How Much Frequency Can Be Reused in 5G Cellular Networks—A Matrix Graph Model

Yaoqing Yang, Student Member, IEEE, Bo Bai, Member, IEEE, and Wei Chen, Senior Member, IEEE

Abstract—The 5th Generation cellular network may have the key feature of smaller cell size and denser resource employment, resulted from diminishing resource and increasing communication demands. However, small cell may result in high interference between cells. Moreover, the random geographic patterns of small cell networks make them hard to analyze, at least excluding schemes in the well-accepted hexagonal grid model. In this paper, a new model—the matrix graph is proposed which takes advantage of the small cell size and high inter-cell interference to reduce computation complexity. This model can simulate real world networks accurately and offers convenience in frequency allocation problems which are usually NP-complete. An algorithm dealing with this model is also given, which asymptotically achieves the theoretical limit of frequency allocation, and has a complexity which decreases with cell size and grows linearly with the network size. This new model is specifically proposed to characterize the next-generation cellular networks.

Index Terms—Matrix Graph, Cellular Network, Frequency Reuse, Graph Coloring.

I. INTRODUCTION

FREQUENCY reuse and the cellular concept [1] is the driven force behind several decades of innovations in the wireless communication field. Many pioneering works [2][3][4] are based on the convenient assumptions that cells, frequency reuse patterns and even user demands, are geographically periodic, often simulated by a regular hexagonal grid model. However, these assumptions have long been suspicious by research simulations and industry practices [1][5]. A recent survey [7] suggested that the existing 4G and future mobile networks may actually have a geographic pattern which falls in between the regular grid model and a totally random graph. This phenomenon indicates the necessity of a new cellular model.

Moreover, the recently emerging 5th generation cellular networks concept also poses new challenges to the conventional models. First, 5G networks are believed to have small cells [7][8]. Small cells are advantageous in dramatically higher energy efficiency and indoor coverage, and are viewed as the promising candidate for the future green and efficient communications. Second, 5G networks approved of advanced interference management schemes. Smaller cells cause higher inter-cell interference. This means that the classic one-base-station-downlink model [9] cannot be used here. Many emerging techniques are devoted to this problem, e.g., Fractional Frequency Reuse [6] and other interference management schemes; multi-cell coordination, e.g. CoMP [10] and Multicell Cooperations [11]. Thus, we would like to answer the question: how much frequency can be reused in a highly cooperative and high-interference small-cell network?

In this paper, we propose a new model called the matrix graph to answer this question. A matrix graph is a lattice-like conflict graph while each lattice point is substituted by a small random graph called a small cell. The vertices in the graph represent communication links [17][19], i.e. either uplink or downlink, while the edges represent interference. Confliction graph is widely adopted in cellular communications [13][20] and frequency allocation in a conflict graph can be conveniently treated as multi-coloring problems. We still consider coloring, i.e., frequency allocation, in matrix graphs. But we then show why this lattice-like matrix graph is especially suitable to deal with frequency allocation in the 5G network.

In the matrix graph model, we make the cell shapes and sizes random, but still reserve a lattice pattern. As stated above, this matches the real 4G cellular network structure shown in [7]. Thus, the first merit of the matrix graph model is its high resemblance to real-world networks. One could generally view this model as an eclectic model between a strict grid and a totally random graph. The second advantage of the matrix graph is that it is convenient in a high-interference small-cell network. As shown in the analysis part, if we increase the inter-cell interference and reduce the cell size, the computation complexity of multi-coloring will be lower. Preceding works widely recognized the trend of small cells, but seldom did they actively design network models and algorithms to meet this trend.

The third virtue of the matrix graph, compared to other graph-based models [13][18], is being mathematically tractable to reach the fundamental limit of frequency allocation. Although in this paper, obtaining the optimal frequency allocation in a matrix graph is proved to be NP-complete, we still obtained a linear-time approximation algorithm with a solution guaranteed to converge to the optimal value. This means that for the small-cell network, we can directly tell how much frequency can be reused as long as the corresponding matrix graph has been properly constructed. This is in contrast with frequency allocation in general graphs. In fact, frequency allocation problems, like multi-coloring and the related Maximum Weighted Independent Set (MWIS) problems are excluded from APX-complete problems [20][21], which means that even...
making a performance-guaranteed approximation is NP Hard. That is why previous works on coloring-based heuristics often lack analytical results. Moreover, it is showed in this paper that coloring a one-dimensional matrix graph has linear-time solution. This property gives us huge flexibility in tackling two-dimensional matrix graphs. We will call a one-dimensional matrix graph a Vector Graph. This model itself is also important, because large networks can be one-dimensional, e.g., a femtocell network along a long road or a wifi-network in a long train.

In this paper, the final goal is to achieve the best reuse-interference tradeoff, i.e., obtaining the maximum weighted sum of frequency used by each communication link without interference. This is often called the Maximum Service Frequency Allocation (MSFA) and has been widely accepted as a benchmark of efficiency, e.g., see survey [18]. Our method does not rely on specific resource type. For simplicity we assume resources to be frequency bands or OFDMA subcarriers. The only requirement is that any two resources are orthogonal and any resource cannot be reused by interfering communication links. There are both works on assuming links [17][19] or User Terminals [15][16] as confliction agents. We follow the first one because in a 5G network, there may be cooperations between cells and thus, one UT may have a few communication links. Also, we assume that all the heterogeneous base stations are linked to the central network with wired backhaul [7][10][11]. Thus, scheduling can be carried out in the whole network. This large-scale scheduling only incurs an $O(MN)$ overhead where the network size is $M$-by-$N$. So it prevails exact algorithms which usually have exponential complexity. Briefly summarizing our contributions in this paper, we

- originally designed a graph model suitable in small cells, which complements the insufficiencies of hexagonal grid models and conflict graph models;
- designed algorithms to allocate frequencies efficiently, with a computation complexity growing linearly with network size and decreases polynomially with cell size;
- obtained sufficient numerical results to support our analysis.

The paper is arranged as follows: in Section II, the matrix graph system modelling issues are covered; in Section III, a high-efficiency and low-complexity scheduling algorithm is thoroughly proposed; Section IV discusses simulation results.

II. System Model and Problem Formulation

In this section, the matrix graph and the corresponding frequency allocation problem are proposed. First the traditional conflict graph model is reviewed (see Fig. 1). Then key procedures are presented to translate conflict graphs into matrix graphs (see Fig. 2).

