

Research Article

On the Complexity of Scheduling in Wireless Networks

Changhee Joo,¹ Gaurav Sharma,² Ness B. Shroff,³ and Ravi R. Mazumdar⁴

¹Department of EECE, Korea University of Technology and Education, 1800 Chungjeollo, Cheonan, Chungnam 330-708, Republic of Korea

²D. E. Shaw & Co., L.P., 120 West Forty-Fifth Street, New York, NY 10036, USA

³Departments of ECE & CSE, The Ohio State University, 2015 Neil Avenue, Columbus, OH 43210, USA

⁴Department of ECE, University of Waterloo, 200 University Avenue, West Waterloo, ON, Canada N2L 3G1

Correspondence should be addressed to Changhee Joo, changheejoo@gmail.com

Received 11 January 2010; Accepted 1 September 2010

Academic Editor: Mohammad Shikh-Bahaei

Copyright © 2010 Changhee Joo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We consider the problem of throughput-optimal scheduling in wireless networks subject to interference constraints. We model the interference using a family of K -hop interference models, under which no two links within a K -hop distance can successfully transmit at the same time. For a given K , we can obtain a throughput-optimal scheduling policy by solving the well-known maximum weighted matching problem. We show that for $K > 1$, the resulting problems are NP-Hard that cannot be approximated within a factor that grows polynomially with the number of nodes. Interestingly, for geometric unit-disk graphs that can be used to describe a wide range of wireless networks, the problems admit polynomial time approximation schemes within a factor arbitrarily close to 1. In these network settings, we also show that a simple greedy algorithm can provide a 49-approximation, and the maximal matching scheduling policy, which can be easily implemented in a distributed fashion, achieves a guaranteed fraction of the capacity region for “all K .” The geometric constraints are crucial to obtain these throughput guarantees. These results are encouraging as they suggest that one can develop low-complexity distributed algorithms to achieve near-optimal throughput for a wide range of wireless networks.

1. Introduction

Scheduling link transmissions in a wireless network so as to optimize one or more of the performance objectives (e.g., throughput, delay, or energy) has been the topic of paramount interest over the past several decades. In their seminal work, Tassiulas and Ephremides [1] characterized the *capacity region* of constrained queuing systems, such as a wireless network. They developed a queue length-based scheduling scheme that is *throughput-optimal*, that is, it stabilizes the network if the user rates fall within the capacity region of the network. Unlike wireline networks, where all links have fixed capacities, the capacity of a wireless link can be influenced by channel variation due to fading, changes in power allocation or routing, changes in network topology, and so forth. Thus, the capacity region of a wireless network can vary due to changes in power allocation or routing. To efficiently utilize the wireless resources, one must therefore develop algorithms

that can perform jointly routing, link scheduling, and power control under possibly varying channel conditions and network topology. This has spurred recent interest in developing cross-layer optimization algorithms (see, e.g., [2–5]).

Motivated by the works on fair resource allocation in wireline networks [6, 7], researchers have also incorporated congestion control into the cross-layer optimization framework [8–10]. The congestion control component controls the rate at which users inject data into the network to ensure that the user rates fall within the capacity region.

Most of the above cross-layer optimization problems have been shown to exhibit a mathematical decomposition [2, 8]. To elaborate, the cross-layer optimization problem can be decomposed into multiple subproblems, where each subproblem corresponds to optimization across a single layer. The subproblems are loosely coupled through parameters that correspond to congestion prices or queue lengths at the individual links.

The main component of all these cross-layer optimization schemes is the optimal scheduler that solves a very difficult global optimization problem of the form:

$$\begin{aligned} & \text{maximize} && \sum_{l \in \mathcal{L}} p_l r_l \\ & \text{subject to} && \mathbf{r} \in \Lambda, \end{aligned} \quad (1)$$

where \mathcal{L} denotes the set of wireless links; \mathbf{r} is the vector of link rates r_l , $l \in \mathcal{L}$; p_l is the congestion price or possibly some function of queue length at link $l \in \mathcal{L}$; Λ is the capacity region of the network.

The main difficulty in solving the above optimization problem is that the capacity region Λ depends on the network topology and, in general, has no easy representation in terms of the power constraints at the individual links or nodes. The above optimization problem is, in general, NP-Complete and Nonapproximable.

