight the practical applicability of our method in optimizing PCI assignments for large-scale, real-world 5G networks.

Synopsis. The subsequent sections of the paper are structured as follows. Section II introduces the system model and formulates the PCI assignment problem. Section III describes our approach for the PCI assignment based on the CRT, decomposing the problem into sequentially solving Min-k-Partition problems and a graph coloring problem. Building upon this decomposition, Section IV presents a relaxation-free reformulation of Min-k-Partition and details the proposed PMD algorithm along with convergence analysis. Section V extends the proposed framework to a more practical case where the PCI of a subset of cells is optimized. Section VI presents numerical experiments on synthetic and real-world data. Finally, Section VII concludes the paper.

Notations. We use $\mathbb{Z}$ and $\mathbb{R}$ to denote the set of integers and real numbers, respectively. Moreover, $\mathbb{Z}_+$ denotes the set of non-negative integers, $\mathbb{Z}_k$ is the set integers ranging from 0 to $k-1$, $\mathbb{R}^N$ is the set of N-dimensional real-valued vectors, and $\mathbb{Z}_k^N$ is the set of N-dimensional integer vectors with entries from $\mathbb{Z}_k$. For $a, b \in \mathbb{Z}$ and $k \in \mathbb{Z}_+$, we write $a \equiv b \pmod{k}$ if $a - b = nk$ for some $n \in \mathbb{Z}$. The mod k value of a, denoted by $m_k(a)$, is the unique integer $r \in \mathbb{Z}_k$ such that $a \equiv r \pmod{k}$. For $x \in \mathbb{R}$, we use $\lfloor x \rfloor$ to denote the largest integer less than or equal to x. We use lowercase and uppercase boldface letters to denote column vectors and matrices, respectively. The notation $[\boldsymbol{A}]_{i,j}$ represents the element in the i-th row and j-th column of a matrix $\boldsymbol{A}$. The transpose of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^T$. The trace of a matrix $\boldsymbol{A}$ is denoted by $\mathrm{Tr}(\boldsymbol{A})$. The ℓ_p-norm of a vector $\boldsymbol{v}$ is denoted by $\|\boldsymbol{v}\|_p$. The $\ell_{p,q}$-norm of a matrix $\boldsymbol{A}$ is given by $\|\boldsymbol{A}\|_{p,q} = \left(\sum_{i=1}^n \left(\sum_{j=1}^n |[\boldsymbol{A}]_{i,j}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}}$. The notation $\langle \cdot, \cdot \rangle$ represents the generalized inner product. The vector $\boldsymbol{1}_N$ denotes the N-dimensional all-one vector. The function $\mathbb{1}\{A\}$ serves as the indicator function for event A. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$.

II. SYSTEM MODEL AND PROBLEM FORMULATION

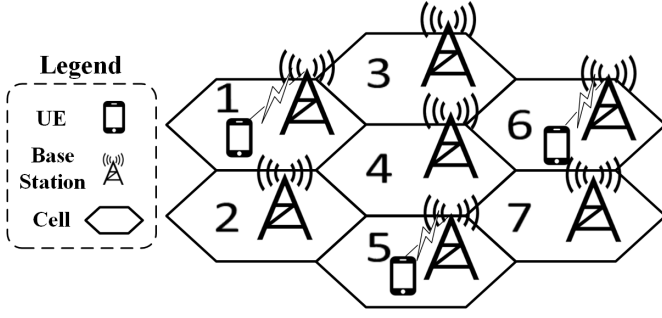In this section, we formally define the PCI assignment problem.

Fig. 1: Illustration of a 5G wireless network comprising $N = 7$ co-frequency cells. Each hexagon is a cell, and two cells are neighboring cells if they share an edge. For example, $(1, 4) \in \mathcal{E}_1$ indicates a collision between cell 1 and cell 4 when they share the same PCI. Moreover, since $(4, 1), (4, 7) \in \mathcal{E}_1$, this implies $(1, 7) \in \mathcal{E}_2$, and confusion arises when cells 1 and 7 are assigned the same PCI. When the PCIs of cells 1 and 2 share the same mod-3 value, mod-3 interference occurs with a magnitude of $[\boldsymbol{W}]_{1,2}$. Mod-30 interference follows a similar pattern.

A. System Model

We model a 5G wireless network with N cells as a graph, where the vertices are given by $\mathcal{V} = \{1, 2, \cdots, N\}$, and vertex i represents the i-th cell. Between cells, there is interference and a neighboring relation. The interference between cells is encoded in an *interference matrix* $\boldsymbol{W} \in \mathbb{R}^{N \times N}$, where $[\boldsymbol{W}]_{i,j} = [\boldsymbol{W}]_{j,i} \geq 0$ is the interference between cell i and cell j. For two cells i and j, we say that j is a *neighboring cell* of i if the signal of cell j can be received by devices connected to cell i. The set of neighboring cell pairs is denoted by $\mathcal{E}_1 \subseteq \mathcal{V} \times \mathcal{V}$. We further define the *second-order neighboring set*

$$\mathcal{E}_2 = \{(i, j) \in \mathcal{V} \times \mathcal{V} \mid \exists \ell \in \mathcal{V} \text{ such that } (\ell, i), (\ell, j) \in \mathcal{E}_1\},$$

which consists of cell pairs that share the same neighbor. We let $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. Similar to [15], we assume that $\boldsymbol{W}$, $\mathcal{E}_1$, and $\mathcal{E}_2$ have been estimated and are known.

B. Problem Formulation

The task of the 5G PCI assignment problem is to assign to each cell $i \in \mathcal{V}$ an integer $\mathrm{PCI}_i \in \mathbb{Z}_{1008}$, called the *PCI value*, with the goal of minimizing mod-3 and mod-30 interference and eliminating collisions and confusions.

- *Mod-3 interference* occurs between two cells i and j, if their PCIs have the same mod-3 value, i.e. $\mathrm{PCI}_i \equiv \mathrm{PCI}_j \pmod 3$. In this case, the magnitude of the mod-3 interference is given by $[\boldsymbol{W}]_{i,j}$. Physically, the mod-3 value is used for the scheduling of the physical downlink and uplink shared channel as well as the sounding reference signal. An overlap of mod-3 value leads to conflicts in resource scheduling [17].
- *Mod-30 interference* occurs when two cells have the same mod-30 value, i.e. $\mathrm{PCI}_i \equiv \mathrm{PCI}_j \pmod{30}$. In this case, the magnitude of the mod-30 interference is given by

$[\boldsymbol{W}]_{i,j}$. Physically, an overlap of mod-30 value indicates the reuse of the same Zadoff-Chu root sequences for uplink demodulation reference signals and sounding reference signals, leading to severe sequence collision [17].

- A *collision* arises when two neighboring cells i and j share the same PCI, that is, $(i, j) \in \mathcal{E}_1$ and $\mathrm{PCI}_i = \mathrm{PCI}_j$. It prevents a UE from determining the serving cell and establishing a connection [15].
- A *confusion* arises when two cells $(i, j) \in \mathcal{E}_2$ share the same PCI. In other words, this means that two cells i, j sharing a common neighbor have the same PCI. This can potentially cause a UE confusion during cell handovers.

In Figure 1, we provide a visualization of a 5G network and illustrate the concepts of interference, collisions, and confusions. Although the physical mechanisms that generate mod-3 and mod-30 interference differ, we represent both with the same interference matrix $\boldsymbol{W}$. In practice, PCI assignment is performed at the network layer, where the only reliable source of interference information is the MR data collected from live networks. MR data already aggregates all signal-level degradations, so a unified matrix $\boldsymbol{W}$ is aligned with common industrial practice and ensures that the optimization concentrates on measurable interference patterns rather than unobservable physical-layer particulars [18, 19].

Mathematically, the PCI assignment problem can be formulated as

$$\text{(P1)} \quad \min_{\mathbf{PCI}} \quad \left(\sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{\mathrm{PCI}_i \equiv \mathrm{PCI}_j \pmod 3\}, \right. \tag{1a}$$

$$\left. \sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{\mathrm{PCI}_i \equiv \mathrm{PCI}_j \pmod{30}\} \right), \tag{1b}$$

$$\text{s.t.} \quad \mathrm{PCI}_i \neq \mathrm{PCI}_j, \quad \forall (i, j) \in \mathcal{E}, \tag{1c}$$

$$\mathrm{PCI}_i \in \mathbb{Z}_{1008}, \quad \forall i \in \mathcal{V}, \tag{1d}$$

where $\mathbf{PCI} = (\mathrm{PCI}_1, ..., \mathrm{PCI}_N)$. This is a multi-objective optimization problem where the objectives are to minimize the mod-3 interference (1a) and the mod-30 interference (1b), and the constraints include eliminating collisions and confusions (1c) and the PCI range requirement (1d).

III. A MULTI-STAGE DECOMPOSITION APPROACH

In this section, we present our approach to PCI assignment, which decomposes the problem into multiple stages of graph partitioning and graph coloring, effectively decoupling the multiple objectives and constraints.

A. Decomposition Framework

For each cell i, we can decompose its PCI value as

$$\mathrm{PCI}_i = 30q_i + r_i,$$

where $q_i = \lfloor \mathrm{PCI}_i/30 \rfloor \in \mathbb{Z}_+$ is the quotient, and $r_i = m_{30}(\mathrm{PCI}_i) \in \mathbb{Z}_{30}$ is the mod-30 value of PCI_i. Thus, a PCI assignment $\mathbf{PCI}$ is equivalently characterized by the mod-30

value assignment $\boldsymbol{r} = (r_1, ..., r_N)$ and the quotient assignment $\boldsymbol{q} = (q_1, ..., q_N)$. The PCI assignment problem (P1) can then be reformulated as

$$(P2) \quad \min_{\boldsymbol{q}, \boldsymbol{r}} \quad \left(\sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i \equiv r_j \ (\mathrm{mod}\ 3)\}, \quad (2a) \right.$$

$$\left. \sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\} \right), \quad (2b)$$

$$\text{s.t.} \quad q_i \neq q_j \text{ or } r_i \neq r_j, \quad \forall (i,j) \in \mathcal{E},$$
$$\boldsymbol{q} \in \mathbb{Z}_+^N,$$
$$\boldsymbol{r} \in \mathbb{Z}_{30}^N,$$
$$30 \times \boldsymbol{q} + \boldsymbol{r} \in \mathbb{Z}_{1008}^N.$$

Note that the objective (2b) does not involve modulo operations because $r_i \in \mathbb{Z}_{30}$. Moreover, since both objectives (2a) and (2b) depend solely on $\boldsymbol{r}$, then we can break down the problem into first optimizing over $\boldsymbol{r}$ to minimize both mod-3 interference and mod-30 interference, and then assigning $\boldsymbol{q}$ to eliminate collisions and confusions. This leads to the following two sub-problems:

- **Sub-problem 1: Mod-**30 **value assignment** $\boldsymbol{r}$**.** We determine the mod-30 value assignment $\boldsymbol{r}$ by minimizing the mod-3 and mod-30 interference:

$$\min_{\boldsymbol{r}} \quad \left(\sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i \equiv r_j \ (\mathrm{mod}\ 3)\}, \right.$$

$$\left. \sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\} \right),$$

$$\text{s.t.} \quad \boldsymbol{r} \in \mathbb{Z}_{30}^N.$$

- **Sub-problem 2: Quotient assignment** $\boldsymbol{q}$**.** Suppose we have found a mod-30 value assignment $\boldsymbol{r}$ for sub-problem 1. Then, we need to find a quotient assignment $\boldsymbol{q}$ that eliminates collisions and confusions. This is a feasibility problem:

$$\text{find} \quad \boldsymbol{q} = (q_1, \cdots, q_N),$$
$$\text{s.t.} \quad q_i \neq q_j, \quad \forall (i,j) \in \mathcal{E} \text{ with } r_i = r_j,$$
$$\boldsymbol{q} \in \mathbb{Z}_+^N,$$
$$30\boldsymbol{q} + \boldsymbol{r} \in \mathbb{Z}_{1008}^N.$$

In the following two subsections, we will further decompose sub-problems 1 and 2 into smaller problems, which greatly reduces the problem size.

B. Two-Stage Graph Partitioning for Mod-30 Value Assignment

We observe that the mod-3 interference minimization objective in sub-problem 1 depends solely on the mod-3 values of the cells' PCIs, regardless of their specific mod-30 values. In light of this observation, we will first determine the mod-3 values of the cells' PCIs to minimize the mod-3 interference. Based on the mod-3 values, we will then decide the mod-30 values to minimize the mod-30 interference. To this end, we invoke the Chinese Remainder Theorem (CRT) [20], which

states that two mod values of a number, say the mod-n_1 and mod-n_2 values, uniquely determine its mod-$n_1 n_2$ value as long as n_1 and n_2 are coprime. Accordingly, the mod 30 value $r_i = m_{30}(\mathrm{PCI}_i)$ is uniquely determined by its mod-3 value $r_i^{(3)} = m_3(\mathrm{PCI}_i)$ and mod-10 value $r_i^{(10)} = m_{10}(\mathrm{PCI}_i)$. Therefore, we can further decompose sub-problem 1 by first finding a mod 3 value assignment $\boldsymbol{r}^{(3)} = (r_1^{(3)}, ..., r_N^{(3)})$ to minimize the mod 3 interference, and then finding a mod 10 value assignment $\boldsymbol{r}^{(10)} = (r_1^{(10)}, ..., r_N^{(10)})$ to minimize the mod 30 interference. Mathematically, this corresponds to the following two sub-problems:

- **Sub-problem 1.1: Mod-**3 **value assignment** $\boldsymbol{r}^{(3)}$**.** We determine the mod-3 value assignment $\boldsymbol{r}^{(3)}$ by minimizing the mod-3 interference:

$$(S1.1) \quad \min_{\boldsymbol{r}^{(3)}} \quad \sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i^{(3)} = r_j^{(3)}\},$$

$$\text{s.t.} \quad \boldsymbol{r}^{(3)} \in \mathbb{Z}_3^N.$$

Note that the objective is indeed the mod-3 interference minimization objective in sub-problem 1, because $r_i \equiv r_j \ (\mathrm{mod}\ 3)$ if and only if $r_i^{(3)} = r_j^{(3)}$. Sub-problem 1.1 is in fact a *Min-3-Partition* problem which is NP-hard [21–23]. We will propose an algorithm for solving it in Section IV.

- **Sub-problem 1.2: Mod-**10 **value assignment** $\boldsymbol{r}^{(10)}$**.** Suppose we have found a mod-3 value assignment $\boldsymbol{r}^{(3)}$ in sub-problem 1.1. We now need to find a mod-10 value assignment $\boldsymbol{r}^{(10)}$ that minimizes the mod-30 interference. An important observation is that if two cells are assigned different mod-3 values, then regardless of the mod-10 value assignment, they must have different mod-30 values and hence no mod-30 interference. We can take advantage of this observation to reduce the problem size. Specifically, we partition the network into three *partitions* $\mathcal{V}_0, \mathcal{V}_1, \mathcal{V}_2$ according to the mod-3 values of the cells:

$$\mathcal{V}_r = \{i \in \mathcal{V} \mid r_i^{(3)} = r\}, \quad r \in \mathbb{Z}_3.$$

Then we perform mod-10 value assignment on each partition *in parallel*: for each $r \in \mathbb{Z}_3$, we determine the mod-10 value assignment $r_i^{(10)}$ for cells i in $\mathcal{V}_r$ by solving

$$(S1.2.r) \quad \min \quad \sum_{i,j \in \mathcal{V}_r} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i^{(10)} = r_j^{(10)}\},$$

$$\text{s.t.} \quad r_i^{(10)} \in \mathbb{Z}_{10}, \quad \forall i \in \mathcal{V}_r.$$

As each $\mathcal{V}_r$ can be much smaller than $\mathcal{V}$, this approach of optimizing over $\mathcal{V}_r$ in parallel is more efficient than optimizing over the whole set $\mathcal{V}$ directly. The problem $(S1.2.r)$ is in fact a *Min-10-Partition* problem which is NP-hard [21–23]. We will propose an algorithm for solving it in Section IV.