A. Frequency Allocation on a Confliction Graph

Fig. 1(a) is an imaginary scenario of the future 5G small-cell cellular network. A graph $G = (V, E)$ (as shown in Fig. 1(b)) is constructed to represent the network. Each vertex $v \in V$ denotes a communication link and each edge $e \in E$ represents a confliction between two neighboring communication links. We consider communication links, rather than User Terminals (UTs), as the conflicting agents [17][19]. The term vertex and communication links will be used interchangeably. Assume there are $C$ colors $\Lambda = \{1, 2, ..., C\}$ to allocate. Each color $c$ can be treated as a frequency band. A mapping $\mathcal{C}$, called multi-coloring, can be used to represent frequency allocation.

$$\mathcal{C}(v, \lambda) = \begin{cases} 1, & \text{when } c \text{ is allocated to } v, \\ 0, & \text{otherwise.} \end{cases}$$

We write the reuse ratio

$$f_v = \frac{1}{C} \sum_{c=1}^{C} \mathcal{C}(v, c) \mu(v, c)$$

for the portion of the total frequency bands used by $v$. $\mu(v, c)$ stands for the color weight of $c$ when used by communication link $v$ and will be discussed in Remark 1. Then the conventional frequency allocation problem can be written as a multi-coloring problem:

$$\max \bar{f} = \left( \sum_{v \in V} w_v f_v \right) / \left( \sum_{v \in V} w_v \right),$$

s.t. $\mathcal{C}(v_1, c) + \mathcal{C}(v_2, c) \leq 1$, if $(v_1, v_2) \in E$.

where $w_v$ to denotes the vertex weight which is explained in Remark 1. The constraint ensures that no conflicting links are assigned the same frequency band. Note that multi-coloring, compared to coloring, allows each vertex to be
available frequency bands that is available at the central node. However, we can also assume we only consider large-scale fading so that this information factor \[1\] \[2\] \[6\]. Therefore, optimizing channel capacity becomes optimizing channel capacity. Here \(P(v,c)\) and \(h(v,c)\) denotes the transmit power and channel gain of communication link \(v\) on frequency band \(c\). We only consider large-scale fading so that this information is available at the central node. However, we can also assume \(\mu(v,c) = 1\). Then \(f_v\) is exactly the ratio of the available frequency bands that \(v\) can utilize. Optimizing \(f_v\) now is the same as the Maximum Service Frequency Allocation [18] or the inverse of the classic co-channel reuse factor [1][2][6]. Therefore, optimizing \(f_v\) is consistent with multiple classic optimization problems in communications.

B. The Matrix Graph Model in Small-Cell Networks

In this section we show that in small-cell networks, a confliction graph can be transformed into a matrix graph. Fig. 2(a) shows a typical small-cell conflict graph. Each cell has \(1 \sim 3\) communication links. However, due to small cell size and the effects of heterogeneous multi-layer networks, inter-cell and intra-cell interference is quite complicated. This motivates us to transform this kind of graphs into a general graph structure.

The new structure is a matrix graph (as shown in Fig. 2(b)). Although the cell shapes and sizes in Fig. 2 can be random, a shape or size parameter regarding all cells should be characterized by a universal probabilistic distribution. Assume each cell has a height \(a\) and a width \(b\). Denote the average height and width by \(\bar{a}\) and \(\bar{b}\), respectively. If we draw \(M+1\) parallel lines horizontally with distance \(\bar{a}\) and draw \(N+1\) parallel lines vertically with distance \(\bar{b}\) on Fig. 2(a), we can partition the space into \(M N\) squares, with approximately one cell lying in one square. Meanwhile, vertices are separated into each square

\[
V = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} V_{m,n}, \tag{4}
\]

where \(V_{m,n}\) denotes the vertices in the square constituted by lattice points \(\{(m-1, n-1), (m,n-1), (m-1,n), (m,n)\}\) and

\[
V_{m,n} = \{v_i^{m,n}\}_{i=1}^{l_{m,n}}. \tag{5}
\]

Here \(l_{m,n}\) is the number of vertices in in this square and \(v_i^{m,n}\) means the \(i\)th vertex. We use \(G_{m,n}\) to denote the induced graph by \(V_{m,n}\) from \(G\), which means that \(G_{m,n} = (V_{m,n}, E_{m,n})\) and \(E_{m,n} = \{e \in E | e = (v_i^{m,n}, v_j^{m,n}), 1 \leq i < j \leq l_{m,n}\}\). Each \(G_{m,n}\) is denoted by a circle in Fig. 2(b). From now on, \(G_{m,n}\) will also be called a cell.

Then we reserve all edges from \(G\), except those edges that connect non-adjacent cells in the constructed graph, i.e., \(G_{m,n}\) and \(G_{m+2,n}\). Finally we get a matrix graph like Fig. 2(b). Since in Fig. 2(a), interference only exists between neighboring physical cells, if \(a_n\) and \(b_n\) are chosen properly, edges connecting non-adjacent cells in a matrix graph can be avoided and the original physical graph topology will be maintained.

Remark 2. After partitioning the graph into squares, edges connecting non-adjacent squares might appear. But since the confliction distance is larger than typical cell size (\(\bar{a}\) and \(\bar{b}\)), the interference will be quite low. Thus, we assume that after partitioning the graph, edges only exist in non-adjacent cells \(G_{m,n}\). This square partitioning technique is particularly useful in tackling a high-interference random graph, since the resulted matrix graph has a good structure and as shown in the next section, higher inter-cell interference will results in lower computation complexity.

Definition 1. A matrix graph is a conflict graph \(G = (V, E)\) where \(V\) satisfies [4][5] and an edge \((v_i^{m_1,n_1}, v_j^{m_2,n_2}) \in E\) only if

\[
(|m_1 - m_2| \leq 1)\ and (|n_1 - n_2| \leq 1). \tag{6}
\]

The constraint [5] ensures that only neighboring cells in the matrix graph have conflictions.
We assume that this graph is periodically modified due to network topology change. The network topology variance is usually slow. Also, since we are mainly dealing with small cell networks and they are usually located in houses, the mobility of the network is not a concern either. Thus, even if the matrix graph construction is complicated, we do not need to care about it. However, our frequency scheduling algorithms (presented in the next section) can be carried out in a more frequent manner than the network maintenance. There are indeed many works which consider fast network topology changes and interference variances, like vehicular ad hoc network and cognitive radio, but this kind of fast changing environment is too complicated and may introduce large overhead in small cell networks.

By abuse of notation in (2), we use \( f_{m,n}^i \) to represent the reuse ratio for \( v_{m,n}^i \) in a matrix graph, meaning that

\[
f_{m,n}^i = f_{v_{m,n}^i} = \frac{1}{C} \sum_{c=1}^{C} \mathcal{G}(v_{m,n}^i, c) \mu(v_{m,n}^i, c),
\]

where \( \mu = (\mu(v_{m,n}^i, c)) \) denotes the color weight discussed in remark 1. Then the ultimate goal of maximizing the weighted reuse ratio in a matrix graph can be written as

\[
\max_{\mathcal{G}} \hat{f} = \frac{1}{C} \sum_{m,n=1}^{M,N} \sum_{i=1}^{l_m,n} w_{m,n}^i f_{m,n}^i,
\]

s.t. \( \mathcal{G}(v_1, c) + \mathcal{G}(v_2, c) \leq 1 \), if \( (v_1, v_2) \in E \).

where the vertex weight \( w = (w_{m,n}^i) \) indicates the vertex weight of the communication link \( v_{m,n}^i \). This is called the matrix graph coloring problem (MGC).

**Theorem 1.** MGC problem is NP-complete.

**Proof:** See Appendix A. Note that in the conference paper we only deal with special problems when vertex weights \( w_{m,n} \) and color weights \( \mu(v_{m,n}, c) \) equal to 1.

This result suggests that no polynomial-time algorithms exist for MGC in a general matrix graph, unless \( P = NP \). However, the MGC problem still has an approximate solution with a bounded optimality gap, as shown in the next section. Thus, matrix graphs are approximated models specified for small-cell networks, which are easier to solve than general conflict graph models.