In this paper, we consider a class of scheduling problems that we term Maximum Weighted K -Valid Matching Problems (*MWKVMPs*). These problems arise as simplifications to the scheduling problem specified by (1). The basic idea is to limit the interference to only K hops, where K is a positive integer. By varying K , one can capture the interference characteristics of a broad range of wireless networks.

The rest of the paper is organized as follows. The model, problem formulation, related works, motivation, and main contributions of this work are presented in the next section. Some hardness and approximability results for the class of scheduling problems that we consider are presented in Section 3. We then restrict our attention to geometric unit-disk graphs that naturally model the connectivity graph of wireless networks, and develop approximation schemes for our scheduling problems in Section 4. By focusing on the throughput performance in Section 5, we reduce the complexity of scheduling schemes further, and show that a distributed maximal matching algorithm achieves a provable throughput guarantee. The geometric constraints of graphs remain crucial to obtain the throughput guarantees. Finally, we provide concluding remarks in Section 6.

2. System Model and Problem Formulation

We consider a set V of wireless nodes, each communicating over a single wireless interface. We assume that all transmissions are carried out over the same wireless channel, and therefore interfere with each other. We assume that all transmissions from a node are carried out at the same power level (which can be different for different nodes). We connect two nodes with an (undirected) edge if each of them can successfully receive from the other, provided no other node in the network transmits at the same time. The set of (undirected) edges so formed is denoted by E . Note that the existence of an edge between two nodes depends on the power allocated to the nodes, noise variances, as well as coding and modulation schemes. Our emphasis on bidirectional edges stems from the fact that most network and transport layer protocols assume bidirectional communications between the nodes. We also

note that our main results can easily be extended to settings where directed edges are allowed between the nodes.

We next introduce the class of scheduling problems we consider in this paper. We first introduce some notation. Let $G = (V, E)$ be an undirected graph (connectivity graph of a wireless network, in our case) having V as the set of vertices (nodes) and E as the set of edges (link). A *matching* is a set of edges no two of which share a common vertex. We now generalize this concept of matching to K -valid matchings for an integer $K \geq 1$.

Let $d_h(x, y)$ denote the minimum number of hops between vertices $x, y \in V$. Letting \mathbb{N} denote the set of nonnegative integers, we define a distance function $d : (E, E) \rightarrow \mathbb{N}$ as follows: for two edges $e_u = u_1 u_2$, $e_v = v_1 v_2 \in E$, let

$$d(e_u, e_v) \triangleq \min_{i,j \in \{1,2\}} d_h(u_i, v_j). \quad (2)$$

We call a set of edges M a " K -valid matching" if for all $e_1, e_2 \in M$ with $e_1 \neq e_2$, we have $d(e_1, e_2) \geq K$. Observe that the concept of matching discussed before is equivalent to the concept of 1-Valid matching in this new terminology. Let S_K denote the set of K -Valid matchings of the graph G . We consider the following scheduling problems:

$$\begin{aligned} (\text{MWKVMP}) \quad & \text{maximize} && \sum_{e \in M} w_e \\ & \text{subject to} && M \in S_K, \end{aligned} \quad (3)$$

where w_e denotes the weight of edge $e \in \mathcal{L}$. Note that the weight of each edge e is a positive, but otherwise arbitrary, number that can possibly depend on many factors (e.g., congestion price, supported rate, queue length). The above class of problems will henceforth be referred to as Maximum Weighted K -Valid Matching Problems (*MWKVMPs*). When all edge weights are set to unity, we obtain the following class of problems:

$$\begin{aligned} (\text{MKVMP}) \quad & \text{maximize} && |M| \\ & \text{subject to} && M \in S_K, \end{aligned} \quad (4)$$

where $|M|$ denotes the cardinality of the set M . In the sequel, we refer to these problems as Maximum K -Valid Matching Problems (*MKVMPs*).

We note that the scheduling problems specified by (3) are natural simplifications of the complex scheduling problem specified by (1). This is because for a given K , by satisfying the K -hop interference constraints one can guarantee a certain fixed data rate at a given edge. The weight of each edge can then be determined as some function of the rate it supports and the congestion price at the edge. The scheduling problem specified by (1) then corresponds to *MWKVMP* for that particular value of K . For simplicity of notation, we did not explicitly show the dependence of edge weights on K in (3).