After solving sub-problems 1.1 and 1.2, we obtain a mod-3 value assignment $\boldsymbol{r}^{(3)} \in \mathbb{Z}_3^N$ and a mod-10 value assignment $\boldsymbol{r}^{(10)} \in \mathbb{Z}_{10}^N$. We then invoke the CRT to get a mod-30 value assignment $\boldsymbol{r} \in \mathbb{Z}_{30}^N$.

C. Graph Coloring for Quotient Assignment

Finally, we study sub-problem 2. Recall that the goal is to determine a quotient assignment q that eliminates collisions and confusions. Similar to sub-problem 1.2, a key observation for reducing the problem size is that if two cells are assigned different mod-30 values, then regardless of the quotient assignment, they must have different PCI values and thus no collision or confusion. In light of this, we divide the network into 30 *clusters* $\mathcal{C}_0, \mathcal{C}_1, ..., \mathcal{C}_{29}$ according to the mod-30 value assignment:

$$\mathcal{C}_r = \{i \in \mathcal{V} \mid r_i = r\}, \quad r \in \mathbb{Z}_{30}.$$

We then perform quotient assignment on each $\mathcal{C}_r$ in parallel: for each $r \in \mathbb{Z}_{30}$, we determine the quotient assignment q_i for cells i in $\mathcal{C}_r$ by solving

$$(S2.r) \quad \text{find} \quad q_i, \quad i \in \mathcal{C}_r,$$
$$\text{s.t.} \quad q_i \neq q_j, \quad \forall (i,j) \in \mathcal{E} \cap (\mathcal{C}_r \times \mathcal{C}_r),$$
$$q_i \in \mathbb{Z}_+, \quad \forall i \in \mathcal{C}_r,$$
$$30q_i + r_i \in \mathbb{Z}_{1008}, \quad \forall i \in \mathcal{C}_r.$$

This problem can be reformulated as a *graph coloring problem* [11]. In the graph coloring problem, one is given an undirected graph and seeks to assign a *color* to each vertex so that vertices sharing an edge have different colors. It is easy to see that (S.2.r) can be solved by performing graph coloring over a graph with vertices $\mathcal{C}_r$ and edges $\mathcal{E} \cap (\mathcal{C}_r \times \mathcal{C}_r)$, and setting up a one-to-one correspondence between the colors and values of the quotient. To solve the graph coloring problem, we directly use GGC from [11, 24].

The graph coloring method does not always guarantee that the PCI range constraint $30q_i + r_i \in \mathbb{Z}_{1008}$ is satisfied. When too many colors are used, some cells may be assigned overly large q_i, violating the PCI range constraint. To tackle this issue, we develop a heuristic post-processing procedure that re-assigns quotients q_i to cells i whose PCIs exceed 1007, at the cost of potentially increasing the numbers of collisions and confusions. For each of these cells i, we fix its mod-30 value r_i, and update q_i to a value $q \in \mathbb{Z}$ that satisfies $30q + r_i \leq 1007$ and minimizes the number of collisions and confusions with cells that already satisfy the PCI range constraint. We process the constraint-violating cells in descending order of the total number of neighbors and second-order neighbors, so as to prioritize highly constrained regions. The full algorithm is provided in Section A of the supplementary material [25].

D. Summary

Below we summarize our approach for PCI assignment, which decomposes the problem into assigning mod-3 values $r^{(3)}$, mod-10 values $r^{(10)}$, and quotients q. An illustration is given by Figure 2.

1) **Mod-3 value assignment.** Solve the Min-3-Partition problem (S1.1) to obtain a mod-3 value assignment $r^{(3)} \in \mathbb{Z}_3^N$. This step minimizes the mod-3 interference. The algorithm for solving Min-3-Partition is introduced in Section IV.
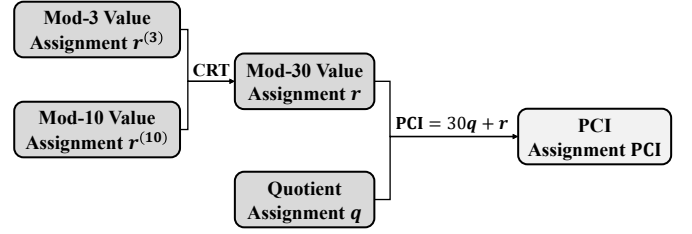

Fig. 2: Illustration of the proposed PCI assignment framework. The framework comprises two primary steps: mod-30 value assignment and quotient assignment. In the mod-30 value assignment, we first solve Min-3-Partition to obtain mod-3 values $r^{(3)}$ and then solve Min-10-Partition to determine the mod-10 values $r^{(10)}$. Subsequently, we derive the mod-30 values r based on the CRT. In the quotient assignment, solve the graph coloring problems based on greedy graph coloring to determine the quotient values q. Finally, the PCI value **PCI** for each cell is computed based on the mod values r and quotient values q.

2) **Mod-10 value assignment.** Solve the Min-10-Partition problems (S1.2.r), $r = 0, 1, 2$ in parallel to obtain a mod-10 value assignment $r^{(10)} \in \mathbb{Z}_{10}^N$. This step minimizes the mod-30 interference. The algorithm for solving Min-10-Partition is introduced in Section IV.

3) **Mod-30 value assignment.** With $r^{(3)}$ and $r^{(10)}$, apply the CRT to obtain a mod-30 value assignment $r \in \mathbb{Z}_{30}^N$.

4) **Quotient assignment.** Solve the graph coloring problems (S2.r), $r = 0, 1, ..., 29$ in parallel to obtain a quotient assignment $q \in \mathbb{Z}_+^N$. This step eliminates collisions and confusions.

5) **PCI assignment.** The PCI assignment is given by

$$\mathbf{PCI} = 30q + r.$$

While our main focus is the joint minimization of mod-3 and mod-30 interference, our decomposition framework naturally extends to arbitrary single mod-k interference by expressing each PCI as $\text{PCI}_i = kq_i + r_i$, solving a Min-$k$-Partition for r_i, and applying graph coloring for q_i. Furthermore, it supports joint optimization over multiple moduli, provided they have the coprime structure required by the CRT. This yields a unified and scalable framework for systematically decomposing the multiple modular interferences minimization.

IV. RELAXATION-FREE MIN-k-PARTITION AND PMD

In this section, we propose a relaxation-free approach to the general Min-k-Partition problem, which appears in the mod-3 value assignment and mod-10 value assignment in our PCI assignment approach. The Min-k-Partition problem is NP-hard [21–23]. Current approaches, such as semidefinite relaxation [26–30] and rank-2 relaxation methods [31] are effective only for Min-2-Partition, which is inadequate for large-scale 5G wireless networks. Furthermore, the inherent relaxation error associated with these methods often leads to suboptimal solutions. We will now formally define the Min-k-Partition problem and introduce our relaxation-free approach.

A. Relaxation-Free Min-k-Partition

The general *Min-k-Partition* problem aims to divide the vertices in a graph into k subsets and minimize the interference between vertices within each subset. Using our notation, it can be formulated as

$$\min_{\boldsymbol{r}} \quad \sum_{i,j=1}^{N} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\},$$
$$\text{s.t.} \quad \boldsymbol{r} \in \mathbb{Z}_k^N.$$

In our PCI assignment approach, we take $k = 3$ (resp. $k = 10$) for mod-3 (resp. mod-10) value assignment, where each mod-3 (resp. mod-10) value corresponds to a subset of cells. Here, we have slightly abused the notation N, as the mod-10 assignment is done over each mod-3 partition instead of the whole set of cells.

To solve Min-k-Partition, we use a one-hot encoding scheme for each of the k subsets. Specifically, we set

$$\boldsymbol{x}_i = \boldsymbol{e}_{r_i},$$

where $\boldsymbol{e}_{r_i}$ denotes the k-dimensional unit vector whose r_i-th entry is 1, and all other entries are 0. Then, we can reformulate Min-k-Partition as

$$(\text{P3}) \quad \min_{\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} [\boldsymbol{W}]_{i,j} \boldsymbol{x}_i^T \boldsymbol{x}_j \quad (3)$$
$$\text{s.t.} \quad \boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_k\}, \quad \forall i \in \mathcal{V}.$$

The constraint $\boldsymbol{x}_i \in \{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_k\}$ in problem (P3) is discrete and nonconvex. As $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_k$ are precisely the k-sparse vertices of the probability simplex $\Delta_k = \{(x_1, \cdots, x_k)^T \in \mathbb{R}^k \mid \sum_{i=1}^{k} x_i = 1, x_i \geq 0\}$, we utilize the following norm-based characterization of sparsity proposed by [32].

Lemma 1 (Norm Condition [32]). *Fix $q > p \geq 1$. Any non-zero vector $\boldsymbol{x} = (x_1, \cdots, x_k)^T \in \mathbb{R}^k$ has one non-zero entry if and only if $\|\boldsymbol{x}\|_p = \|\boldsymbol{x}\|_q$.*

Proof. See Appendix A. □

Using Lemma 1, we arrive at the following relaxation-free reformulation of Min-k-Partition:

$$(\text{P4}) \quad \min_{\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N]} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} [\boldsymbol{W}]_{i,j} \boldsymbol{x}_i^T \boldsymbol{x}_j \quad (4\text{a})$$
$$\text{s.t.} \quad \|\boldsymbol{x}_i\|_p = \|\boldsymbol{x}_i\|_q, \quad \forall i \in \mathcal{V}, \quad (4\text{b})$$
$$\boldsymbol{x}_i \in \Delta_k, \quad \forall i \in \mathcal{V}, \quad (4\text{c})$$

where $1 \leq p < q$.

B. Penalized Mirror Descent Algorithm

Given the equivalence of (P3) and (P4), we propose a PMD algorithm to solve (P4) by constructing a sequence of penalized problems and solving them by mirror descent (MD).

1) Norm-Equality Penalization: In view of its discrete feasible region, directly optimizing (P4) may exhibit sensitivity to the initial point. Therefore, we apply the penalty method [33–36] to gradually arrive at a binary solution, mitigating sensitivity to initial points. Consider the following penalized version of (P4):

$$\min_{\boldsymbol{X}=[\boldsymbol{x}_1,\cdots,\boldsymbol{x}_N]} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} [\boldsymbol{W}]_{i,j} \boldsymbol{x}_i^T \boldsymbol{x}_j + \frac{\rho}{v} \sum_{i=1}^{N} (\|\boldsymbol{x}_i\|_p^v - \|\boldsymbol{x}_i\|_q^v), \quad (5\text{a})$$
$$\text{s.t.} \quad \boldsymbol{X} \in \Delta_k^N, \quad (5\text{b})$$

where $v > 0$, and Δ_k^N is the Cartesian product of N copies of the k-dimensional probability simplex. We note that $\|\boldsymbol{x}\|_p^v - \|\boldsymbol{x}\|_q^v \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^k$ whenever $q \geq p \geq 1$. To facilitate efficient computation, we adopt the smooth setting in [32] with $p = 1$, $q = 2$, and $v = 2$, so that the resulting penalty is continuously differentiable. This enables the use of scalable first-order optimization algorithms, which are significantly more efficient than proximal methods typically required for non-smooth penalties. The penalized version of (P4) can be written as

$$(\text{P5}) \quad \min_{\boldsymbol{X}} \quad \text{Tr}\left(\boldsymbol{X}\left(\boldsymbol{W} - \frac{\rho}{2}\boldsymbol{I}\right)\boldsymbol{X}^T\right) + \frac{N\rho}{2} \triangleq F(\boldsymbol{X};\rho)$$
$$\text{s.t.} \quad \boldsymbol{X} \in \Delta_k^N, \quad (6)$$

where $\rho > 0$ is the penalty parameter. The detailed derivation of problem (P5) is provided in Appendix B.

Next, we present the following Theorem 1 to establish the theoretical foundation for the exactness of the proposed penalty method.

Theorem 1 (Exactness of the Penalized Problem). *Let $\rho_0 = \max(2\lambda_{\max}(\boldsymbol{W}), 0)$. When $\rho > \rho_0$, the problem (P5) has the same optimal solutions as (P4).*

Proof. See Appendix C. □

Remark 1. *Although penalization is often considered a form of relaxation, Theorem 1 rigorously establishes that the proposed framework remains relaxation-free under the condition that the penalty parameter ρ is sufficiently large. Specifically, when ρ exceeds a certain threshold determined by the problem constants, the penalization guarantees equivalence to the original problem. This ensures that the proposed approach maintains the exactness of the original optimization formulation while benefiting from the computational advantages of penalization.*

2) Solving the Penalized Problem based on MD: For a fixed ρ, (P5) comprises optimizing a non-convex objective function over the product of probability simplices. The MD algorithm is well-suited for this task. In particular, the m-th iteration of MD is

$$\boldsymbol{X}^{(m+1)} = \arg\min_{\boldsymbol{X} \in \Delta_k^N} \Big\{ F(\boldsymbol{X}^{(m)};\rho)$$
$$+ \left\langle \nabla F(\boldsymbol{X}^{(m)};\rho), \boldsymbol{X} - \boldsymbol{X}^{(m)} \right\rangle$$
$$+ \frac{1}{t_m} B_\omega(\boldsymbol{X}, \boldsymbol{X}^{(m)}) \Big\},$$

where t_m is the step-size of the m-th iteration, B_ω is the Bregman divergence with distance generating function ω:

$$B_\omega(\boldsymbol{X}, \boldsymbol{Y}) = \omega(\boldsymbol{X}) - \omega(\boldsymbol{Y}) - \langle \nabla\omega(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y} \rangle. \quad (7)$$

To account for the geometric structure of the probability simplex, we employ the KL divergence as the specific Bregman divergence in our approach. The KL divergence effectively quantifies the disparity between probability vectors while capturing the variations within a distribution. It is defined as

$$\mathrm{KL}(\boldsymbol{X}, \boldsymbol{X}^{(m)}) = \sum_{j=1}^{N} \sum_{i=1}^{k} [\boldsymbol{X}]_{i,j} \log \frac{[\boldsymbol{X}]_{i,j}}{[\boldsymbol{X}^{(m)}]_{i,j}}. \quad (8)$$

The following proposition indicates that each iteration of MD for solving P5 has a closed-form solution.