### III. Solving The Matrix Graph Coloring Problem

In this section, we first use a floor division to map the original MGC problem into many one-dimensional Maximum Weighted Independent Set (MWIS) problems. Then we solve each MWIS problem and combine the results with approximation techniques. The final algorithm to solve the MGC problem is outlined in Algorithm 1. In subsection A we give an overview of Algorithm 1. In subsection B, we analyze the performance and complexity of Algorithm 1.

#### A. Approximation Algorithm with a Floor Dividing Method

1) Finding Independent Sets in one-dimensional graph: First we define independent sets. Independent sets (IS) are especially useful in multi-coloring because they can be viewed as the basis for the valid coloring subspace [19]. In a matrix graph, an IS generally represents a subset of communication links who do not conflict with each other when utilizing the same resource. Thus, when we allocate frequency, we can allocate each frequency band (each color) to a specific IS with the maximum total weight, which will decompose the original problem of allocating many frequency bands. Specifically, for a graph \( G = (V,E) \), a vertex subset \( S \subseteq V \) is called independent if no two vertices in \( S \) share the same edge in \( E \). ISs are easy to be found in a matrix graph because an IS \( S \) in a matrix graph \( G \) can be decomposed into \( MN \) small ISs

\[
S = \bigcup_{m=1}^{M} \bigcup_{n=1}^{N} S_{m,n},
\]

and each \( S_{m,n} \subset V_{m,n} \) is an independent set in the cell \( G_{m,n} \). For each vertex \( v_{m,n}^i \), if we use \( q_{m,n}^i \in \{0,1\} \) to denote whether \( v_{m,n}^i \in S_{m,n} \), we can define the normalized weighted cardinality (NWC) \(|·|_N\) of \( S \) as

\[
|S|_N = \frac{\sum_{v \in V} q_v u_v}{\sum_{v \in V} u_v} = \frac{\sum_{m,n=1}^{M,N} \sum_{i=1}^{l_m,n} q_{m,n}^i u_{m,n}^i}{\sum_{m,n=1}^{M,N} \sum_{i=1}^{l_m,n} u_{m,n}^i}, \tag{10}
\]

where \( u_{m,n}^i \) is the vertex weight of vertex \( v_{m,n}^i \). If \( u_{m,n}^i = 1 \) for all vertices, the NWC simply equals to ratio of \( |S|/|V| \) where \(|·|\) means cardinality. We know clearly that \( |S|_N \) takes value in \([0,1]\). The indicator vector \( q = (q_{m,n}^i) \) in (10) can represent the solution \( S \). In the following we call this \( q \) the indicator representation of an independent set.

**Definition 2.** We call \( S^* \subset V \) the maximum weighted independent set (MWIS) of graph \( G = (V,E) \) if it is an independent set with the maximum normalized weighted cardinality (10).

Note that MIS (maximum independent set) problems are special cases of MWIS (maximum weighted independent set) problems when \( u_{m,n}^i = 1 \). So lemmas and theorems in this section can also be extended to corresponding lemmas and theorems for MIS.

**Lemma 1.** Finding MWIS in a one-dimensional matrix graph can be completely solved with \( O(KN) \) time complexity by a viterbi-like dynamic programming [12]. Here \( K \) is the supremum of the number of Independent Sets in each cell \( G_{m,n} \).

**Proof:** See Appendix B. In the conference paper we only deal with special problems when vertex weights \( w_{m,n} \) and
color weights $\mu(v_{m,n}^i,c)$ equal to 1. So the general proofs here can be easily extended.

A one-dimensional matrix graph is a matrix graph with height $M = 1$. Lemma 1 ensures the linear complexity of finding the MWIS on a one-dimensional matrix graph. We know that MWIS problem can be viewed as coloring with one color, so it is still NP-complete. However, for one-dimensional matrix graph, we have Lemma 1. Therefore, the original two-dimensional matrix graph can be partitioned into many one-dimensional subgraphs and MWISs can be found on each of them efficiently. Subsection III B shows that this scheme can achieve a bounded optimality gap.

2) Floor Dividing and the matrix graph decomposition:
In order to decompose the MWIS problem, we need to divide the whole M-by-N matrix graph into many one-dimensional subgraphs. Thus, a method called Floor Dividing (as shown in Fig. 3) is proposed. This scheme concurrently separates several copies of the M-by-N graph into several slender subgraphs (as shown in Fig. 3) and views each subgraph as a one-dimensional matrix graph. Assume that matrix graph $G = (V,E)$ has $M$ rows and $N$ columns with $M \leq N$. Even if $M > N$, we can solve the MWIS problem in the transpose of $G$. First we choose a positive integer $L < M$ as a parameter, called the floor height. We divide $M$ by $L$ and get
\[ M = L(Q - 1) + r, 0 < r \leq L. \tag{11} \]
It is notable that this division rounds up to get the quotient $Q$. Then we divide the row set $F = \{1,2,\ldots,M\}$ of $G$ into $Q$ subsets $F = \bigcup_{j=1}^{Q} F_j$, which represents one way of dividing the matrix graph into $Q$ slender layers. We call each subset $F_j$ a floor and call this set division the $t$th floor division. For example, for $t = 0$,
\[ F_0^Q = \{L(j - 1) + 1, L(j - 1) + 2, \ldots, L_j\}, j = 1, 2, \ldots, Q - 1, \]
\[ F_0^Q = \{L(Q - 1) + 1, L(Q - 1) + 2, \ldots, M\}. \tag{12} \]
Indeed this division is like dividing a mansion of height $M$ into $Q$ floors. In a floor division, the first $Q - 1$ floors have $L$ rows while the last one has $r$ rows. Fig. 3 shows 4 floor divisions. In each floor, we may choose one row to be a marginal row so that if all marginal rows in one floor division are eliminated, the remaining rows in different floors become non-adjacent. This property is crucial: if we take out these marginal rows in the M-by-N matrix graph $G$, the remaining graph is naturally divided into $Q$ non-interfering subgraphs, each with a size $M_j$-by-$N$ where $M_j < L$ (can be either $L - 1$ or $r - 1$). Thus, we can find the MWIS in all non-marginal rows by searching for the MWIS in each floor excluding the marginal row and then combine them together. A floor division scheme (as shown in Fig. 3) is a group of different floor divisions. The following lemma ensures the existence of a floor division scheme that makes each row being the marginal row exactly once.

**Lemma 2.** For a given $M$-by-$N$ matrix graph with $M < N$ and a floor height $L < M$, we can get a floor division scheme that has $L$ different floor divisions, with the $t$th division written as $\{ F_i^j \}_{j=1}^{Q}$. Each $F_i^j$ contains at most one marginal row, s.t.
(i) each division $t$ divides $G$ into $Q$ subgraphs which are only adjacent on marginal rows;
(ii) all marginal rows constitute $F = \{1,2,\ldots,M\}$.

**Proof:** See Appendix C.

The main idea is shown in Fig. 3. A cyclic construction scheme is utilized to make the required floor division scheme. Lemma 2 suggests that the entire graph $G$ can be divided into $Q$ subgraphs in $L$ different ways. Since each subgraph has a height $M_j$ bounded by $L$, we can view each one as a one-dimensional matrix graph and completely solve it with a bounded complexity according to results of Lemma 1. The aim of concurrently dividing $L$ copies is to ensure the property 2 of Lemma 2 and Theorem 2.