From the above discussion, it is not surprising to see that *MWKVMPs* can represent the scheduling problem specified by (1) under a wide variety of interference models. Below we discuss two widely used interference models that can be obtained as special cases of the interference constraints in (3).

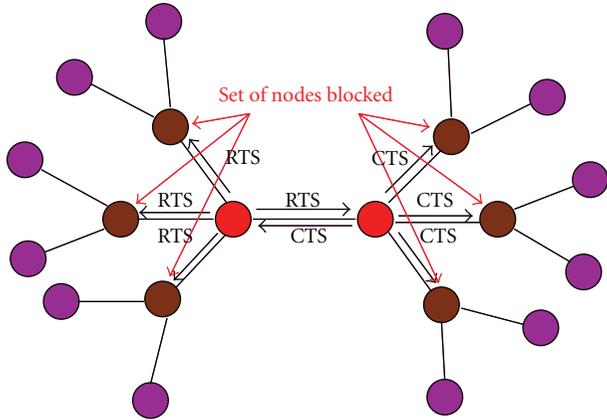


FIGURE 1: The 2-hop interference set of a given edge for RTS/CTS based communication model of IEEE 802.11 DCF.

Node-Exclusive (or Primary) Interference Model. This is a commonly used model for Bluetooth and FH-CDMA networks [11, 12]. Under this model, the set of edges that transmit simultaneously must constitute a matching. Then the scheduling problems specified by (3) and (4) correspond to the classical Maximum Weighted Matching Problem (*MWMP*) and the Maximum Matching Problem (*MMP*), respectively. Both these problems can be solved in polynomial time [13].

IEEE 802.11-Based Interference Model. This is a commonly used model for IEEE 802.11-based wireless networks [9, 14], under which the chosen set of edges must constitute a 2-Valid matching. It models the communication under the RTS/CTS-based scheme of IEEE 802.11 DCF (see Figure 1). Note that the sender and the receiver exchange RTS and CTS messages preventing their neighboring nodes from participating in a communication, which is equivalent to saying that the chosen set of communicating node pairs must constitute a 2-Valid matching.

In general, we use the term “ K -hop interference model,” under which a scheduler should provide a K -valid matching. The node-exclusive and IEEE 802.11-based interference models correspond to the K -hop interference model with $K = 1$ and $K = 2$, respectively.

2.1. Related Work. The 1-hop interference model has been studied in many different contexts due to its simplicity [1, 4, 8, 12, 15–17]. A polynomial time link scheduling algorithm has been developed in [12], and distributed schemes that guarantee a throughput within a constant factor of the optimal have been developed in [8, 15]. Recently, a class of throughput-optimal scheduling policies, called *pick-and-compare*, has been proposed [16, 17]. Although they achieve the throughput-optimality with a low complexity, they result in causing significantly long queue lengths, which in turn results in high delays, and for practical buffer sizes, can result in low throughput performance [18].

In [9], the performance of maximal scheduling schemes has been studied under the 2-hop interference model. It has

been shown that the maximal scheduling schemes achieve a throughput within a factor of $(N_\epsilon + 1)$ of the capacity region, where N_ϵ denotes the maximum link degree. In [15], the maximal scheduling schemes are shown to achieve at least a factor of $\Delta_K(G)$ of the optimal throughput, where $\Delta_K(G)$ is the interference degree of the connectivity graph (see Definition 11). It also has been shown in [19, 20] that random access scheduling policies can achieve comparable performance.

The *MKVMP* for $K = 2$ is often known as the induced matching problem, which has been shown to be NP-Hard [21]. The work of [14] is closest in spirit to our work. The authors consider the induced matching problem from the perspective of carrying out maximum number of simultaneous transmissions. They study the approximability of the induced matching problem for general as well as specific kinds of graphs, and develop a distributed constant factor Polynomial-Time Approximation Scheme (PTAS) for the induced matching problem under geometric unit-disk graphs.

However, most previous studies are limited to the 1-hop or 2-hop interference model. It has been observed through simulations in [22] that, under different network settings, the K -hop interference model with $K > 2$ can better capture the network interference constraints. For the detailed results, we refer to our technical report [22].

2.2. Main Contributions. From a theoretical perspective, we provide several results on the hardness and approximability of *MWKVMP* and *MKVMP* for $K > 1$. Although some of these results have previously been obtained for $K = 2$, to the best of our knowledge no prior work has studied *MWKVMP* or *MKVMP* for $K > 2$. Since weighted matching problems arise in a variety of contexts, these results might find applications in other fields (e.g., VLSI) as well.