Proposition 1 (Closed-form Iteration of MD [37]). *The m-th iteration of MD for solving (P5) is given by*

$$\boldsymbol{X}^{(m+1)} = \arg\min_{\boldsymbol{X} \in \Delta_k^N} \left\{ \left\langle \boldsymbol{X}^{(m)}(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{X} \right\rangle \right.$$
$$\left. + \frac{1}{t_m} \mathrm{KL}(\boldsymbol{X}, \boldsymbol{X}^{(m)}) \right\}. \quad (9)$$

Equivalently, the update can be computed entrywise via

$$[\boldsymbol{X}^{(m+1)}]_{i,j}$$
$$= \frac{[\boldsymbol{X}^{(m)}]_{i,j} \exp(-t_m[\boldsymbol{X}^{(m)}(2\boldsymbol{W} - \rho\boldsymbol{I})]_{i,j})}{\sum_{p=1}^{k} [\boldsymbol{X}^{(m)}]_{p,j} \exp(-t_m[\boldsymbol{X}^{(m)}(2\boldsymbol{W} - \rho\boldsymbol{I})]_{p,j})}. \quad (10)$$

3) Convergence Analysis: MD has been extensively studied in stochastic and non-smooth optimization problems [38–40]. While these studies provide valuable insights into the convergence behavior of MD in more general and challenging scenarios, we focus on its specific application for solving the penalized problem (P5) as formulated in (10).

Before delving into the theoretical underpinnings of the proposed MD, we will first establish fundamental definitions to characterize a proper solution to problem (P5).

Definition 1 (First-order Stationary Point). *For problem (P5), $\boldsymbol{X}^* \in \Delta_k^N$ is a stationary point if and only if $\langle \boldsymbol{X}^*(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{X} - \boldsymbol{X}^* \rangle \geq 0$ for all $\boldsymbol{X} \in \Delta_k^N$.*

Then, we can derive the overall convergence analysis of the MD algorithm (9) for solving the penalized problem P5 in the following Theorem 2.

Theorem 2 (Convergence of the MD). *For step size $t_m = \frac{1}{L+m+1}$, the sequence $\{\boldsymbol{X}^{(m)}\}_{m=0}^{T-1}$ generated by the MD algorithm (9) converges to a stationary point of P5 at a rate $T \geq \sqrt{\frac{2\|\boldsymbol{W}\|_{1,1} + \rho N}{\epsilon_1}}$ with respect to the stopping criterion $\mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}) \leq \epsilon_1$, where T is the number of iterations.*

Proof. See Appendix D. $\square$

Remark 2. *Theorem 2 implies that the convergence rate, as measured by the stopping criterion $\mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}) \leq \epsilon_1$, achieves an efficient convergence rate of $\mathcal{O}\left(\sqrt{\frac{1}{\epsilon_1}}\right)$. We choose the diminishing step size $t_m = \frac{1}{L+m+1}$, where the smoothness*
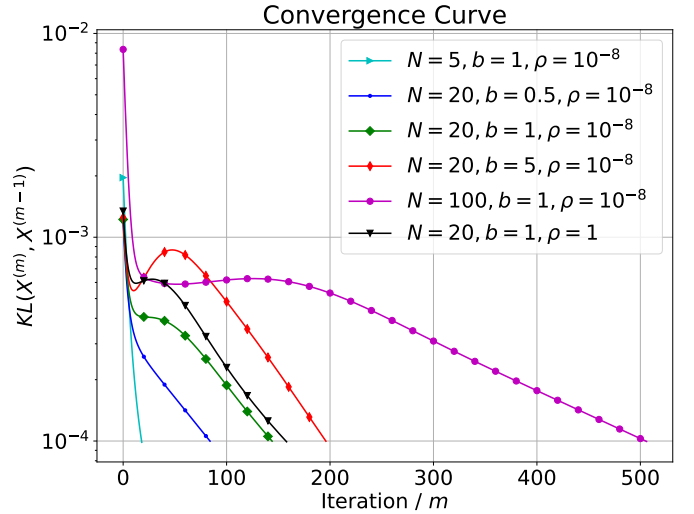
Fig. 3: Convergence curve for mod-3 optimization problem of MD with KL divergence.

constant L accounts for the curvature of the objective in early exploration, and the $O(1/m)$ decay mitigates oscillation in later stages. Moreover, as the $\ell_{1,1}$-norm of matrix $\boldsymbol{W}$, the penalized parameter ρ, and the number of cells N increase while keeping ϵ_1 fixed, the number of iterations required for convergence will also increase. Upon obtaining MD for solving the penalized problem P5 in (6), we can leverage the proposed PMD method to gradually increase ρ in P5 and solve the corresponding sequence of penalized problems via MD to achieve a robust solution of P4.

C. Empirical Analysis of MD

In this section, we empirically investigate the convergence rate of the MD algorithm, as detailed in Theorem 2. We synthetically construct an instance of the matrix $\boldsymbol{W}$ by first generating a random non-negative matrix $\boldsymbol{W}_0$ whose entries are i.i.d. uniformly distributed over $[0, b]$ for some $b > 0$, and then setting symmetrization $\boldsymbol{W} = (\boldsymbol{W}_0 + \boldsymbol{W}_0^T)/2$.

We consider the Min-3-Partition problem ($k = 3$), with $b \in \{0.5, 1, 5\}$, and $N \in \{5, 20, 100\}$. We set $\epsilon_1 = 10^{-4}$ and $\rho \in \{10^{-8}, 1\}$. Figure 3 plots the difference between consecutive iterates $\mathrm{KL}(\boldsymbol{X}^{(m)}, \boldsymbol{X}^{(m-1)})$ against the iteration count m. Each curve exhibits a general downward trend until the stopping criterion is reached.

Specifically, in the original scenario with $N = 20$, $b = 1$, and $\rho = 10^{-8}$, convergence is achieved in 145 iterations. Decreasing the parameter b decreases the iteration count. Similarly, decreasing N leads to convergence in fewer iterations. Changes in N have a more pronounced effect on the iteration count than changes in b, reflecting the linear convergence rate with respect to N and the square root rate with respect to b. Increasing the penalization parameter ρ slightly increases the iteration count. These trends are consistent with the analysis in Theorem 2.

D. Pre-processing and Post-processing of the Proposed PMD Algorithm

To enhance the efficiency of the PMD algorithm, we further propose the following pre-processing and post-processing procedures.

In pre-processing, to ensure that the gradient of the distance-generating function is well-defined, we avoid initializing at boundary points of the simplex Δ_k^N. This is achieved via a random initialization strategy for the starting point $X^0 = [x_1^0, \cdots, x_N^0]$, where each column x_j^0 is generated as follows:

Random Initialization:

$$\text{Step 1: Generate} \quad z \overset{\text{i.i.d.s}}{\sim} U^{k \times 1}[0, 1],$$

$$\text{Step 2: Normalize} \quad x_j^0 = \frac{z}{\|z\|_1}. \tag{11}$$

In our experiments, we observe that PMD almost always returns a feasible solution to (P4), and thus a feasible solution to the Min-k-Partition problem. In the post-processing procedure, we propose a local search method in Algorithm 1 to improve the PMD solution by searching over cell pairs. It introduces a mild additional $\mathcal{O}(N^2)$ complexity.

Algorithm 1: Local Search

Data: The solution obtained by PMD X
Result: The refined solution $\widetilde{X}$
for $i = 1$ *to* N **do**
  for $j = i + 1$ *to* N **do**
    Calculate $(k_1^*, k_2^*) = \arg\min_{(k_1, k_2)}$

$$F([x_1, \cdots, \underbrace{e_{k_1}}_{i\text{-th column}}, \cdots, \underbrace{e_{k_2}}_{j\text{-th column}}, \cdots, x_N]; 0)$$

    Set $(x_i, x_j) = (e_{k_1^*}, e_{k_2^*})$

return $\widetilde{X} = X$

E. Overall Implementation of the PMD algorithm

The overall implementation of the PMD for relaxation-free Min-k-Partition is shown in Algorithm 2. The PMD iteration consists of an outer loop and an inner loop, where the outer loop gradually increases the penalization parameter ρ in (P5), and the inner loop solves (P5) via the update (10). In the update (10), we choose $t_m = 1/(L + m + 1)$ according to Theorem 2. The termination criterion for the outer loop is

$$\frac{\|QX^\tau(QX^\tau)^T - I_k\|_F}{k^2} \le \epsilon_2,$$

where Q is a diagonal matrix that scales the rows of X^τ so that they have unit 2-norms. It is based on the fact that $X \in \Delta_k^N$ is feasible for (P4) only if it has orthogonal rows. This criterion is also used in [32].

V. EXTENSION: OPTIMIZING PCIS OF A SUBSET OF CELLS

In our discussion so far, we have taken the approach of assigning the PCI for a given network from scratch. In practice,

---

Algorithm 2: Proposed PMD Method for Relaxation-free Min-k-Partition

Data: The interference matrix W, the initial penalty parameter ρ_0, the penalty increasing parameter γ, tolerance parameters $\epsilon_1, \epsilon_2 \ge 0$
Result: Solution $\widetilde{r}$

```
// Random Initialization
```
for $j = 1$ *to* N *independently* **do**
  Generate x_j^0 by Random Initialization (11)
Set $X^0 = [x_1^0, \ldots, x_N^0]$;

```
// PMD Iteration
```
Set $\tau = 0, \rho = \rho_0$
repeat
  Set $m = 0$, $X^{(m)} = X^\tau$
  repeat
    Update $X^{(m+1)}$ by (10)
    Set $m = m + 1$
  until $\text{KL}(X^{(m)}, X^{(m-1)}) \le \epsilon_1$;
  Set $X^{\tau+1} = X^{(m)}$
  Set $\tau = \tau + 1$
  Update $\rho = \gamma\rho$
until $\|QX^\tau(QX^\tau)^T - I_k\|_F/k^2 \le \epsilon_2$, *where Q is a diagonal matrix such that the rows of QX^τ have unit 2-norms*;

```
// Local Search
```
Set $\widetilde{X} = X^\tau$
repeat
  $\widetilde{X} = $ **Local Search**$(\widetilde{X})$
until *No further improvement*;
Convert $\widetilde{X}$ into $\widetilde{r}$ via: $\widetilde{r}_i = j$ if $\widetilde{x}_i = e_j$, $j = 1, ..., k$.
return $\widetilde{r}$

---

however, practitioners usually have a (possibly handcrafted and imperfect) PCI assignment beforehand and seek to improve it by making changes to the PCIs of only *a subset* of the cells. Such a partial update can be more desirable than directly updating the PCI of the entire network because the latter may drastically change the stability of the whole 5G network. In this section, we show that our graph partitioning approach can be naturally extended to this setting.

Mathematically, we are given an existing PCI assignment $\mathbf{PCI}^{(0)} = (\text{PCI}_1^{(0)}, ..., \text{PCI}_N^{(0)})$ and a subset of *changeable cells* $\mathcal{S} \subseteq \mathcal{V}$. Let $\mathcal{U} = \mathcal{V} \backslash \mathcal{S}$, which denotes the set of *unchangeable cells*. Our goal is to optimize mod-3 and mod-30 interference and eliminate collisions and confusions, by only updating the PCIs of the changeable cells.

To adapt our approach to this setting, we apply the decomposition framework in Section III to only the changeable cells, where the PCI assignment $(\text{PCI}_i)_{i \in \mathcal{S}}$ for the changeable cells is decomposed into mod-3 value assignment $(\widetilde{r}_i^{(3)})_{i \in \mathcal{S}}$, mod-10 value assignment $(\widetilde{r}_i^{(10)})_{i \in \mathcal{S}}$, and quotient assignment $(q_i)_{i \in \mathcal{S}}$. At the same time, we need to take into account the interaction between changeable and unchangeable cells. We now explain

how this can be achieved.

A. Mod-3 Value and Mod-10 Value Assignment

To assign mod-3 and mod-10 values (sub-problems 1.1 and 1.2 in Section III-B), we consider a variant of the Min-k-Partition problem where the optimization variables are restricted to the changeable cells:

$$\min_{(r_i)_{i \in \mathcal{S}}} \quad \sum_{i,j \in \mathcal{S}} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\}$$
$$+ 2 \sum_{i \in \mathcal{S}, j \in \mathcal{U}} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\}$$
$$+ \sum_{i,j \in \mathcal{U}} [\boldsymbol{W}]_{i,j} \cdot \mathbb{1}\{r_i = r_j\},$$
$$\text{s.t.} \quad r_i \in \mathbb{Z}_k, \quad \forall i \in \mathcal{S}.$$

Here $(r_j)_{j \in \mathcal{U}}$ are pre-determined from the PCIs of the un-changeable cells. Moreover, the last term in the objective function corresponds to the interference between unchangeable cells and is thus a constant that can be ignored.

To modify the PMD algorithm for Min-k-Partition in Section IV, we introduce block matrix notations $\boldsymbol{W}_{\mathcal{S},\mathcal{S}} = [\boldsymbol{W}_{i,j}]_{i \in \mathcal{S}, j \in \mathcal{S}}$ and $\boldsymbol{W}_{\mathcal{S},\mathcal{U}} = [\boldsymbol{W}_{i,j}]_{i \in \mathcal{S}, j \in \mathcal{U}}$. Then the problem (P5) for the PMD algorithm becomes

$$\min_{\boldsymbol{X}_{\mathcal{S}}} \quad \text{Tr}\left(\boldsymbol{X}_{\mathcal{S}}\left(\boldsymbol{W}_{\mathcal{S},\mathcal{S}} - \frac{\rho}{2}\boldsymbol{I}\right)\boldsymbol{X}_{\mathcal{S}}^T\right) + 2\text{Tr}\left(\boldsymbol{X}_{\mathcal{S}}\boldsymbol{W}_{\mathcal{S},\mathcal{U}}\boldsymbol{X}_{\mathcal{U}}^T\right)$$
$$\text{s.t.} \quad \boldsymbol{X}_{\mathcal{S}} \in \Delta_k^{|\mathcal{S}|},$$

where $\boldsymbol{X}_{\mathcal{U}} \in \{\boldsymbol{e}_1, ..., \boldsymbol{e}_k\}^{|\mathcal{U}|}$ can be deduced from the existing PCI assignment. The corresponding PMD algorithm can then be easily derived.

B. Quotient Assignment

Finally, to assign quotient values (Section III-C), we solve a variant of the graph coloring problem where an existing coloring (quotient assignment) is given, and the goal is to update the colors (quotients) of changeable cells to minimize collisions and confusions. Since we cannot eliminate the collisions and confusions between two unchangeable cells, we will focus only on cell pairs that have at least one changeable cells. Thus, the problem can be formulated as

$$\text{find} \quad q_i, \quad i \in \mathcal{S},$$
$$\text{s.t.} \quad q_i \neq q_j, \quad \forall (i,j) \in \mathcal{E} \cap \left[(\mathcal{S} \times \mathcal{S}) \cup (\mathcal{S} \times \mathcal{U})\right], \quad (12)$$
$$q_i \in \mathbb{Z}_+, \quad \forall i \in \mathcal{S}.$$

Here again, $(q_j)_{j \in \mathcal{U}}$ can be computed from the existing PCI assignment, and we ignore the PCI range constraint as in Section III-C. We have also slightly abused notation, as quotient assignment is done over each individual mod-30 cluster instead of the whole set of changeable cells.