3) MGC Algorithm 1: Assume we have a matrix graph coloring problem with $G = (V,E)$ and a color pool $\Lambda = \{1,2,\ldots,C\}$. We would like to solve the MGC problem associated with vertex weights $w = (w_{m,n}^i)$ and color weights $\mu = (\mu(v_{m,n}^i,c))$. The way to solve it is to assign each color to a corresponding MWIS and all these MWISs as a whole yields the final solution. The only difference is that we change the weight using equation (13). The remaining problem is only the MWIS problem. We can then use the floor dividing to separate the whole graph into several subgraphs and solve the MWIS on each subgraph. Based on this idea, we give out the final Algorithm 1 to calculate a solution to the MGC problem. The performance is guaranteed by Theorem 2 in the next part.

**Remark.** In the algorithm, the approximation scheme lies in that instead of searching for MWIS in a whole subgraph, we find MWIS of each subgraph excluding the marginal row so that adjacent MWISs do not affect each other. Since marginal rows are between non-adjacent MWISs and contains no vertices in the MWISs, additional vertices can be added in to stuff the marginal rows. A larger floor height $L$ can result in higher complexity, but more accurate approximation.

**B. Reuse Ratio Lower Bound and Complexity Analysis**

In this subsection, we present the Theorem 2 which analyses the performance of Algorithm 1.
Algorithm 1 Solving MGC problem

**Input:** A matrix graph $G = (V, E)$, a color pool $\Lambda$, vertex weight $w$ and color weight $\mu$

**Output:** A matrix graph Coloring $\mathcal{C} = (\mathcal{C}(v_1, c))$ which optimizes $\hat{f}$ in (5) to $1 - 1/L$ of the optimal value.

Initialize

/*Floor Dividing*/
Calculate the floor dividing scheme $F_i^j, \forall t \in \{0, 1, \ldots, L - 1\}, \forall j \in \{1, 2, \ldots, Q\}$ based on Lemma 1:

/*MWIS for each color*/
FOR each color $c \in \Lambda$
   Solve a MWIS problem in $G$ associated with vertex weights $u = (u_{m,n}^i)$ defined as
   $$u_{m,n}^i = w_{m,n}^i \mu(v_{m,n}^i, c), \forall m, n, i$$  (13)
   FOR each floor division $t$ from 0 to $L - 1$
      /*MWIS for each one-dimensional graph*/
      FOR each floor row $j \in \{1, 2, \ldots, Q\}$
         Set $F_i^j = F_i^j$ the marginal row;
         View all rows that have index $m \in F_i^j$ as a one-dimensional matrix graph $\bar{G}_i^j$;
         Use Algorithm 2 to find a MWIS $\bar{S}_i^j$ in one-dimensional Graph $\bar{G}_i^j$ with no extra constraints;
      END
      /*MWIS for each marginal row*/
      FOR each floor row $j \in \{1, 2, \ldots, Q\}$
         View the marginal row in $F_i^j$ as a one-dimensional Matrix graph $\bar{G}_i^j$ and use Algorithm 2 to find a
         MWIS $\bar{S}_i^j$ in it with extra constraints induced by $\bar{S}_i^j$ and $\bar{S}_i^{j-1}$;
         Set $S_i^j = \bar{S}_i^j \cup \bar{S}_i^{j-1}$;
      END
   END
/*Combine all one-dimensional MWIS*/
Form a set $S_t = \bigcup_{j=1}^Q S_i^j$.
END

Choose $S_c \in \{S_0, S_1, \ldots, S_{M-1}\}$ that has the maximum normalized weighted cardinality.

/*Assign $c$ to $S_c$*/
Use the indicator form $q = (q_{m,n}^i)$ to represent $S_c$ and set
$$\mathcal{C}(v_{m,n}^i, c) = q_{m,n}^i, \forall m, n, i$$  (14)

END

Output the solution $\mathcal{C}$.

Theorem 2. Let $\mathcal{C}^*$ be the exact solution for the matrix graph Coloring (MGC) problem in the matrix graph $G$ and let $\hat{f}^*$ be the corresponding maximum weighted reuse ratio. Then Algorithm 1 obtains an approximate solution $S$ with complexity $O(CK^{L-1}MN)$. Furthermore, the corresponding weighted reuse ratio $\hat{f}$ satisfies
$$\hat{f} > \hat{f}^* \cdot \frac{L - 1}{L},$$  (15)
where $C$ is the number of colors. $L$ is the floor height designed beforehand. $K = \max_{m,n} K_{m,n}$ and $K_{m,n}$ denotes the number of independent sets in $G_{m,n}$.

Proof: Note that in the conference paper we only deal with special problems when vertex weights $w_{m,n}$ and color weights $\mu(v_{m,n}, c)$ equal to 1. So the proof here can be easily extended. The proof will be divided into three parts. We first show that proving (15) can be decomposed into proving the corresponding inequality for each color $c$. Then we prove that the floor division scheme can ensure the inequality for each color $c$. Finally we analyze the computation complexity.

To decompose (15), we plug (7) into (8) and get
$$\hat{f} = \frac{1}{C} \sum_{v \in V} w_v \mu(v, c) \mathcal{C}(v, c) = \frac{\sum_{v \in V} \mathcal{C}(v, c) \mu(v, c)}{\sum_{v \in V} w_v}.$$ Then we change the summation order of the numerator and arrive at
$$\hat{f} = \frac{1}{C} \sum_{c=1}^C \frac{\sum_{v \in V} w_v \mathcal{C}(v, c) \mu(v, c)}{\sum_{v \in V} w_v}.$$ (16)

where
$$B_c = \frac{\sum_{v \in V} \mathcal{C}(v, c) w_v \mu(v, c)}{\sum_{v \in V} w_v \mu(v, c)}.$$ (17)

For each fixed $c \in \Lambda$, $B_c$ is only determined by $\mathcal{C}(v, c), v \in V$, i.e., how this specific color $c$ is assigned to the vertices in $G$. Therefore, optimizing $B_c$ has nothing to do with other color assignments. If we use a set $S_c \subset V$ to denote the vertex set such that $\mathcal{C}(v, c) = 1$ and we define weights as (13), then it is easily seen that $B_c$ is the normalized weighted cardinality of $S_c$. Thus, optimizing $\hat{f}$ in (16) can be decomposed into subproblems and each of them regards maximizing a specific $B_c$ by finding a specific MWIS $S_c$. Then we assign each $c$ to $S_c$ like (14). As long as we get the approximate MWIS $S_c$ with a performance guarantee $1 - 1/L$, we can conclude that (15) holds.
We next claim that the floor division scheme indeed yields $B_c = |S_c|_N > (1 - 1/L)|S_c^*|_N$. Define $q = (q_{m,n}^i)$ as the indicator from of $S_c^*$, the MWIS of $G$ with the vertex weights defined as $\{13\}$. In the following we compare the normalized cardinality of $S_c^*$ to the floor-division-based approximate solution $S_c^*$ by induction.