From a wireless networking perspective, we provide a Polynomial-Time Approximation Scheme (PTAS) for *MWKVMP* restricted to geometric unit-disk graphs, which can be used to represent the connectivity graph of a wide range of wireless networks. We also characterize the performance of “natural” greedy scheme under the same class of graphs. Although it has been known that the greedy scheme yields a constant factor approximation to *MWKVMP*, we are more interested in specific performance bounds of the scheme for all K . We note that both PTAS and the greedy algorithm can be used to construct scheduling policies that achieve a constant fraction of the capacity region under K -hop interference models, but they can be implemented in a limited class of wireless networks (e.g., wireless mesh networks) due to high complexity and requirement for centralized control.

We complement the results by showing that the maximal scheduling policy that can be implemented in a distributed manner with a low complexity achieves a guaranteed fraction of the *capacity region*. These results are encouraging as they indicate that one can develop distributed algorithms to achieve near optimal throughput in case of a wide range of wireless networks. Finally, we observe that the topological

constraints of the underlying graphs play a critical role to guarantee the throughput performance, and that the maximal scheduling policy can achieve an arbitrarily small fraction of the capacity region in general network graphs.

3. Hardness and Approximability Results

We now formulate the decision problems *KVMP* and *WKVMP* corresponding to *MKVMP* and *MWKVMP*, respectively, and prove that they are NP-Complete. We also show that *MKVMP* and *MWKVMP* cannot be approximable within $\Theta(|V|^{1/6})$ in polynomial time while we can approximate them within $\Theta(|E|/(\log|E|)^2)$. We begin with the following definitions.

Definition 1. For a given graph G and number m , *KVMP* is a decision process that determines whether G has a K -valid matching of size m .

Definition 2. For a given graph G , number m , and weight W_M , *WKVMP* is a decision process that determines whether G has a K -valid matching of size m and total weight W_M .

The following theorem shows that *WKVMP* \in NP, which also implies that *KVMP* \in NP.

Theorem 3. *WKVMP* \in NP for all K .

Proof. Given a certificate in the form of a list of edges, it can easily be verified in polynomial time whether that list corresponds to a set of m edges that are at a distance of K or more from each other and have a total weight of W_M or not. Thus, whether the set of edges constitute a K -valid matching of size m with a total weight of W_M can be verified in polynomial time. Hence, *WKVMP* \in NP. \square

We next show that *KVMP* is NP-Hard, which implies that the decision problem *WKVMP* is NP-Hard as well.

Theorem 4. *KVMP* is NP-Hard for $K \geq 2$.

Proof. The proof uses a novel technique reducing 3-CNF-SAT problem to *KVMP* [23]. Since their result is stronger, *MKVMP*, and hence *KWVMP*, are Nonapproximable for $K \geq 2$. \square

We now analyze the approximability of *MKVMP* for $K \geq 2$. We have the following result.

Theorem 5. Let η be a constant such that $(|V| + K|E|)^\eta = \Theta(|V|)$. Then, *MKVMP* (and hence, *MWKVMP*) for $K \geq 2$ is not approximable within $|V|^{\eta/2-\epsilon}$ for any $\epsilon > 0$, unless NP = P. Further, it is not approximable within $|V|^{\eta-\epsilon}$ for any $\epsilon > 0$, unless NP is equivalent to Zero-error Probabilistic Polynomial time (ZPP) problems [24].

Before we prove Theorem 5, we introduce some terminology. Consider a graph $G = (V, E)$. A subset of vertices is termed ‘‘independent’’ provided that no two vertices in the set have an edge between them. The classical Maximum

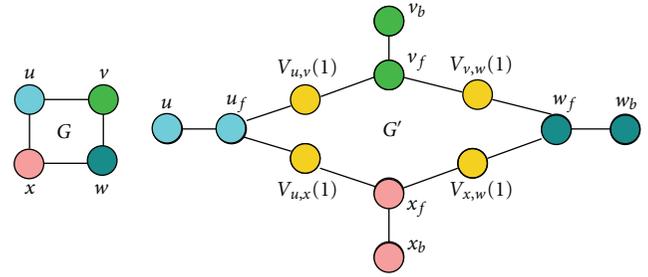


FIGURE 2: A graph G along with the graph G' constructed as specified in the proof of Theorem 5 for $K = 4$.