We now show that the problem (12) can, in fact, be reduced to the graph coloring problem. Let $c_1, ..., c_A$ be the colors (quotients) used by the unchangeable cells. We will replace the unchangeable cells with a clique consisting of A vertices $v_1, ..., v_A$, where v_a represents the used color c_a. Then we add an edge between a changeable cell i and v_a whenever some

unchangeable cell j has used the color c_a and $(i,j) \in \mathcal{E}$. This leads to a graph $\mathcal{H}$ with vertices $\mathcal{S} \cup \{v_1, ..., v_A\}$ and edges $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$, where

$$\mathcal{F}_1 = \mathcal{E} \cap (\mathcal{S} \times \mathcal{S}),$$
$$\mathcal{F}_2 = \{(i, v_a) : i \in \mathcal{S}, \text{ and } \exists j \in \mathcal{U} \text{ s.t. } (i,j) \in \mathcal{E}, q_j = c_a\},$$
$$\mathcal{F}_3 = \{(v_a, v_{a'}) : a \neq a'\}.$$

We will now see that performing graph coloring over this new graph $\mathcal{H}$ yields a solution to (12). Thanks to the clique structure $\mathcal{F}_3$, the vertices $v_1, ..., v_A$ will receive distinct colors, which can be made $c_1, ..., c_A$ after relabeling. Moreover, $\mathcal{F}_1$ and $\mathcal{F}_2$ force the color assignment to eliminate collisions and confusions between pairs $(i,j) \in \mathcal{S} \times \mathcal{S}$, and between pairs $(i,j) \in (\mathcal{S}, \mathcal{U})$, respectively. Therefore, this gives a solution to (12).

VI. NUMERICAL EXPERIMENTS

This section presents experiments to demonstrate the performance and efficiency of our PCI assignment approach compared with state-of-the-art baselines. All experiments are implemented in Python on a machine equipped with an Intel i9-12900K core.

A. List of Baseline Methods

We call our proposed method of graph partitioning with PMD by *GP-PMD*. The baseline methods for comparison are described as follows. We select four representative methods for solving the PCI problem.

Greedy Graph Coloring Method (GGC) [11]: This method employs a greedy coloring algorithm to address collisions and confusions. In the simulation, we utilize the codes provided from [24].

Greedy Search for BQP (BQP) [8]: We employed BQP methods to handle collisions, confusions, mod-3 interference, and mod-30 interference. We add the mod-30 interference into the objective function and set the hyperparameters to those used in the experiment *1-1-1 Greedy* in [8].

Genetic Method (Genetic) [13]: We apply a genetic method for comparison. The hyperparameter settings include a population size of 30 and 200 iterations, and the probabilities of crossover, mutation, and selection are set to 0.77, 0.3, and 0.8, respectively.

Memetic Algorithm (MA-NWS) [15]: We adopt the non-warm-start version as recommended by [15], which relies solely on the core evolutionary mechanism of the Memetic Algorithm. All hyperparameters are set according to the best-performing configuration reported in [41].

Graph Partitioning with Penalized Gradient Projection (GP-PGP): To evaluate the performance of the PMD algorithm, we consider an alternative benchmark that uses the same graph partitioning framework as GP-PMD but applies penalized gradient projection (PGP) to solve sub-problems 1.1 and 1.2. It also uses the local search procedure in Algorithm 1 to improve the PGP solution. The stopping criterion for GP-PGP is set by the Euclidean Bregman divergence: $\frac{1}{2}\|\boldsymbol{X}^{(m+1)} - \boldsymbol{X}^{(m)}\|_F^2 \leq \epsilon_1$. All hyperparameters are the same as those used in GP-PMD.

TABLE I: Structural statistics of the random geometric graphs used in our experiments (averaged over 30 instances per setting).

(N, r)	Density	Max Clique	Avg. Degree	Clustering Coef.
(100, 0.1)	0.0279	5.00	5.075	0.719
(100, 0.2)	0.1044	9.67	26.333	0.717
(100, 0.3)	0.2141	15.17	54.299	0.787
(200, 0.1)	0.0285	7.43	13.050	0.706
(200, 0.2)	0.1049	15.40	58.734	0.715
(200, 0.3)	0.2156	25.90	114.909	0.796
(500, 0.1)	0.0289	13.43	40.993	0.655
(500, 0.2)	0.1055	30.47	159.088	0.721
(500, 0.3)	0.2159	55.40	298.456	0.802

Graph Partitioning with Local Search (GP-LS): To evaluate the performance of the PMD algorithm, we include an additional baseline. It uses the same graph partitioning framework as GP-PMD, but directly generates a random solution when solving sub-problems 1.1 and 1.2, and then applies the local search procedure in Algorithm 1.

B. Synthetic Data Experiments

In this section, we test our method on synthetically generated networks.

1) Experiment Setup: We generate synthetic networks using the random geometric graph model $\text{RGG}(N, r)$, where N denotes the number of cells and r is called the connection radius. Cells are uniformly distributed in a 2D unit square, and cell j is treated as a neighbor of node i if their Euclidean distance is at most r. The interference matrix $\boldsymbol{W}$ is defined by setting the interference inversely proportional to the Euclidean distance, simulating spatial interference in wireless networks. This model is widely used to emulate realistic wireless topologies [42, 43].

To evaluate algorithm performance under diverse topological conditions, we systematically vary both the network size and spatial density. Specifically, we consider network sizes $N \in \{100, 200, 500\}$, and connection radii $r \in \{0.1, 0.2, 0.3\}$ which correspond to sparse, medium, and dense graphs, respectively. For each configuration $\text{RGG}(N, r)$, we generate 30 independent instances, which yields a total of 270 graph instances. For each type of graph, we compute standard structural metrics, including edge density, maximum clique size, average degree, and clustering coefficient. These statistics are summarized in Table I and allow us to interpret algorithm performance under varying structural complexities. The hyperparameters of GP-PMD is set as $\rho_0 = 10^{-8}$, $\gamma = 1.1$, $\epsilon_1 = 10^{-5}$, and $\epsilon_2 = 10^{-10}$, which is the same configuration as [32]. We also conduct an ablation study to examine the effects of the hyperparameters on the performance of the algorithm, and the results are provided in Section B of the supplementary material [25].

2) Experiment Results: We compare all baseline algorithms on the random geometric graphs using five evaluation metrics: mod-3 interference, mod-30 interference, number of collisions, number of confusions, and computational time. The overall performance is visualized in a heatmap (Fig. 4). The detailed box-plots are included in Section C of the supplementary material [25]. The key observations are summarized as follows.

GGC is the fastest among all methods (under $80\,\text{s}$ across all instances), but fails to reduce mod-3 or mod-30 interference. BQP eliminates all collisions and confusions across all test instances, and its interference control is generally competitive, though slightly behind MA-NWS on small and medium-sized graphs. However, its computational cost escalates as the graph size increases. Genetic achieves moderate performance across all metrics. It reduces mod-3 and mod-30 interference but fails to eliminate collisions and confusions. Compared to our proposed methods, it exhibits weaker control over mod-3 and mod-30 interference. While its runtime is substantially shorter than that of BQP and MA-NWS, it remains noticeably longer than ours. MA-NWS successfully eliminates collisions and confusions and achieves comparable performance on mod-3 interference compared with our methods on smaller instances. However, it performs worse than our methods on mod-30 interference and has a relatively high runtime. GP-PGP and GP-PMD consistently achieve strong performance: both eliminate all collisions and confusions and achieve the lowest interference on mod-3 and mod-30. While both approaches yield comparable solution quality, GP-PMD is more efficient and requires less computational time. Overall, the GP-PMD achieves a compelling trade-off between feasibility, interference suppression, and scalability, making it the most robust and versatile algorithm among the compared baselines. Moreover, to assess the statistical significance of our results, we conduct statistical hypothesis tests in Section D of the supplementary materials [25].

C. Real-Data Experiments

1) Experiment Setup: We test our method on real-world data collected in Beijing. The dataset features antennas deployed in a SISO system, focusing on scenarios involving mod-3 interference, mod-30 interference, collisions, and confusions in simulations. The raw MR data was extracted from a live downlink network comprising $10,000$ cells, of which 673 are changeable, covering a dense urban area of 53.0859 square kilometers. This data, gathered through regular measurements by UEs reporting network status and performance, was utilized to generate the sets $\mathcal{E}_1$ and $\mathcal{E}_2$, and the matrix $\boldsymbol{W}$.

For collision detection, we include the pair $(i, j) \in \mathcal{E}_1$ if both cells share the same frequency and MR data indicates that cell j is a neighbor of cell i. In the case of confusion, we include the pair $(i, j) \in \mathcal{E}_2$ if they share the same frequency and there exists another cell ℓ for which MR data shows that i and j are both its neighbors. For the interference matrix $\boldsymbol{W}$, if cell j is a neighboring cell of cell i and shares the same frequency as cell i, the entry $[\boldsymbol{W}]_{i,j}$ reflects the number of MRs between these two cells. Furthermore, to assess performance across different network sizes, we select subsets of the main dataset with varying numbers of cells N and changeable cells $S = |\mathcal{S}|$. These subsets consist of cells with consecutive cell IDs, and have N values of 401, 1398, 4396, and $10,000$, with corresponding S values of 136, 231, 368, and 673. These choices are designed to assess the algorithm's performance across a range of problem scales.
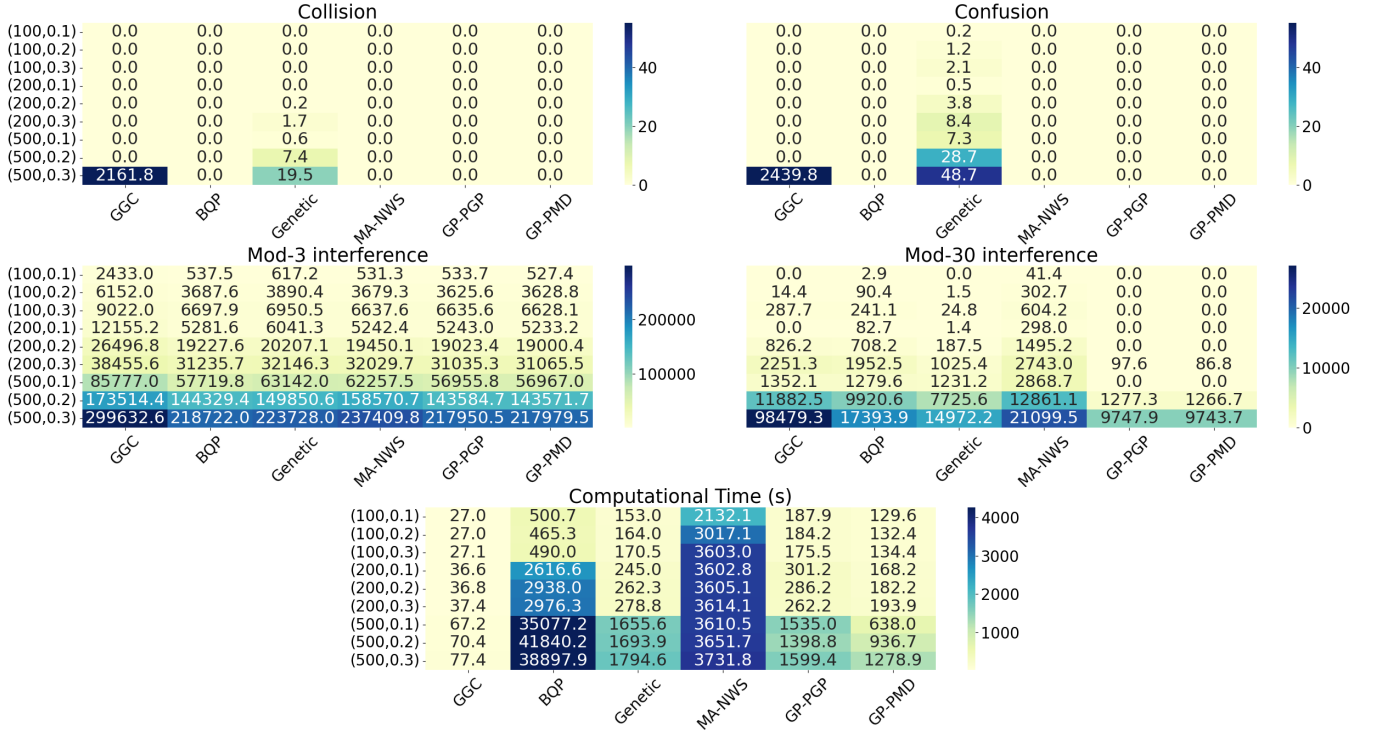
Collision

	GGC	BQP	Genetic	MA-NWS	GP-PGP	GP-PMD
(100,0.1)	0.0	0.0	0.0	0.0	0.0	0.0
(100,0.2)	0.0	0.0	0.0	0.0	0.0	0.0
(100,0.3)	0.0	0.0	0.0	0.0	0.0	0.0
(200,0.1)	0.0	0.0	0.0	0.0	0.0	0.0
(200,0.2)	0.0	0.0	0.2	0.0	0.0	0.0
(200,0.3)	0.0	0.0	1.7	0.0	0.0	0.0
(500,0.1)	0.0	0.0	0.6	0.0	0.0	0.0
(500,0.2)	0.0	0.0	7.4	0.0	0.0	0.0
(500,0.3)	2161.8	0.0	19.5	0.0	0.0	0.0

Confusion

	GGC	BQP	Genetic	MA-NWS	GP-PGP	GP-PMD
(100,0.1)	0.0	0.0	0.2	0.0	0.0	0.0
(100,0.2)	0.0	0.0	1.2	0.0	0.0	0.0
(100,0.3)	0.0	0.0	2.1	0.0	0.0	0.0
(200,0.1)	0.0	0.0	0.5	0.0	0.0	0.0
(200,0.2)	0.0	0.0	3.8	0.0	0.0	0.0
(200,0.3)	0.0	0.0	8.4	0.0	0.0	0.0
(500,0.1)	0.0	0.0	7.3	0.0	0.0	0.0
(500,0.2)	0.0	0.0	28.7	0.0	0.0	0.0
(500,0.3)	2439.8	0.0	48.7	0.0	0.0	0.0

Mod-3 interference

	GGC	BQP	Genetic	MA-NWS	GP-PGP	GP-PMD
(100,0.1)	2433.0	537.5	617.2	531.3	533.7	527.4
(100,0.2)	6152.0	3687.6	3890.4	3679.3	3625.6	3628.8
(100,0.3)	9022.0	6697.9	6950.5	6637.6	6635.6	6628.1
(200,0.1)	12155.2	5281.6	6041.3	5242.4	5243.0	5233.2
(200,0.2)	26496.8	19227.6	20207.1	19450.1	19023.4	19000.4
(200,0.3)	38455.6	31235.7	32146.3	32029.7	31035.3	31065.5
(500,0.1)	85777.0	57719.8	63142.0	62257.5	56955.8	56967.0
(500,0.2)	173514.4	144329.4	149850.6	158570.7	143584.7	143571.7
(500,0.3)	299632.6	218722.0	223728.0	237409.8	217950.5	217979.5

Mod-30 interference

	GGC	BQP	Genetic	MA-NWS	GP-PGP	GP-PMD
(100,0.1)	0.0	2.9	0.0	41.4	0.0	0.0
(100,0.2)	14.4	90.4	1.5	302.7	0.0	0.0
(100,0.3)	287.7	241.1	24.8	604.2	0.0	0.0
(200,0.1)	0.0	82.7	1.4	298.0	0.0	0.0
(200,0.2)	826.2	708.2	187.5	1495.2	0.0	0.0
(200,0.3)	2251.3	1952.5	1025.4	2743.0	97.6	86.8
(500,0.1)	1352.1	1279.6	1231.2	2868.7	0.0	0.0
(500,0.2)	11882.5	9920.6	7725.6	12861.1	1277.3	1266.7
(500,0.3)	98479.3	17393.9	14972.2	21099.5	9747.9	9743.7

Computational Time (s)

	GGC	BQP	Genetic	MA-NWS	GP-PGP	GP-PMD
(100,0.1)	27.0	500.7	153.0	2132.1	187.9	129.6
(100,0.2)	27.0	465.3	164.0	3017.1	184.2	132.4
(100,0.3)	27.1	490.0	170.5	3603.0	175.5	134.4
(200,0.1)	36.6	2616.6	245.0	3602.8	301.2	168.2
(200,0.2)	36.8	2938.0	262.3	3605.1	286.2	182.2
(200,0.3)	37.4	2976.3	278.8	3614.1	262.2	193.9
(500,0.1)	67.2	35077.2	1655.6	3610.5	1535.0	638.0
(500,0.2)	70.4	41840.2	1693.9	3651.7	1398.8	936.7
(500,0.3)	77.4	38897.9	1794.6	3731.8	1599.4	1278.9

Fig. 4: Experiment results on random geometric graphs.