As shown in algorithm 2, we have got the floor division scheme $\{F_j^i\}$ beforehand, where $t$ is from 0 to $L - 1$ and $j$ is from 1 to $Q$. Deleting the marginal row $m(t,j)$ in each floor $F_j^i$, we get a one-dimensional matrix graph $\bar{G}_j^i$ and we can use Algorithm 2 in the Appendix B to obtain an exact MWIS solution $\bar{S}_j^i$. We denote this solution in an indicator form $\bar{\theta} = (\bar{\theta}_{m,n}^i)$. By definition of the MWIS, $\bar{S}_j^i$ must have a larger normalized weighted cardinality than any other independent sets. Recall that $q$ is the indicator form of $S_c^*$, we have, for each $\{F_j^i\}$, that

$$|\bar{S}_j^i|_N \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i \theta_{m,n}^i = \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i \bar{\theta}_{m,n}^i \geq \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j.$$

Summing up the above inequality for all floor $j \in \{1, ..., Q\}$ with a specific $t$, we obtain

$$Q \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i \theta_{m,n}^i \geq \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j.$$

Defining $\bar{S}_t^i = \bigcup_{j = 1}^{Q} \bar{S}_j^i$, we have

$$|\bar{S}_t^i|_N = \frac{Q \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j}{Q \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i \theta_{m,n}^i} \geq \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j \geq \frac{1}{L} \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j,$$

where

$$\Sigma = \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i.$$

After adding new nodes in $\bar{S}_t^i$, we get a $S_t$ with larger normalized cardinality, thus we have

$$|S_t|_N \cdot \Sigma > |\bar{S}_t|_N \cdot \Sigma > \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j.$$

Lemma 1 ensures that each row $j$ appears in exactly $L - 1$ different floors divisions (except being the marginal row only once), so if we sum the above equation for all $t$, we arrive at

$$\frac{1}{L} \sum_{t = 0}^{L - 1} |S_t|_N \cdot \Sigma \geq \sum_{t = 0}^{L - 1} \sum_{j = 1}^{Q} \sum_{m \in F_j^i} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^j = (L - 1) \sum_{m = 1}^{M} \sum_{n = 1}^{N} \sum_{i = 1}^{l_{m,n}} u_{m,n}^i q_{m,n}^i = (L - 1) |S_c^*|_N.$$

Dividing both sides with $\Sigma L$ yields

$$\frac{1}{L} \sum_{t = 0}^{L - 1} |S_t|_N > \frac{L - 1}{L} |S_c^*|_N.$$

If we choose $t = t^*$ s.t. $S_{t^*}$ has the largest normalized cardinality, we will have

$$|S_{t^*}|_N > \frac{L - 1}{L} |S_c^*|_N.$$

$S_{t^*}$ is exactly our approximate solution for $S_c$. Thus, we know that $B_c$ is guaranteed to obtain the $1 - 1/L$ of the optimal value. And based on (16), we know that (15) holds.

The complexity scales like the following: For each $c$, we need to find the MWIS $S_c$, which is further decomposed into totally $QL$ subproblems. Each problem is solving the MWIS problem in a one-dimensional matrix graph. Based on Lemma 1, we can show that each problem will be completely solved with complexity $O(K^{L - 1}N)$. Therefore, the final problem will be solved in $O(QKL^{L - 1}N) = O(CK^{L - 1}MN)$.

The complexity $O(K^{L - 1}N)$ is obtained like the following. In fact, each cell contains at most $K$ Independent Sets. Based on the IS decomposition [10], we know that if we view each $M_j$-by-$N$ subgraph as a one-dimensional matrix graph, then one big cell is constituted of $M_j$ cells vertically, and each big cell contains at most $K^{M_j}$ Independent Sets. We know from Lemma 1 that $M_j < L - 1$, thus, each sub-problem can be solved with complexity $O(K^{L - 1}N)$.

**Remark 4.** Theorem 2 characterizes the tradeoff between computation complexity and efficiency that we can get, which forms a theoretical foundation to get the performance-guaranteed coloring scheme in a matrix graph. We have made the statement that matrix graphs are especially computing-efficient for small cell graphs. Now it is supported here. Since $K$ is a very small number, $O(K^{L})$ will not be especially large if the floor height $L$ is not that large. Moreover, if inter-cell interferences are high, the complexity $O(K^{L})$ further shrinks due to the branch trimming in finding one-dimensional MWISs (The Dynamic Programming in Lemma 1). In practice, if we choose $L = 5$, then based on Theorem 2, we can get a performance guaranteed to be better than $1 - 1/5 = 80\%$ of the optimal one. Moreover, simulation results suggest that this lower bound is quite loose. Usually the performance reaches more than 95%. A tighter bound is our goal in the future.
Remark 5. One might be concerned with the computational complexity which grows exponentially with the parameter $L$ to achieve the frequency allocation bound. However, this $(O(L^{-1}), O(K^L))$ performance-complexity tradeoff is inevitable due to the NP-Completeness. In fact, if we get a $(O(L^{-1}), O(L^d))$ tradeoff in the MGC problem and $L$ could go to infinity, we can simply set $L$ to be the same as the number of vertices in the graph, set $C = 1$ and set all weights to be 1, which finally yields an approximate Maximum Independent Set solution that hits $1/L$ to the bound with polynomial complexity of the network size. However, since $L$ is the number of vertices, the smallest granularity of a Maximum Independent Set problem (specific MWIS problem when all weights are 1) is now $1/L$. Thus, the approximate solution is exactly the same as the optimal one. However, this contradicts with the general belief that in NP-complete problems, we cannot find any polynomial-time solution that achieves the bound. Nonetheless, one can still explore new ways to lower the base $K$ of $O(K^L)$ in order to get the best exponential.

IV. SIMULATIONS

In this section, simulation results are obtained for large-scale small-cell networks. The test bed is set to be a $M$-by-$N$ matrix graph with totally $MN$ small cells. We set $M = 60$ and $N$ might change. In the first simulation we change $N$ from 1 to 200 to view the convergence result. After that we set $N = 200$ to view the performance variation with other parameters. No matter $N$ changes or not, $M$ and $N$ are set before generating the matrix graph, generating $MN$ small cells. However, vertices and edges in each cell are generated randomly. Vertices are generated with a uniform distribution with average $V_d$, called the vertex density, and each vertex is connected with any other vertex that satisfies constraint (6) with a constant probability $E_d \in [0, 1]$, called the edge density. This means that there will be approximately $V_d$ communication links in each cell and each communication link interferes with neighboring links with a probability of $E_d$. Assume we have $C = 6$ colors, which is the same setting in [16]. The color number does not affect the conclusion. In order to compare with other algorithms [15][16], we simply set color weight $\mu(v, e) \in \{0, 1\}$, which equals to 1 with probability $p_f$. Thus, the equivalent vertex density is actually $V_d \cdot p_f$, because we never assign a color to communication links with 0 weights. In the following when we refer to vertex density, we actually refer to $V_d \cdot p_f$.

The performance criterion is the weighted reuse ratio defined in [8]. We assume all vertex weights are 1, which does not affect the simulation results. So, this criterion now just equals to the average ratio of frequency bands that is used by each communication link which directly shows the resource reuse efficiency.