2) Experiment Results: We summarize the experiment results in Table II, III, IV, and V. [1] The results demonstrate that GP-PMD significantly outperforms all baseline methods in minimizing mod-3 and mod-30 interference, and resolving collisions and confusions, all while taking the least computational time. Specifically, compared to the state-of-the-art methods, GP-PMD reduces mod-3 interference by 8.3% and mod-30 interference by 99.7%, and achieves a 20 times speedup in runtime on average. In contrast, GGC fails to address the minimization of mod-3 and mod-30 interference. Genetic fails to achieve collision-free and confusion-free solutions. Genetic and BQP all produce significantly higher mod-3 and mod-30 interference and require substantially longer computational time. Moreover, BQP is unable to generate any solution within two weeks for large-scale instances. Lastly, the comparison with GP-PGP highlights the superior efficiency and effectiveness of the proposed PMD algorithm in minimizing mod-3 and mod-30 interference.

VII. CONCLUSION

In this paper, we introduce a novel approach for PCI assignment, decomposing it into Min-3-Partition, Min-10-Partition, and graph coloring. To solve the challenging NP-hard Min-k-Partition problem, we present a relaxation-free reformulation that can be solved by an efficient PMD algorithm. Theoretical results are developed to provide strong guarantees for

[1]In these tables, 'Coll.' indicates the number of collisions, 'Conf.' refers to the number of confusions, 'Mod-3' denotes the mod-3 interference value, 'Mod-30' is the mod-30 interference value, and 'Time' represents the runtime of the methods.

TABLE II: The evaluation results on the real-world data with $N = 401, S = 136$.

Models	Coll.	Conf.	Mod-3.	Mod-30.	Time (s)
GGC [11, 24]	0	0	2887700	614093	48
BQP [8]	0	0	668280	15587	2297
Genetic [13]	2	2	758546	17745	376
MA-NWS [15]	0	0	640415	55505	422
GP-LS	0	0	690063	1	179
GP-PGP	0	0	641961	**0**	279
GP-PMD	0	0	**628418**	**0**	197

TABLE III: The evaluation results on the real-world data with $N = 1398, S = 231$.

Models	Coll.	Conf.	Mod-3.	Mod-30.	Time (s)
GGC [11, 24]	0	0	4283366	429858	85
BQP [8]	0	0	1216883	39039	18936
Genetic [13]	2	5	1592459	32406	1407
MA-NWS [15]	0	16	1236074	104253	731
GP-LS	0	0	1207612	326	332
GP-PGP	0	0	1185974	**159**	372
GP-PMD	0	0	**1143088**	170	352

the proposed method. Experimental evaluations on synthetic and real-world datasets show the superior performance and computational efficiency of the proposed method over baseline methods. While our method shows strong performance on networks with up to $10,000$ cells, scaling to even larger networks may benefit from more advanced techniques. Future work could explore neural optimization or learning-based surrogates to further improve efficiency and generalization in interference-aware settings.

TABLE IV: The evaluation results on the real-world data with $N = 4396, S = 368$.

Models	Coll.	Conf.	Mod-3.	Mod-30.	Time (s)
GGC [11, 24]	0	0	9349226	753756	107
BQP [8]*	-	-	-	-	-
Genetic [13]	14	16	5159102	140707	7403
MA-NWS [15]	0	25	2909566	255494	2000
GP-LS	0	0	3429811	495	1099
GP-PGP	0	0	2833178	287	1179
GP-PMD	0	0	**2831831**	**216**	971

∗: BQP fails to return a solution in two weeks of time.

TABLE V: The evaluation results on the real-world data with $N = 10000, S = 673$.

Models	Coll.	Conf.	Mod-3.	Mod-30.	Time (s)
GGC [11, 24]	0	0	18039785	1052895	254
BQP [8]*	-	-	-	-	-
Genetic [13]	33	32	11642161	574454	55515
MA-NWS [15]	0	91	6838767	580523	12398
GP-LS	0	0	7466815	19367	6967
GP-PGP	0	0	6607312	**6451**	6672
GP-PMD	0	0	**6584238**	6617	6185

∗: BQP fails to return a solution in two weeks of time.

APPENDIX A
PROOF OF LEMMA 1

Proof. Clearly, if $\boldsymbol{x}$ has one non-zero entry, then $\|\boldsymbol{x}\|_p = \|\boldsymbol{x}\|_q$. Below we prove that if $\|\boldsymbol{x}\|_q = \|\boldsymbol{x}\|_p$, then $\boldsymbol{x}$ has exactly one non-zero entry.

First, we have

$$
\|\boldsymbol{x}\|_q^q = \sum_{i=1}^{k} |x_i|^q = \sum_{i=1}^{k} |x_i|^p |x_i|^{q-p}
$$
$$
\overset{(a)}{\leq} \left(\sum_{i=1}^{k} |x_i|^p \right) \left(\sum_{i=1}^{k} |x_i|^{q-p} \right), \qquad (13)
$$

and

$$
\|\boldsymbol{x}\|_q^q \overset{(b)}{=} \|\boldsymbol{x}\|_p^q = \left(\sum_{i=1}^{k} |x_i|^p \right)^{\frac{q}{p}}
$$
$$
= \left(\sum_{i=1}^{k} |x_i|^p \right) \left(\sum_{i=1}^{k} |x_i|^p \right)^{\frac{q-p}{p}}
$$
$$
\overset{(c)}{\geq} \left(\sum_{i=1}^{k} |x_i|^p \right) \left(\sum_{i=1}^{k} |x_i|^{p \cdot \frac{q-p}{p}} \right)
$$
$$
= \left(\sum_{i=1}^{k} |x_i|^p \right) \left(\sum_{i=1}^{k} |x_i|^{q-p} \right). \qquad (14)
$$

Here, (a) is due to the inequality $\sum_{i=1}^{k} a_i b_i \leq (\sum_{i=1}^{k} a_i)(\sum_{i=1}^{k} b_i)$ for $a_i, b_i \geq 0$, (b) follows from the assumption in the necessary condition, and (c) uses the inequality $\sum_{i=1}^{k} a_i^t \leq (\sum_{i=1}^{k} a_i)^t$ for $a_i \geq 0$ and $t > 0$.

Combining (13) and (14), we have

$$
\|\boldsymbol{x}\|_q^q = \left(\sum_{i=1}^{k} |x_i|^p \right) \left(\sum_{i=1}^{k} |x_i|^{q-p} \right).
$$

On the other hand,

$$
\|\boldsymbol{x}\|_q^q = \sum_{i=1}^{k} |x_i|^p |x_i|^{q-p}.
$$

We thus obtain

$$
\sum_{i \neq j} |x_i|^p |x_j|^{q-p} = 0,
$$

which implies

$$
|x_i|^p |x_j|^{q-p} = 0, \quad \forall i \neq j. \qquad (15)
$$

Since $\boldsymbol{x} \neq \boldsymbol{0}$, there exists an index i^* such that $|x_{i^*}|^p > 0$. From (15), it follows that

$$
|x_j|^{q-p} = 0, \quad \forall j \neq i^*, \qquad (16)
$$

which implies $x_j = 0$ for all $j \neq i^*$. Hence, $\boldsymbol{x}$ has exactly one non-zero entry. $\square$

APPENDIX B
DERIVATION OF (P5)

We substitute $p = 1, q = 2, v = 2$ into (5a) to obtain

$$
\sum_{i=1}^{N} \sum_{j=1}^{N} [\boldsymbol{W}]_{i,j} \boldsymbol{x}_i^T \boldsymbol{x}_j + \frac{\rho}{2} \sum_{i=1}^{N} (\|\boldsymbol{x}_i\|_1^2 - \|\boldsymbol{x}_i\|_2^2)
$$
$$
= \text{Tr}(\boldsymbol{X} \boldsymbol{W} \boldsymbol{X}^T) + \frac{\rho}{2} \sum_{i=1}^{N} (1 - \|\boldsymbol{x}_i\|_2^2)
$$
$$
= \text{Tr}(\boldsymbol{X} \boldsymbol{W} \boldsymbol{X}^T) - \frac{\rho}{2} \text{Tr}(\boldsymbol{X} \boldsymbol{X}^T) + \frac{\rho}{2} N
$$
$$
= \text{Tr} \left(\boldsymbol{X} \left(\boldsymbol{W} - \frac{\rho}{2} \boldsymbol{I} \right) \boldsymbol{X}^T \right) + \frac{\rho}{2} N. \qquad (17)
$$

APPENDIX C
PROOF OF THEOREM 1

We first prove the strictly concavity of $F(\boldsymbol{X}; \rho)$ when $\rho > \rho_0 = 2 \max(\lambda_{\max}(\boldsymbol{W}), 0)$.

Lemma 2 (Strict Concavity of F). *Let $\rho_0 = \max(2\lambda_{\max}(\boldsymbol{W}), 0)$. When $\rho > \rho_0$, $F(\boldsymbol{X}; \rho)$ is a strictly concave function of $\boldsymbol{X}$.*

Proof. Let the eigenvalue decomposition of the interference $\boldsymbol{W}$ be given by $\boldsymbol{W} = \boldsymbol{P} \boldsymbol{\Sigma} \boldsymbol{P}^T$, where $\boldsymbol{P}$ is an orthogonal matrix, $\boldsymbol{\Sigma}$ is a diagonal matrix whose diagonal entries are the eigenvalues of $\boldsymbol{W}$. Then

$$
\boldsymbol{W} - \frac{\rho}{2} \boldsymbol{I} = \boldsymbol{P} \boldsymbol{\Sigma} \boldsymbol{P}^T - \frac{\rho}{2} \boldsymbol{P} \boldsymbol{P}^T = \boldsymbol{P} \left(\boldsymbol{\Sigma} - \frac{\rho}{2} \boldsymbol{I} \right) \boldsymbol{P}^T,
$$

Since $\rho > \rho_0 = 2\lambda_{\max}(\boldsymbol{W})$, we have the diagonal elements of $\boldsymbol{\Sigma} - \frac{\rho}{2} \boldsymbol{I}$ are all strictly less than 0, making $\boldsymbol{W} - \frac{\rho}{2} \boldsymbol{I}$ a negative definite matrix. Hence, $F(\boldsymbol{X}; \rho)$ is strictly concave. $\square$

A useful fact we will use is that every $\boldsymbol{X} \in \Delta_k^N$ can be written as a convex combination of matrices where each

column vector only has one non-zero element. Specifically, we express $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \Delta_k^N$ as

$$
\begin{aligned}
\boldsymbol{X} &= \sum_{i_1=1}^{k} [\boldsymbol{X}]_{i_1,1}[\boldsymbol{e}_{i_1}, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \\
&= \sum_{i_1=1}^{k} \sum_{i_2=1}^{k} [\boldsymbol{X}]_{i_1,1}[\boldsymbol{X}]_{i_2,2}[\boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \cdots, \boldsymbol{x}_N] \\
&\cdots \\
&= \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) [\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{e}_{i_N}].
\end{aligned} \quad (18)
$$

Moreover, for all $\boldsymbol{X} \in \Delta_k^N$,

$$
\sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} = \prod_{j=1}^{N} \sum_{i=1}^{k} [\boldsymbol{X}]_{i,j} = \prod_{j=1}^{N} 1 = 1. \quad (19)
$$

Combining (18) and (19), we conclude that every $\boldsymbol{X} \in \Delta_k^N$ can be expressed as a convex combination of matrices where each column vector has only one non-zero entry.

We are now ready to prove Theorem 1. To this end, we will that every optimal solution to (P4) is optimal for (P5), and vice versa. We denote the feasible regions of (P4) and (P5) by $\mathcal{X}$ and $\mathcal{X}_\rho$, respectively.

First, suppose $\boldsymbol{X}^*$ is an optimal solution to (P4). Clearly $\boldsymbol{X}^*$ is a feasible solution to (P5). By the concavity of $F(\boldsymbol{X}; \rho)$ from Lemma 2 and (18), for every feasible solution $\boldsymbol{X}$ to (P5),

$$
\begin{aligned}
&F(\boldsymbol{X}; \rho) \\
&\overset{(a)}{=} F\left(\sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) [\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{e}_{i_N}]; \rho \right) \\
&\overset{(b)}{\geq} \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) F([\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{e}_{i_N}]; \rho) \\
&\overset{(c)}{\geq} \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) \min_{\boldsymbol{X} \in \mathcal{X}} F(\boldsymbol{X}; \rho) \\
&\overset{(d)}{=} \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) \min_{\boldsymbol{X} \in \mathcal{X}} F(\boldsymbol{X}; 0) \\
&\overset{(e)}{=} \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}]_{i_j,j} \right) F(\boldsymbol{X}^*; 0) \\
&\overset{(f)}{=} F(\boldsymbol{X}^*; 0) \overset{(d)}{=} F(\boldsymbol{X}^*; \rho).
\end{aligned} \quad (20)
$$

Here (a) follows from (18); (b) is due to the concavity of $F(\boldsymbol{X}; \rho)$ from Lemma 2; (c) holds because $[\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{e}_{i_N}] \in \mathcal{X}$; (d) is due to $F(\boldsymbol{X}; \rho) = F(\boldsymbol{X}; 0)$ for all $\boldsymbol{X} \in \mathcal{X}$ and $\rho \geq 0$, as the penalized term $\sum_{i=1}^{N}(\|\boldsymbol{x}_i\|_1^2 - \|\boldsymbol{x}_i\|_2^2)$ becomes 0; (e) uses the global optimality of $\boldsymbol{X}_*$ in problem (P4); (f) follows from (19). Thus, (20) shows that every optimal solution to (P4) is optimal for (P5).

Now suppose that $\boldsymbol{X}^*$ is an optimal solution to (P5). We note that $\boldsymbol{X}^*$ must be a feasible solution to (P4). Otherwise

there would exist $i_1, ..., i_N$ such that $0 < \prod_{j=1}^{N} [\boldsymbol{X}^*]_{i_j,j} < 1$. Then,

$$
\begin{aligned}
&F(\boldsymbol{X}^*; \rho) \\
&\overset{(g)}{>} \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}^*]_{i_j,j} \right) F([\boldsymbol{e}_{i_1}, \cdots, \boldsymbol{e}_{i_N}]; \rho) \\
&\geq \sum_{i_1=1}^{k} \cdots \sum_{i_N=1}^{k} \left(\prod_{j=1}^{N} [\boldsymbol{X}^*]_{i_j,j} \right) \min_{\boldsymbol{X} \in \mathcal{X}_\rho} F(\boldsymbol{X}; \rho) \\
&\overset{(h)}{=} \min_{\boldsymbol{X} \in \mathcal{X}_\rho} F(\boldsymbol{X}; \rho),
\end{aligned} \quad (21)
$$

where (g) follows from the strictly concave of F from Lemma 2, and (h) is due to (19). This contradicts the optimality of $\boldsymbol{X}^*$. Thus, $\boldsymbol{X}^*$ is a feasible solution to (P4). (A quicker proof: every strictly concave function must attain optimum at extreme points of the convex domain, which correspond to the feasible region of (P4).)

For all feasible solutions $\boldsymbol{X}$ to (P4), we have $F(\boldsymbol{X}; \rho) = F(\boldsymbol{X}; 0)$, so

$$
F(\boldsymbol{X}^*; 0) = F(\boldsymbol{X}^*; \rho) \leq F(\boldsymbol{X}; \rho) = F(\boldsymbol{X}; 0),
$$

where the inequality is due to the fact that $\boldsymbol{X}^*$ is a feasible solution of (P4). This shows that $\boldsymbol{X}^*$ is an optimal solution of (P4). The proof is finished.

APPENDIX D
PROOF OF THEOREM 2

Before proceeding with the proof, we introduce an equivalent representation of the stationary point defined in Definition 1.

Lemma 3 (Equivalent Representation of the Stationary Point). *Define* $T_t^F(\boldsymbol{X}) := \arg\min_{\boldsymbol{Y} \in \Delta_k^N} \{ \langle t\boldsymbol{X}(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{Y} \rangle + \mathrm{KL}(\boldsymbol{Y}, \boldsymbol{X}) \}$. *Then* $\mathrm{KL}(T_t^F(\boldsymbol{X}^*), \boldsymbol{X}^*) = 0$ *if and only if* $\boldsymbol{X}^*$ *is a stationary point of problem P5 for all* $t \in (0, \frac{1}{L+1})$.

Proof. For $\boldsymbol{X}^*$ such that $\mathrm{KL}(T_t^F(\boldsymbol{X}^*), \boldsymbol{X}^*) = 0$, it also holds that

$$
\boldsymbol{X}^* = \arg\min_{\boldsymbol{X} \in \Delta_k^N} \{ \langle t\boldsymbol{X}^*(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{X} \rangle + \mathrm{KL}(\boldsymbol{X}, \boldsymbol{X}^*) \}.
$$

As the inner function is convex with respect to $\boldsymbol{X}$ and Δ_k^N is convex, the optimal $\boldsymbol{X}^*$ is a stationary point of this problem. Therefore, $\forall \boldsymbol{X} \in \Delta_k^N$,

$$
\left\langle \frac{\partial(\langle t\boldsymbol{X}^*(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{X} \rangle + \mathrm{KL}(\boldsymbol{X}, \boldsymbol{X}^*))}{\partial \boldsymbol{X}}|_{\boldsymbol{X}=\boldsymbol{X}^*}, \boldsymbol{X} - \boldsymbol{X}^* \right\rangle \geq 0.
$$

This implies

$$
\langle \boldsymbol{X}^*(2\boldsymbol{W} - \rho\boldsymbol{I}), \boldsymbol{X} - \boldsymbol{X}^* \rangle \geq 0,
$$

showing that $\boldsymbol{X}^*$ is a stationary point of problem P5. □

Then, we can introduce the following lemmas to analyze the benign properties of KL divergence and the objective function.

Lemma 4 (Benign Properties of KL Bregman Divergence [37]). *The KL Bregman divergence (8) can be expressed as*

$$
\mathrm{KL}(\boldsymbol{X}, \boldsymbol{Y}) = \omega(\boldsymbol{X}) - \omega(\boldsymbol{Y}) - \langle \nabla\omega(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y} \rangle, \quad (22)
$$

where

$$\omega(\boldsymbol{X}) = \sum_{i=1}^{k} \sum_{j=1}^{N} [\boldsymbol{X}]_{i,j} \log [\boldsymbol{X}]_{i,j}, \tag{23}$$

and $\boldsymbol{X}, \boldsymbol{X}^{(m)} \in \Delta_k^N$.

1) *The function* $\omega(\boldsymbol{X})$ *is 1-strongly convex on* Δ_k^N.
2) *The function* $\mathrm{KL}(\boldsymbol{X}, \boldsymbol{Y})$ *satisfies that* $\mathrm{KL}(\boldsymbol{X}, \boldsymbol{Y}) \geq \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2$.

Lemma 5 (Benign Properties of Objective Function). *The objective function* $F(\boldsymbol{X}; \rho)$ *in the penalized problem P5 satisfies following properties.*

1) *(L-smooth):* $F(\boldsymbol{X}; \rho)$ *is L-smooth over* Δ_k^N *with respect to* $\mathrm{KL}(\cdot)$ *in a sense that*

$$\begin{aligned} F(\boldsymbol{Y}; \rho) \leq &F(\boldsymbol{X}; \rho) + \langle \nabla F(\boldsymbol{X}; \rho), \boldsymbol{Y} - \boldsymbol{X} \rangle \\ &+ L \cdot \mathrm{KL}(\boldsymbol{Y}, \boldsymbol{X}), \end{aligned} \tag{24}$$

where $L = \|2\boldsymbol{W} - \rho \boldsymbol{I}\|_F$.

2) *(Bounded):* $F(\boldsymbol{X}; \rho)$ *is bounded over* Δ_k^N, *which is*

$$-\frac{\rho}{2} N \leq F(\boldsymbol{X}; \rho) \leq \|\boldsymbol{W}\|_{1,1}. \tag{25}$$

Proof. **Part 1.** Considering that $\frac{\partial F(\boldsymbol{X}; \rho)}{\partial \boldsymbol{X}} = \boldsymbol{X}(2\boldsymbol{W} - \rho \boldsymbol{I})$, we have

$$\begin{aligned} \|\nabla F(\boldsymbol{X}; \rho) - \nabla F(\boldsymbol{Y}; \rho)\|_F &= \|(\boldsymbol{X} - \boldsymbol{Y})(2\boldsymbol{W} - \rho \boldsymbol{I})\|_F \\ &\leq \|\boldsymbol{X} - \boldsymbol{Y}\|_F \|(2\boldsymbol{W} - \rho \boldsymbol{I})\|_F. \end{aligned}$$

Thus, $F(\boldsymbol{X}; \rho)$ is $\|(2\boldsymbol{W} - \rho \boldsymbol{I})\|_F$-smooth with respect to $\|\cdot\|_F$, which means

$$\begin{aligned} F(\boldsymbol{Y}; \rho) \leq &F(\boldsymbol{X}; \rho) + \langle \nabla F(\boldsymbol{X}; \rho), \boldsymbol{Y} - \boldsymbol{X} \rangle \\ &+ \frac{\|2\boldsymbol{W} - \rho \boldsymbol{I}\|_F}{2} \|Y - X\|_F^2 \\ \overset{(a)}{\leq} &F(\boldsymbol{X}; \rho) + \langle \nabla F(\boldsymbol{X}; \rho), \boldsymbol{Y} - \boldsymbol{X} \rangle \\ &+ \|2\boldsymbol{W} - \rho \boldsymbol{I}\|_F \mathrm{KL}(\boldsymbol{Y}, \boldsymbol{X}), \end{aligned}$$

where (a) is due to Lemma 4.

Therefore, $F(\boldsymbol{X}; \rho)$ is $\|2\boldsymbol{W} - \rho \boldsymbol{I}\|_F$-smooth with respect to $\mathrm{KL}(\cdot)$.

Part 2. For the upper bound, we derive

$$F(\boldsymbol{X}; \rho) \leq \mathrm{Tr}(\boldsymbol{X}\boldsymbol{W}\boldsymbol{X}^T) = \sum_{i=1}^{n} \sum_{j=1}^{n} [\boldsymbol{W}]_{i,j} x_i^T x_j \leq \|\boldsymbol{W}\|_{1,1}.$$

For the lower bound, we obtain

$$\begin{aligned} F(\boldsymbol{X}; \rho) &\geq -\frac{\rho}{2} \mathrm{Tr}(\boldsymbol{X}\boldsymbol{X}^T) = -\frac{\rho}{2} \sum_{i=1}^{N} \|\boldsymbol{x}_i\|_2^2 \\ &\geq -\frac{\rho}{2} \sum_{i=1}^{N} \|\boldsymbol{x}_i\|_1^2 = -\frac{\rho}{2} N. \end{aligned}$$

$\square$

Now, we establish a sufficient decrease lemma for MD.

Lemma 6 (Sufficient Decrease Lemma). *For the step size* $\{t_m\}$ *such that* $0 < t_m < \frac{1}{L+1}$, $\{\boldsymbol{X}^{(m)}\}_{m=0}^T$ *is sequence*

generated by MD. Then, $F(\boldsymbol{X}^{(m+1)}; \rho) \leq F(\boldsymbol{X}^{(m)}; \rho) - (\frac{1}{t_m} - L)\mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)})$.

Proof. We begin by proving a relevant inequality. Since ω is convex, we have

$$\omega(\boldsymbol{X}) \geq \omega(\boldsymbol{Y}) + \langle \nabla \omega(\boldsymbol{Y}), \boldsymbol{X} - \boldsymbol{Y} \rangle, \quad \forall \boldsymbol{X}, \boldsymbol{Y} \in \Delta_k^N. \tag{26}$$

This inequality leads to

$$\langle \nabla \omega(\boldsymbol{Y}) - \nabla \omega(\boldsymbol{X}), \boldsymbol{Y} - \boldsymbol{X} \rangle \geq \mathrm{KL}(\boldsymbol{Y}, \boldsymbol{X}), \forall \boldsymbol{X}, \boldsymbol{Y} \in \Delta_k^N. \tag{27}$$

Considering that each iteration (7) is convex with respect to $\boldsymbol{X}$ and Δ_k^N is convex, the optimal $\boldsymbol{X}^{(m+1)}$ is a stationary point of the problem, implying that $\forall \boldsymbol{X} \in \Delta_k^N$,

$$\begin{aligned} \Big\langle t_m \nabla F(\boldsymbol{X}^{(m)}; \rho) + \nabla \omega(\boldsymbol{X}^{(m+1)}) - \nabla \omega(\boldsymbol{X}^{(m)}), \\ \boldsymbol{X} - \boldsymbol{X}^{(m+1)} \Big\rangle \geq 0. \end{aligned} \tag{28}$$

Substituting $\boldsymbol{X} = \boldsymbol{X}^{(m)}$ into (28) and applying (27), we obtain

$$\begin{aligned} \Big\langle \nabla F(\boldsymbol{X}^{(m)}; \rho), \boldsymbol{X}^{(m+1)} - \boldsymbol{X}^{(m)} \Big\rangle \\ \leq -\frac{1}{t_m} \mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}). \end{aligned} \tag{29}$$

Since $F(\boldsymbol{X}; \rho)$ is L-smooth, combining (24) and (29), we have

$$\begin{aligned} F(\boldsymbol{X}^{(m+1)}; \rho) \leq &F(\boldsymbol{X}^{(m)}; \rho) \\ &- \left(\frac{1}{t_m} - L\right) \mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}). \end{aligned} \tag{30}$$

$\square$

Finally, by applying the sufficient decrease lemma and summing (30) from 0 to $T - 1$, we obtain:

$$\begin{aligned} \sum_{m=0}^{T-1} \left(\frac{1}{t_m} - L\right) \mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}) \\ \leq F(\boldsymbol{X}^{(0)}; \rho) - F(\boldsymbol{X}^{(T)}; \rho). \end{aligned} \tag{31}$$

which indicates

$$\begin{aligned} \min_{m=0, \cdots, T-1} \mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}) \sum_{m=0}^{T-1} \left(\frac{1}{t_m} - L\right) \\ \leq F_{\max} - F_{\min}. \end{aligned} \tag{32}$$

By Lemma 5, supplementing $F_{\max} = \|\boldsymbol{W}\|_{1,1}$, $F_{\min} = -\frac{\rho N}{2}$, and choosing $t_m = \frac{1}{L+m+1}$, we derive

$$\begin{aligned} \min_{m=0, \cdots, T-1} \mathrm{KL}(\boldsymbol{X}^{(m+1)}, \boldsymbol{X}^{(m)}) \\ \leq \frac{2\|\boldsymbol{W}\|_{1,1} + \rho N}{T(T+1)} \leq \epsilon. \end{aligned} \tag{33}$$

Finally, we obtain the convergence rate as

$$T \geq \sqrt{\frac{2\|\boldsymbol{W}\|_{1,1} + \rho N}{\epsilon}} \approx \mathcal{O}\left(\sqrt{\frac{1}{\epsilon}}\right). \tag{34}$$

REFERENCES

[1] R. W. Heath, G. Laus, T. Q. Quek, S. Talwar, and P. Zhou, "Signal processing for the 5G revolution," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 12–13, 2014, doi:10.1109/MSP.2014.2345430.

[2] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, 2017, doi:10.1109/JSAC.2017.2692307.

[3] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015, doi:10.1109/ACCESS.2015.2461602.

[4] G. Liu, Y. Huang, Z. Chen, L. Liu, Q. Wang, and N. Li, "5G deployment: Standalone vs. non-standalone from the operator perspective," *IEEE Commun. Mag.*, vol. 58, no. 11, pp. 83–89, 2020, doi:10.1109/MCOM.001.2000230.

[5] L. Pierucci, "The quality of experience perspective toward 5g technology," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 10–16, 2015, doi:10.1109/MWC.2015.7224722.

[6] Z.-Q. Luo, X. Zheng, D. López-Pérez, Q. Yan, X. Chen, N. Wang, Q. Shi, T.-H. Chang, and A. Garcia-Rodriguez, "SRCON: A data-driven network performance simulator for real-world wireless networks," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 96–102, 2023, doi:10.1109/MCOM.001.2200179.

[7] Y.-F. Liu, T.-H. Chang, M. Hong, Z. Wu, A. Man-Cho So, E. A. Jorswieck, and W. Yu, "A survey of recent advances in optimization methods for wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 11, pp. 2992–3031, 2024, doi:10.1109/JSAC.2024.3443759.

[8] J. Gui, Z. Jiang, and S. Gao, "PCI planning based on binary quadratic programming in LTE/LTE-A networks," *IEEE Access*, vol. 7, pp. 203–214, 2019, doi:10.1109/ACCESS.2018.2885313.

[9] S. Seetharaman and D. Krishnaswamy, "A programmable distributed optimization framework for 5G," *CSI Trans. ICT*, vol. 8, no. 1, pp. 65–68, 2020, doi:10.1007/s40012-020-00271-x.

[10] H. Yang, A. Huang, R. Gao, T. Chang, and L. Xie, "Interference self-coordination: A proposal to enhance reliability of system-level information in OFDM-based mobile networks via PCI planning," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 1874–1887, 2014, doi:10.1109/TWC.2014.030514.130447.