In Fig. 4 and 5, the horizontal axis is the length $N$ of the matrix graph. We set $M = 60$ and changes $N$ from 1 to 200, while taking down the weighted reuse ratio obtained by Algorithm 1. In these two figures, the vertex density is set to be 1.6 and the edge density is 0.6 respectively. We find that when $N$ goes large, each curve converges to a constant value. For different curves (with different floor height $L$), all curves uniformly converge (simultaneously for each $N$) to a limit. This limit is the theoretical limit of frequency allocation.

In order to support Theorem 2 which says that the solution obtained by Algorithm 1 has at most a $1/L$ gap to the optimal solution, we illustrates the performance when the floor height $L$ goes large, under different vertex and edge densities (as shown in Fig. 3). It is clear that when $L$ increases, each curve converges to a limit. Thus, we can approximately tell the theoretical limit of frequency allocation, despite the fact that telling the exact value has been proved to be NP-complete.

A more interesting result is that, when vertex density and edge density increases, this limit shrinks. This is intuitively
right because as interference relationships become complicated, the available resources to be reused decreases. We conjecture that this limit, on a randomly generated large scale network, only depends on vertex density and edge density. A meaningful future work is to investigate this conjecture, which can ultimately tell the frequency reuse limit.

In Fig. 7 and 8 we show the performance comparison of the Algorithm 1 with three other algorithms. GB-DFR is a graph based heuristic proposed in [16], which generalized the conception of saturation-degree graph coloring in [15] and got good performance in cellular system simulations. GLC is the Greedy List-Coloring proposed in [15]. It is simple and efficient. We find that our algorithm performs gradually better when edge density and vertex density increases. This is common since graph-based algorithms usually have good performance in degree-bounded graphs. But when interference become complicated, there is no guarantee that they perform well. By the way, after one color is assigned to a vertex, both GB-DFR and GLC have sorting in the whole network, which drives the complexity to $O(MNfC \cdot MN \log MN)$, where $f$ is the weighted reuse ratio and $M$-by-$N$ is the network size. When network goes large, this becomes impractical. SFR is called Soft Frequency Reuse [16], which uses different reuse factors in cell edge and cell center. In our matrix graph, we just consider the cell center to be vertices that do not interfere with the neighboring cells. Since SFR is essentially a grid-model algorithm, it does not perform quite well in our tests. However, when interference is quite large (edge density reaches 0.8), it has excellent performance. We suspect that this is because when edge density reaches some threshold, interference management schemes does not have much gain compared to interference avoidance schemes.

V. Conclusions

In this paper we are focusing on the ultimate limit of frequency allocation in a 5G network. To study this problem, we proposed a matrix graph model and constructed an analytical framework combining matrix graph coloring (MGC) and maximum weighted independent set (MWIS), based on properties of large-scale small-cell networks. Utilizing this model, we obtain an approximation algorithm that achieves a bounded gap to the optimal performance with a complexity growing linearly with the network size, despite the NP-completeness of the MGC problem. Therefore, if we could build a proper matrix graph, we can find the nearly-optimal way to allocate resources like frequencies and time slots. This is in contrast with conventional graph-coloring based heuristics which usually have no guarantee on performance. Moreover, the proposed scheduling algorithm has lower computational complexity if cells are smaller and inter-cell interference are more complicated. Thus, we conclude that frequency allocation in high-interference small-cell networks can be carried out efficiently and the small-cell networks are indeed practical for the future 5G network.
To reduce the tiling of a square to a MWIS problem in a neighboring tiles have the same color on the common edge. Fig. 9. A square tiling of a 3-by-4 finite square with Wang tiles. Any two in Theorem 2 could be further tightened due to the law of approximation algorithms. At least, the bound of a concrete coloring scheme, even without carrying out the expectation regardless of the NP-completeness of finding according to our observations, a random-graph analytical theories, we are still interested in further improving them.

We will reduce another NPC problem, the Wang tiling, to the MWIS problem on a matrix graph. Since the MWIS problem can be viewed as the MGC problem with one color, the MWIS on a matrix graph can be reduced to the MGC problem. Meanwhile, the MGC problem can be reduced to the general MWIS problem trivially by assigning each color to a MWIS. Thus, if we have proved the first statement, then MGC is NP-complete. Note that MIS (maximum independent set) and MWIS (maximum weighted independent set) problems are both NP-complete so we can still prove MGC to be NP-complete even if all weights $\mu(v,c)$ and $w_v$ equals to 1. The Wang tiling problem is a classic unsolvable combinatorial problem. A Wang tile is a square with its four edges, namely north-, east-, west- and south-edges colored by a set of colors. Now assume that we have a set of Wang tiles $W = \{w_1, w_2, ..., w_{l-1}, w_l\}$. A tiling $T$ is said to be valid, if neighboring tiles has the same color. The following Figure shows an example of Wang tiling of a 3-by-4 square. In [23], the author stated that whether a given set of Wang tiles can validly tile a $M \times N$ square is NP-complete with the size of square. The author has not given the proof in [23], but a following paper proved a special case of original problem to be NP-complete. So the NP-completeness of the original tiling problem in [24] is also ensured.

Assume we have a square lattice denoted by $\{(m, n)\}_{m=1,n=1}^{M,N}$ to be tiled by the given tile set $W$. To reduce the tiling of a square to a MWIS problem in a matrix graph $G = (V,E)$, we first construct the corresponding graph. Writing

$$V = \bigcup_{m,n=1}^{M,N} V_{m,n}$$

for the vertex set of $G$, where each $V_{m,n} = \{v_{m,n}^i\}_{i=1}^l$ is the vertex set of a L-complete graph $G_{m,n} = K_l$. Each vertex $v_{m,n}^i$ is associated with a tile $w_i$. For two horizontally neighboring vertex sets, for example, $V_{m,n}$ and $V_{m+1,n}$, $(v_{m,n}^i, v_{m+1,n}^j) \in E$ if and only if tile $w_i$'s east edge does not match tile $w_j$'s west edge when they are respectively put at lattice point $(m,n)$ and $(m+1,n)$. Similarly, for two vertically neighboring vertex sets $V_{m,n}$ and $V_{m,n+1}$, $(v_{m,n}^i, v_{m,n+1}^j) \in E$ if and only if tile $w_i$'s south edge does not match tile $w_j$'s north edge when they are respectively put at lattice points $(m,n)$ and $(m,n+1)$. One can easily check that $G = (V,E)$ constructed above is a matrix graph consistent with definition 1.

Next we show that tiling the $M$-by-$N$ square can be reduced to finding a Maximum Weighted Independent Set in $G$ with vertex weights $w_{m,n} = 1, \forall m,n,i$. Since each cell of the matrix graph is a complete graph $K_l$, we can only pick up one vertex from each cell. If the maximum weighted independent set that we find in $G$ coincidentally picks up one vertex, with the index $(m,n)$, in each cell $G_{m,n}$, then we can construct a tilling $T(m,n) = i(m,n)$ of the square. Since the conflicions between two edges in the matrix graph indicates the mismatching between corresponding tiles, we know that this tilling $T(m,n) = i(m,n)$ has no mismatching and is valid. As a result, the square tilling problem can be reduced to tell if the maximum weighted independent set in this matrix graph $G$ has a normalized weighted cardinality $1/l(1$ vertex from $l$ vertices in each cell). Since the tilling problem is NP-complete, the general MWIS problem in a matrix graph has the same difficulty.