[11] T. Bandh, G. Carle, and H. Sanneck, "Graph coloring based physical-cell-ID assignment for LTE networks," in *Proc. of the 2009 Int. Conf. Wireless Commun. and Mobile Comput.: Connecting the world wirelessly*, ser. IWCMC '09. Association for Computing Machinery, 2009, p. 116–120, doi:10.1145/1582379.1582406.

[12] A. Pratap, R. Misra, and U. Gupta, "Randomized graph coloring algorithm for physical cell id assignment in lte-a femtocellular networks," *Wireless Pers. Commun.*, vol. 91, pp. 1213–1235, 2016, doi:10.1007/s11277-016-3522-3.

[13] P. Li and J. Wang, "PCI planning method based on genetic algorithm in LTE network," *Telecommun. Sci.*, vol. 32, no. 3, p. 2016082, 2016, doi:10.11959/j.issn.1000-0801.2016082.

[14] A. Shen, B. Guo, Y. Gao, T. Xie, X. Hu, Y. Zhang, J. Shen, Y. Fang, G. Wang, and Y. Liu, "A novel PCI optimization method in LTE system based on intelligent genetic algorithm," in *Int. Conf. Signal and Inf. Process., Netw. And Comput.* Springer, 2017, pp. 350–355, doi:10.1007/978-981-10-7521-6_42.

[15] C. E. Andrade, L. S. Pessoa, and S. Stawiarski, "The physical cell identity assignment problem: A practical optimization approach," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 282–292, 2024, doi:10.1109/TEVC.2022.3185927.

[16] J. Fairbrother, A. N. Letchford, and K. Briggs, "A two-level graph partitioning problem arising in mobile wireless communications," *Comput. Optim. and Appl.*, vol. 69, pp. 653–676, 2018, doi:10.1007/s10589-017-9967-9.

[17] Huawei Technologies Co., Ltd., "5G PCI and PRACH planning," https://www.scribd.com/document/473706035/5G-PCI-and-PRACH-Planning-pdf.

[18] Q. Zhang, S. Chen, j. Sui, Z. Yao, S. Zhang, Y. Liu, and X. Wang, "PCI optimization method and apparatus," China Patent CN105 744 548A, 2014.

[19] M. Ruan, "Relation matrix establishment system and method and PCI intelligent optimization system and method," China Patent CN103 906 078A, 2014.

[20] C. Ding, D. Pei, and A. Salomaa, *Chinese remainder theorem: applications in computing, coding, cryptography*. USA: World Scientific Publishing Co., Inc., 1996, doi:10.1142/3254.

[21] E. Boros and P. L. Hammer, "The max-cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds," *Ann. Operations Res.*, vol. 33, no. 3, pp. 151–180, 1991, doi:10.1007/BF02115753.

[22] V. Kann, S. Khanna, J. Lagergren, and A. Panconesi, "On the hardness of approximating max k-cut and its dual," in *Int. Seminar Theor. Comput. Sci.*, 1996, pp. 61–67.

[23] F. Ma and J.-K. Hao, "A multiple search operator heuristic for the max-k-cut problem," *Ann. Operations Res.*, vol. 248, pp. 365–403, 2017, doi:10.1007/s10479-016-2234-0.

[24] G. G. Subies, "Welsh powell algorithm to color graphs," https://gist.github.com/GuillemGSubies/59b968ec0a68c17384c9f15c15cb38e6.

[25] Y. Qiu, C. Huang, Y. Xue, Z. Jiang, Q. Shi, D. Zhang, and Z.-Q. Luo, "Relaxation-free min-k-partition for PCI assignment in 5G networks," *arXiv preprint arXiv:2506.10362*, 2025, doi:10.48550/arXiv.2506.10362.

[26] C. Delorme and S. Poljak, "Laplacian eigenvalues and the maximum cut problem," *Math. Program.*, vol. 62, pp. 557–574, 1993, doi:10.1007/BF01585184.

[27] L. Lovász, "On the shannon capacity of a graph," *IEEE Trans. Inf. Theory*, vol. 25, no. 1, pp. 1–7, 1979, doi:10.1109/TIT.1979.1055985.

[28] L. Lovász and A. Schrijver, "Cones of matrices and set-functions and 0–1 optimization," *SIAM J. Optim.*, vol. 1, no. 2, pp. 166–190, 1991, doi:10.1137/0801013.

[29] S. Poljak and F. Rendl, "Nonpolyhedral relaxations of graph-bisection problems," *SIAM J. Optim.*, vol. 5, no. 3, pp. 467–487, 1995, doi:10.1137/0805024.

[30] N. Z. Shor, "Quadratic optimization problems," *Sov. J. Comput. and Syst. Sci.*, vol. 25, pp. 1–11, 1987.

[31] S. Burer, R. D. Monteiro, and Y. Zhang, "Rank-two relaxation heuristics for max-cut and other binary quadratic programs," *SIAM J. Optim.*, vol. 12, no. 2, pp. 503–521, 2002, doi:10.1137/S1052623400382467.

[32] S. Wang, T.-H. Chang, Y. Cui, and J.-S. Pang, "Clustering by orthogonal nmf model and non-convex penalty optimization," *IEEE Trans. Signal Process.*, vol. 69, pp. 5273–5288, 2021, doi:10.1109/TSP.2021.3102106.

[33] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999, doi:10.1007/978-0-387-40065-5.

[34] I. W. Selesnick and I. Bayram, "Sparse signal estimation by maximally sparse convex optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1078–1092, 2014, doi:10.1109/TSP.2014.2298839.

[35] Q. Shi and M. Hong, "Penalty dual decomposition method for nonsmooth nonconvex optimization—part I: Algorithms and convergence analysis," *IEEE Trans. Signal Process.*, vol. 68, pp. 4108–4122, 2020.

[36] Q. Shi, M. Hong, X. Fu, and T.-H. Chang, "Penalty dual decomposition method for nonsmooth nonconvex optimization—part II: Applications," *IEEE Trans. Signal Process.*, vol. 68, pp. 4242–4257, 2020, doi:10.1109/TSP.2020.3001397.

[37] A. Beck, *First-order methods in optimization*. SIAM, 2017, doi:10.1137/1.9781611974997.

[38] S. Zhang and N. He, "On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization," *arXiv preprint arXiv:1806.04781*, 2018, doi:10.48550/arXiv.1806.04781.

[39] Z. Zhou, P. Mertikopoulos, N. Bambos, S. Boyd, and P. W. Glynn, "Stochastic mirror descent in variationally coherent optimization problems," in *Advances in Neural Inf. Process. Syst.*, vol. 30. Curran Associates, Inc., 2017.

[40] Z. Zhou, P. Mertikopoulos, N. Bambos, S. Boyd, and P. Glynn, "Mirror descent in non-convex stochastic programming," *arXiv preprint arXiv:1706.05681*, 2017, doi:10.48550/arXiv.1706.05681.

[41] C. E. Andrade, M. G. Resende, W. Zhang, R. K. Sinha, K. C. Reichmann, R. D. Doverspike, and F. K. Miyazawa, "A biased random-key genetic algorithm for wireless backhaul network design," *Appl. Soft Comput.*, vol. 33, pp. 150–169, 2015, doi:10.1016/j.asoc.2015.04.016.

[42] R. Gowaikar, B. Hochwald, and B. Hassibi, "Communication over a wireless network with random connections," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2857–2871, 2006, doi:10.1109/TIT.2006.876254.

[43] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1029–1046, 2009, doi:10.1109/JSAC.2009.090902.

Supplementary Materials of "Relaxation-Free Min-k-Partition for PCI Assignment in 5G Networks"

Yeqing Qiu, Chengpiao Huang, Ye Xue, Zhipeng Jiang, Dong Zhang, and Zhi-Quan Luo

A THE FULL ALGORITHM OF THE GREEDY QUOTIENT RE-ASSIGNMENT FOR PCI RANGE CONSTRAINT SATISFACTION

Algorithm 3: Greedy Quotient Re-assignment for PCI Range Constraint Satisfaction

---

Data: Set of neighboring and second-order neighboring relations $\mathcal{E}$, mod-30 clusters $\{\mathcal{C}_r\}_{r=0}^{29}$, sets of constraint-violating cells $\{\mathcal{I}_r\}_{r=0}^{29}$ within the mod-30 clusters, mod-30 value assignment $\boldsymbol{r} \in \mathbb{Z}_{30}^N$, current quotient assignment $\boldsymbol{q}$.

Result: Re-assigned quotient $\widetilde{\boldsymbol{q}}$.

for $r = 0$ **to** 29 **do**
  for $i \in \mathcal{I}_r$ **do**
    $\deg_i \leftarrow |\{j \in \mathcal{C}_r : (i,j) \in \mathcal{E}, q_i = q_j\}|$
  Sort $\mathcal{I}_r$ in decreasing order of $\{\deg_v\}_{v \in \mathcal{I}_r}$
  $\mathcal{J}_r \leftarrow \mathcal{C}_r \backslash \mathcal{I}_r$ // *set of constraint-satisfying cells*
  for $i \in \mathcal{J}_r$ **do**
    $\widetilde{q}_i \leftarrow q_i$
  for $i \in \mathcal{I}_r$ **do**
    $\widetilde{q}_i \leftarrow \underset{q : 30q + r_i \leq 1007}{\arg\min} |\{j \in \mathcal{J}_r : (i,j) \in \mathcal{E}, q = \widetilde{q}_j\}|$
    $\mathcal{J}_r \leftarrow \mathcal{J}_r \cup \{i\}$

return $\widetilde{\boldsymbol{q}} = \{\widetilde{q}_i\}_{i \in \mathcal{V}}$

B ABLATION STUDY ON HYPERPARAMETERS

To assess the robustness of the hyperparameters, we conducted an ablation study on $\mathrm{RGG}(200, 0.3)$ graphs by scaling each hyperparameter up and down. As shown in Table VI, our methods remain stable under such perturbations.

TABLE VI: Sensitivity analysis of GP-PMD on $(N = 200, r = 0.3)$ RGG instances. The original setting is $\epsilon_1 = 10^{-5}, \epsilon_2 = 10^{-10}, \gamma = 1.1, \rho_0 = 10^{-8}$. We perturb one parameter each time and test the overall performance.

Setting	Collision ↓	Confusion ↓	Mod-3 ↓	Mod-30 ↓	Time (s) ↓
Original	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	86.7920 ± 59.3875	195.6269 ± 15.6531
$\epsilon_1 \times 10$	0.0000 ± 0.0000	0.0000 ± 0.0000	31059.7002 ± 1050.3168	117.8055 ± 57.7300	200.6586 ± 21.1356
$\epsilon_1 \times 0.1$	0.0000 ± 0.0000	0.0000 ± 0.0000	31044.0332 ± 1046.6022	73.9605 ± 52.5553	200.6314 ± 14.4452
$\epsilon_2 \times 10$	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	86.7920 ± 59.3875	195.9044 ± 16.3598
$\epsilon_3 \times 0.1$	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	86.7920 ± 59.3875	195.7012 ± 15.9792
$\gamma \times 0.918$	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	84.6270 ± 57.6500	202.4566 ± 15.7155
$\gamma \times 1.1$	0.0000 ± 0.0000	0.0000 ± 0.0000	31068.1502 ± 1053.6886	90.6242 ± 58.4496	194.0473 ± 16.5126
$\rho_0 \times 10$	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	86.7920 ± 59.3875	196.0779 ± 16.4285
$\rho_0 \times 0.1$	0.0000 ± 0.0000	0.0000 ± 0.0000	31065.5022 ± 1052.4893	86.7920 ± 59.3875	197.1436 ± 16.4936