In this section we show that MWIS problem in a one-dimensional matrix graph can be solved completely in linear time. Before giving out the dynamic programming algorithm, we need to review some properties of a one-dimensional matrix graph. We call a one-dimensional matrix graph is a Vector Graph. Solving MWIS in a Vector Graph can give us convenience on solving MWIS in general matrix graphs. Moreover, apart from this convenience, we have mentioned that one-dimensional cellular network itself is of particular practical interests. Similar to Definition 1, we have

**Definition 3.** A Graph $G = (V,E)$ is a Vector Graph if

$$V = \bigcup_{n=1}^{N} V_n$$

$$V_n = \{v_n^i\}_{i=1}^l$$

Fig. 9. A square tiling of a 3-by-4 finite square with Wang tiles. Any two neighboring tiles have the same color on the common edge.
An edge \((v'_1, v'_2) \in E\) only if
\[ |n_1 - n_2| \leq 1 \] (22)

**Algorithm 2** Finding MWIS in a one-dimensional matrix graph with constraints \(Y\)

**Input**: A Vector Graph \(G = (V, E)\), vertex weights \(u = (u'_n)\), constraints \(Y = \{Y_n\}_{n=1}^{n}, Y_n \subset X_n, \forall n\)

**Output**: A MWIS \(S^*\) which optimizes (24).

Initialize
For all \(k_1\) s.t. \(a_{k_1} \in X_1\)
if \(a_{k_1} \notin Y_1\), set \(\psi_{(1)}^{k_1} = \emptyset\);
else set \(\psi_{(1)}^{k_1} = (a_{k_1}^{k_1})\);
end

For \(n\) from 2 to \(N\)
For all \(k_n\) s.t. \(a_{k_n} \in X_n\)
if \(a_{k_n} \notin Y_n\) set \(\psi_{(n)}^{k_n} = \emptyset\) (Extra Constraints)
else find \(l^* \in \{1, ..., K_{n-1}\}\) s.t.
1). \((a_{k_{n-1}}^{l_{n-1}}, a_{k_n}^{k_n}) \in R_{n-1,n}\) (1D Constraints)
2). \(\psi_{(n-1)}^{l_{n-1}} \neq \emptyset\)
3). \(l^*\) maximizes \(|\psi_{(n-1)}^{l_{n-1}}|_N\) (Bellman Equation)
set \(\psi_{(n)}^{k_n} = (\psi_{(n-1)}^{l_{n-1}})S_n = (\psi_{(n-1)}^{l_{n-1}}a_{k_n}^{k_n})\).
end
end

Find \(k^* \in \{1, ..., K_N\}\) that maximizes \(|\psi_{(N)}^{k^*}|_N\). \(\psi_{(N)}^{k^*}\) is the maximum weighted independent set that we are seeking for.

**Output** \(S^* = \psi_{(N)}^{k^*}\).

We use the notation \(G_n\) to denote the cell that contains \(V_n\). As a counterpart to (9), we decompose an independent set \(S\) in \(G\) by
\[ S = \bigcup_{n=1}^{N} S_n \] (23)
and the Maximum Weighted Independent Set problem is aimed at maximizing
\[ |S|_N = \frac{\sum_{n=1}^{N} \sum_{i=1}^{k_n} u'_i}{\sum_{n=1}^{N} \sum_{i=1}^{k_n} u_i} \] (24)
where \(u = (u'_n)\) are the vertex weights.

Then we define the sequence representation of an independent set. Noticing that if \(S\) is an independent set of \(G\), then for \(\forall n, S_n\) is an independent set of the corresponding cell \(G_n\). We denote all possible independent sets of \(G_n\) by \(X_n = \{a_1, a_2, ..., a_{K_n}\}\). Suppose that \(S_n = a_{k_n}^{k_n}\) for \(\forall n, S\) can be written in a \(N\)-sequence representation
\[ S = (S_1S_2...S_n) = (a_{k_1}^{k_1}a_{k_2}^{k_2}...a_{k_n}^{k_n}), k_n \in \{1,2,...,K_n\}, \forall n \] (25)

For simplicity of notation, we use the same letter \(S\) for this sequence. When mentioning the normalized weighted cardinality (NWC) of a sequence \(S\), we refer to the NWC of the corresponding independent set.

For each two adjacent cells \(G_n\) and \(G_{n+1}\), we define \(G_{n,n+1}\) as the induced graph containing \(G_n\) and \(G_{n+1}\). i.e. the graph that contains \(G_n, G_{n+1}\) and the confliction edges between them. Then we define a relation
\[ R_{n,n+1} = \{(\alpha, \beta) | \alpha \in X_n, \beta \in X_{n+1}, 2 - sequence (S_nS_{n+1}) = (\alpha\beta)\ is\ an\ independent\ set\ of\ G_{n,n+1}\} \] (26)
where \(X_n\) still denotes all possible independent sets of \(G_n\). The relationship \(R_{n,n+1}\) contains all possible combinations of \((S_n, S_{n+1})\) that satisfy confliction constraints imposed by edges connecting \(G_n\) and \(G_{n+1}\). That is to say, any two adjacent elements in a sequence representation must belongs to \(R_{n,n+1}\). However, belonging to \(R_{n,n+1}\) is not the sufficient condition for a pair \((S_n, S_{n+1})\) to be legal. In fact, apart from conflictions between \(G_n\) and \(G_{n+1}\), there will be constraints on \((S_n, S_{n+1})\). This is particularly important in generalizing one-dimensional solution to a two-dimensional network, because conflictions may be introduced from the other dimension. So we need to formulate extra constraints, which are written as
\[ S_n = a_{k_n}^{k_n} \in Y_n, Y_n \subset X_n \] (27)
This means that for each \(a_{k_n}^{k_n}, k_n\) can only take values in some certain subset of \(\{1,2,...,K_n\}\) due to extra constraints.

Based on the above definitions, we give out a dynamic programming Algorithm 2 to solve the MWIS problem in a Vector Graph. In this algorithm we use the sequence \(\psi_{(n)}^{k_n} = (S_1S_2...S_n)\) to represent the searching branches of the sequence representation of the best independent set up to step \(n\). In fact, \(\psi_{(n)}^{k_n}\) is an \(n\)-sequence, i.e. an independent set of the first \(n\) cells including \(G_1\) to \(G_n\), with the assumption that \(S_n\) equals to a specific \(a_{k_n}^{k_n}\). In another word, the \(n\)-sequence \(\psi_{(n)}^{k_n}\) should be written as \((\ast \ast a_{k_n}^{k_n})\). \(k_n\) obviously denotes the current state in the \(n\)th step. For each \(k_n\), we only reserve one optimal path \(\psi_{(n)}^{k_n}\), which is similar to the classic Viterbi Decoding[12]. By definition, \(|\psi_{(n)}^{k_n}|_N\) still denotes the NWC of \(\psi_{(n)}^{k_n}\), which is going to be optimized. Since Algorithm 2 is a direct application of dynamic programming and the proof is quite straightforward, we omit the proof in this paper.

**APPENDIX C**

**PROOF OF LEMMA 2: A FLOOR DIVISION SCHEME**

Now we prove Lemma 2, which indicates that for any \(M\) and \(L < M\), there is a floor division scheme that guarantees the properties i) to iii). We prove this lemma by explicitly constructing \(L\) floor divisions \(F = \bigsqcup_{j=1}^{L} F'_j\), \(t\) from 1 to \(L\). This construction is also useful in the Algorithm 1. Assume \(r = M - L(Q - 1)\). We know that \(0 < r \leq L\). For \(t = 0, 1, 2, ..., Q - 1\), we can define the scheme as
\[ F'_j = \left\{ k \in \mathbb{Z} \mid k \geq j(L + r) \right\} \] (28)
and finally, we have \(F = \bigsqcup_{j=1}^{L} F'_j\), \(t\) from 1 to \(L\).
we use \( \{12\} \) to build each \( F_i^j, \forall j \). We set the marginal rows as \( m(0,j) = L(j-1)+1 \). For \( 1 < t \leq r-1 \), we build

\[
F_i^t = t + F_i^0 = \{ m \in F | m - t \in F_i^0 \}, t = 1, 2, ..., r-1
\]

(28)

This is like \( t = 1 \) in Fig. 3, i.e. the second floor division where \( r = 2 \). For these floor divisions, we set \( m(t,j) = t + L(j-1) + 1 \). Note that here + and - are in the sense of modulo \( M \). If \( r \) equals to \( L \), which means that \( M \) is divisible by \( L \), we have finished building floors. Otherwise, for \( t \) from \( r \) to \( L - 1 \), we set

\[
F_i^t = t + F_i^0 = \{ m \in F | m - t \in F_i^0 \}, j = 1, ..., Q - 2, \forall t
\]

(29)

\[
F_i^{Q-1} = \{ L(Q-2) + 1 + t, ..., M \}, t = r, ..., L - 1
\]

(30)

\[
F_i^Q = \{ 1, ..., t \}, t = r, ..., L - 1
\]

(31)

For \( j \) from 1 to \( Q - 1 \), we still set the marginal rows as \( m(t,j) = t + L(j-1) + 1 \). For \( j = Q \), we do not set any rows to be marginal rows. These floor divisions are like the third and fourth divisions in Fig. 3. We clearly see from Fig. 3 that this floor division scheme results in the cyclic behavior of marginal rows, and thus, each element from \( \{1, ..., M\} \) shows up as the marginal row once. The property (i) in Lemma 2 can be easily checked.

REFERENCES

[1] V. H. MacDonald. “The cellular concept. Bell System Technical Journal vol. 58, no. 1, pp. 15-41, 1979

[2] S. W. Halpern. “Reuse partitioning in cellular systems. Proceedings of the 33rd IEEE Vehicular Technology Conference vol. 33, pp. 322-327. IEEE, 1983.

[3] A. Gamst. “Some lower bounds for a class of frequency assignment problems.” IEEE Transactions on Vehicular Technology vol. 35, no. 1, pp. 8-14, 1986

[4] I. Katzela and M. Naghshineh. “Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey.” IEEE Personal Communications, vol. 3, no. 2, pp. 10-31, 1996

[5] U. Gotzner and R. Rathgeber. “Spatial traffic distribution in cellular networks.” Proceedings of the IEEE 48th Vehicular Technology Conference, vol. 3, pp. 1994-1998, 1998.

[6] M. Rahman and H. Yani, “Enhancing cell-edge performance: a downlink dynamic interference avoidance scheme with inter-cell coordination.” IEEE Transactions on Wireless Communications, vol. 9 no. 4, pp. 1414-1425, 2010

[7] J. G. Andrews, H. Claussen, M. Dohler, S. Rangan and M. C. Reed. “Femtocells: Past, present, and future.” IEEE Journal on Selected Areas in Communications, vol. 30, no. 3, pp. 497-508, 2012

[8] J. Hoydis, M. Kobayashi and M. Debbah. “Green small-cell networks. IEEE Vehicular Technology Magazine, vol. 6, no. 1, pp. 37-43, 2011

[9] C. Y. Wong, R. S. Cheng, K. B. Lataief and R. D. Murch. “Multiuser OFDM with adaptive subcarrier, bit, and power allocation.” IEEE Journal on Selected Areas in Communications, vol. 17, no. 10, pp. 1747-1758, 1999

[10] M. Sawahashi, Y. Kishiyama, A. Morimoto, D. Nishikawa and M. Tanno. “Coordinated multipoint transmission/reception techniques for LTE-advanced (Coordinated and Distributed MIMO).” IEEE Transactions on Wireless Communications vol. 17, no. 3, pp. 26-34, 2010

[11] D. Gesbert, S. Hanly, H. Huang, S. S. Shitz, O. Simeone and W. Yu. “Multi-cell MIMO cooperative networks: A new look at interference.” IEEE Journal on Selected Areas in Communications, vol. 28, no. 9, pp. 1380-1408, 2010

[12] L. Bahl, J. Cocke, F. Jelinek and J. Raviv. “Optimal decoding of linear codes for minimizing symbol error rate (Corresp.).” IEEE Transactions on Information Theory, vol. 20, no. 2, pp. 284-287, 1974

[13] D. Breslau. “New methods to color the vertices of a graph.” Communications of the ACM, vol. 22, no. 4, pp. 251-256, 1979

[14] M. C. Necker. “A graph-based scheme for distributed interference coordination in cellular OFDMA networks.” Proceedings of the IEEE Vehicular Technology Conference, pp. 713-718, Spring 2008

[15] W. Wang and X. Liu. “List-coloring based channel allocation for open-spectrum wireless networks.” Proceedings of the IEEE Vehicular Technology Conference, vol. 62, no. 1, p. 690, 2005.

[16] S. Uygunelen, G. Auer and Z. Bharucha. “Graph-based dynamic frequency reuse in femtocell networks.” Proceedings of the 73rd IEEE Vehicular Technology Conference, pp. 1-6, 2011.

[17] G. Kulkarni, S. Adlakha and M. Srivastava. “Subcarrier allocation and bit loading algorithms for OFDMA-based wireless networks.” IEEE Transactions on Mobile Computing, vol. 4, no. 6, pp. 652-662, 2005

[18] K. I. Aardal, S. PM Van Hoesel, A. C. M. Koster, C. Mannino and Antonio Sassano. “Models and solution techniques for frequency assignment problems.” Annals of Operations Research vol. 153, no. 1, pp. 79-129, 2007.

[19] Z. Fang and B. Bensaou. “Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks.” Proceedings of the INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 1284-1295, 2004.

[20] I. M. Bomze, M. Budinich, P. M. Pardalos and M. Pelillo. “The maximum clique problem.” Handbook of combinatorial optimization, Springer US, pp. 1-74, 1999.

[21] P. Alimonti and V. Kann. “Some APX-completeness results for cubic graphs.” Theoretical Computer Science, vol. 237, no. 1, pp. 123-134, 2000

[22] B. Bai, W. Chen, Z. Cao and K. B. Letaief. “Max-matching diversity in OFDMA systems.” IEEE Transactions on Communications, vol. 58, no. 4, pp. 1161-1171, 2010

[23] M. R. Garey and D. S. Johnson. “Computers and intractability.” New York: Freeman, vol. 174, 1979.

[24] V. Lukkarila. “The square tiling problem is NP-complete for deterministic tile sets.” TUCS Technical Report, No,754. 2006.