C DETAILED RESULTS ON SYNTHETIC DATA

To provide a more comprehensive view of the experimental results on synthetic data, we present detailed box plots in Figs. 5, 6, 7, 8, and 9, corresponding respectively to collision, confusion, mod-3 interference, mod-30 interference, and computational time. Note that the absence of visible boxes in the plots indicates that the corresponding metric attains zero across all test cases, reflecting a consistently optimal outcome for that criterion.
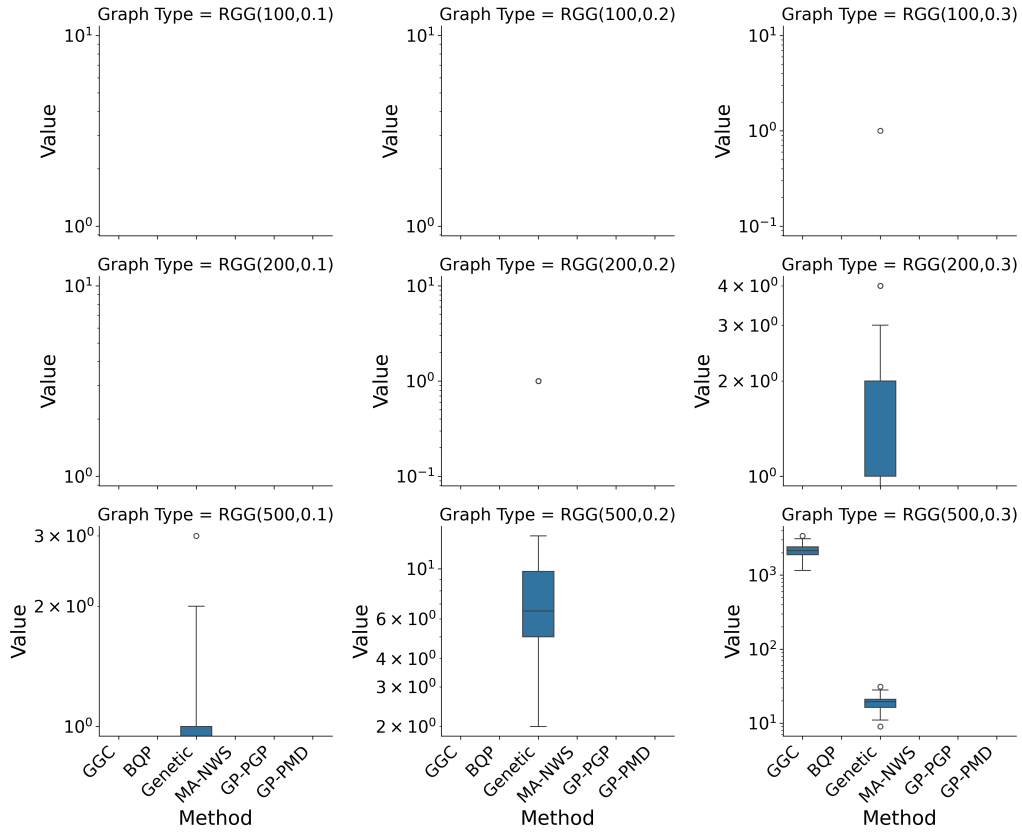
Fig. 5: Boxplot of collision across methods and graph types.
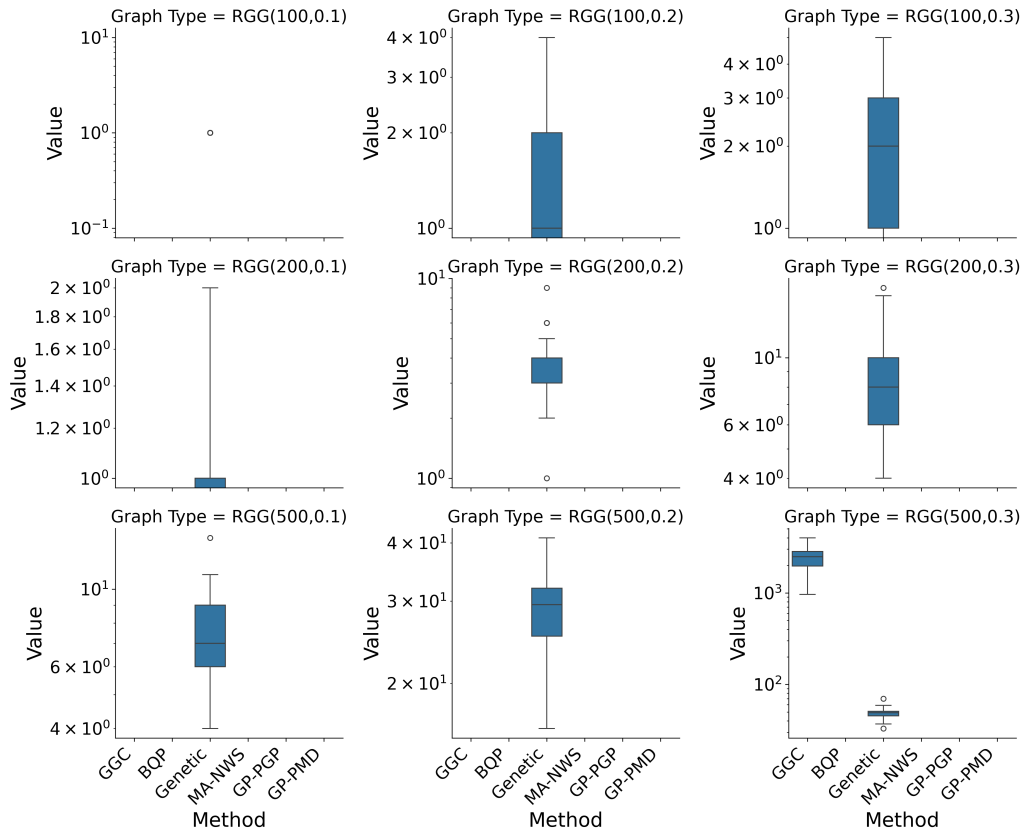
Fig. 6: Boxplot of confusion across methods and graph types.
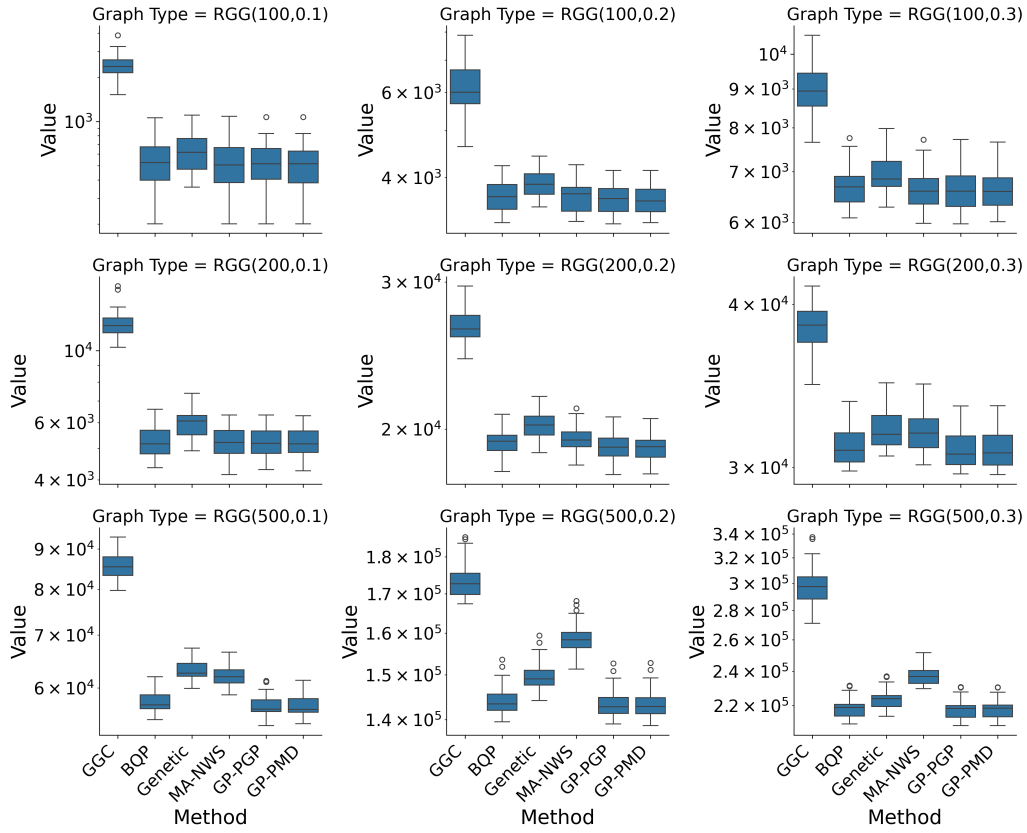
Fig. 7: Boxplot of mod-3 interference across methods and graph types.
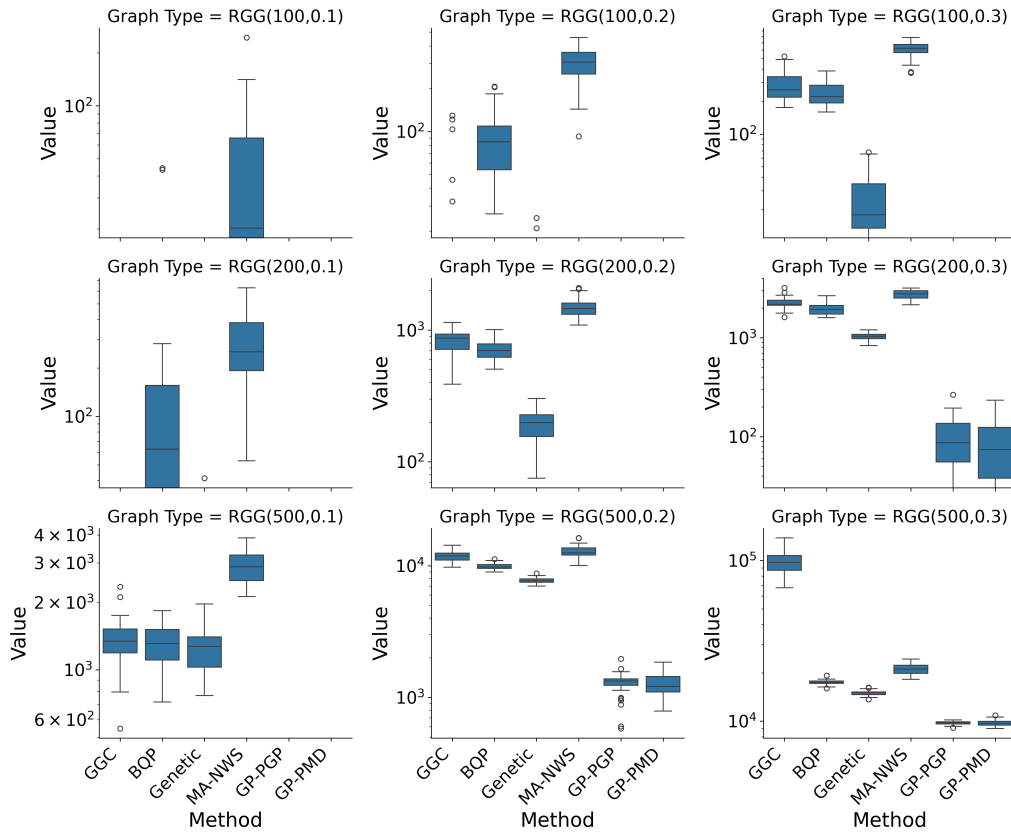
Fig. 8: Boxplot of mod-30 interference across methods and graph types.
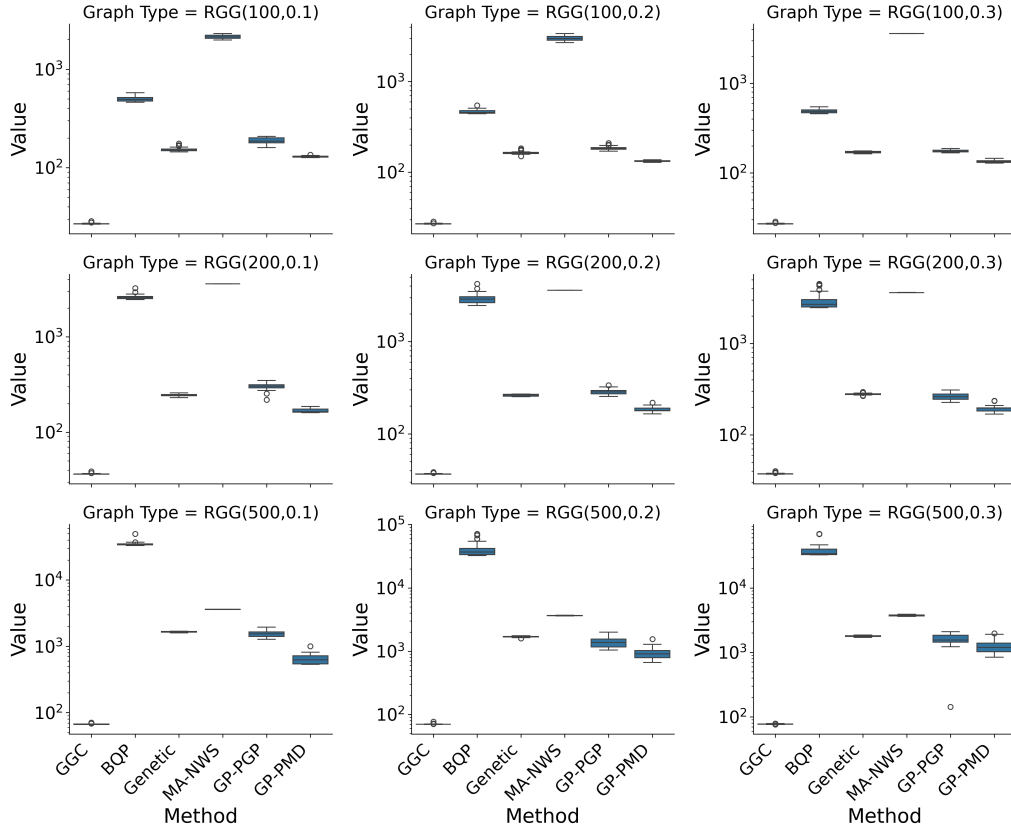
Fig. 9: Boxplot of computational time across methods and graph types.

D DETAILED RESULTS OF STATISTICAL HYPOTHESIS TESTS

To assess statistical significance, we applied the non-parametric Mann-Whitney U test. A p-value less than 0.05 indicates significant performance differences between algorithms at the 0.05 significance level. Moreover, to quantify the strength of these differences, we employed the Vargha-Delaney A test, which estimates the probability that one method outperforms another. An A-value above 0.5 favors our method (GP-PMD); $A = 1$ implies complete dominance and $A = 0.5$ indicates no significant difference. Tables in Figs. 10, 11, 12, 13, 14 report p-value and A statistics for each algorithm pair across all graph types and objectives. Results confirm that GP-PMD statistically and practically outperforms all baseline methods.

RGG(100, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	1.00	0.50	Equal
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(100, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	1.00	0.50	Equal
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(100, 0.3)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	0.33	0.52	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	1.00	0.50	Equal
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	0.01	0.60	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.3)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	0.92	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	0.72	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

Fig. 10: GP-PMD versus other algorithms on number of collisions

RGG(100, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	0.01	0.60	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(100, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	0.85	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(100, 0.3)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	0.93	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	1.00	0.50	Equal
BQP	$\ll 0.01$	0.72	GP-PMD
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	0.98	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(200, 0.3)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.2)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

RGG(500, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	1.00	0.50	Equal
MA-NWS	1.00	0.50	Equal
GP-PGP	1.00	0.50	Equal

Fig. 11: GP-PMD versus other algorithms on number of confusions

RGG(100, 0.1)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	0.07	0.64	GP-PMD
BQP	0.77	0.52	GP-PMD
MA-NWS	0.90	0.51	GP-PMD
GP-PGP	0.92	0.51	GP-PMD

RGG(100, 0.2)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.74	GP-PMD
BQP	0.38	0.57	GP-PMD
MA-NWS	0.46	0.56	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(100, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	0.01	0.71	GP-PMD
BQP	0.49	0.55	GP-PMD
MA-NWS	0.96	0.50	Equal
GP-PGP	0.98	0.50	Equal

RGG(200, 0.1)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	0.86	0.51	GP-PMD
MA-NWS	0.95	0.51	GP-PMD
GP-PGP	0.92	0.51	GP-PMD

RGG(200, 0.2)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.86	GP-PMD
BQP	0.23	0.59	GP-PMD
MA-NWS	0.04	0.66	GP-PMD
GP-PGP	0.91	0.51	GP-PMD

RGG(200, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.76	GP-PMD
BQP	0.41	0.56	GP-PMD
MA-NWS	$\ll 0.01$	0.73	GP-PMD
GP-PGP	0.98	0.50	Equal

RGG(500, 0.1)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.99	GP-PMD
BQP	0.08	0.63	GP-PMD
MA-NWS	$\ll 0.01$	0.97	GP-PMD
GP-PGP	0.98	0.50	Equal

RGG(500, 0.2)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.90	GP-PMD
BQP	0.29	0.58	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	0.99	0.50	Equal

RGG(500, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.78	GP-PMD
BQP	0.46	0.56	GP-PMD
MA-NWS	$\ll 0.01$	0.99	GP-PMD
GP-PGP	0.99	0.50	Equal

Fig. 12: GP-PMD versus other algorithms on mod-3 interference

RGG(100, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	1.00	0.50	Equal
BQP	0.16	0.53	GP-PMD
MA-NWS	$\ll 0.01$	0.75	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(100, 0.2)

Method	p-value	A	Winner
GGC	0.02	0.58	GP-PMD
Genetic	0.16	0.58	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(100, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	0.92	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(200, 0.1)

Method	p-value	A	Winner
GGC	1.00	0.50	Equal
Genetic	0.33	0.52	GP-PMD
BQP	$\ll 0.01$	0.83	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(200, 0.2)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(200, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	0.43	0.56	GP-PMD

RGG(500, 0.1)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	1.00	0.50	Equal

RGG(500, 0.2)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	0.31	0.58	GP-PMD

RGG(500, 0.3)

Method	p-value	A	Winner
GGC	$\ll 0.01$	1.00	GP-PMD
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	0.78	0.52	GP-PMD

Fig. 13: GP-PMD versus other algorithms on mod-30 interference

RGG(100, 0.1)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(100, 0.2)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(100, 0.3)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(200, 0.1)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(200, 0.2)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(200, 0.3)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	0.99	GP-PMD

RGG(500, 0.1)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	1.00	GP-PMD

RGG(500, 0.2)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	1.00	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	0.94	GP-PMD

RGG(500, 0.3)			
Method	p-value	A	Winner
GGC	$\ll 0.01$	0.00	GGC
Genetic	$\ll 0.01$	0.90	GP-PMD
BQP	$\ll 0.01$	1.00	GP-PMD
MA-NWS	$\ll 0.01$	1.00	GP-PMD
GP-PGP	$\ll 0.01$	0.83	GP-PMD

Fig. 14: GP-PMD versus other algorithms on computational time