Independent Set Problem (*MISP*) is to find an independent subset of vertices of maximum possible cardinality. Note that we can easily convert *MKVMP* (4) to *MISP* by mapping an edge to a vertex and connecting two vertices when corresponding two edges are within K -hop distance. Hastad [25] has shown that *MISP* is not approximable within $|V|^{1/2-\epsilon}$ for any $\epsilon > 0$ unless NP = P, and it is not approximable within $|V|^{1-\epsilon}$ for any $\epsilon > 0$ unless NP = ZPP. We are now ready to prove Theorem 5.

Proof. We show that given a graph $G = (V, E)$, we can construct a graph $G'(V', E')$ in polynomial time such that a K -valid matching of G' has cardinality no smaller than that of the maximum independent set of G . Then we show that both $|V'|$ and $|E'|$ are $\Theta(|V| + K|E|)$, which is equal to $O(|V||E|)$. Finally, we will show that given a K -valid matching in G' , one can obtain an independent set of vertices in G with the same cardinality in polynomial time.

Suppose that *MKVMP* admits a polynomial time ρ -approximation scheme (PTAS). Given a graph G , one can construct the corresponding graph G' in polynomial time, and use the PTAS for *MKVMP* to obtain a K -valid matching of size at least $1/\rho$ times the cardinality of any maximum independent set of G . Then we can map it back to an independent set of vertices in G with the same cardinality, in polynomial time. This would then result in a ρ -approximation scheme for *MISP* of G , which, in view of the results in [25], would imply Theorem 5.

We next discuss how to construct the graph G' from G in polynomial time. We first consider even K .

- (1) For each vertex v in V , we place a pair of vertices v_f, v_b , and connect them with an edge.
- (2) For each edge uv in E , we connect the vertices u_f, v_f through a sequence of $K/2$ edges and $(K-2)/2$ vertices. Let the vertices be denoted by $V_{u,v}(1), \dots, V_{u,v}((K-2)/2)$, with $V_{u,v}(1)$ being the vertex adjacent to vertex u .

We denote the resultant graph as G' . Figure 2 illustrates an example of a graph G along with the constructed graph G' when $K = 4$. It is straightforward to see that the graph G' can

be constructed in polynomial (in $|V|$ and $|E|$) time. Also, we have

$$\begin{aligned} |V'| &= 2|V| + \left(\frac{K-2}{2}\right)|E| = O(|E||V|), \\ |E'| &= |V| + \left(\frac{K}{2}\right)|E| = O(|E||V|). \end{aligned} \quad (5)$$

Now, suppose that $\{v^1, v^2, \dots, v^m\}$ constitutes an independent set of vertices in G . It is then clear that $\{v_b^i, v_f^i\}_{i=1,2,\dots,m}$ constitutes a K -valid matching in G' . To see this, observe that since $\{v^1, v^2, \dots, v^m\}$ constitutes an independent set of vertices in G , we have $d_h(v^i, v^j) \geq 2$ for all $i, j \in \{1, 2, \dots, m\}$ with $i \neq j$. Hence, by the construction of G' , we have $d(v_b^i, v_f^j) \geq 2(K/2) = K$ for $i \neq j$. Then it follows that the graph G' has a K -valid matching of cardinality not smaller than the cardinality of the maximum independent set of G .

It remains to show that given a K -valid matching M in G' , one can, in polynomial time, obtain an independent set of vertices in G with the same cardinality. To this end, we propose a systematic construction method in Algorithm 1.

It is easy to see that the running time of Algorithm 1 is bounded above by a polynomial in $|V|$ and $|E|$. We check that the resulting set L from Algorithm 1 is indeed an independent set in G . It suffices to show that $d_h(u, v) \geq 2$ for all $v, u \in L$. Suppose that there exist two vertices $u, v \in L$ such that $d_h(u, v) = 1$. It then follows that there must exist edges $e_1, e_2 \in M$ such that $d(e_1, e_2) \leq K/2 + 2((K-2)/4) < K$, which contradicts our assumption that M is a K -valid matching.

Next, we discuss how to construct the graph G' for $K \geq 5$ and odd. We make a minor change in the construction of the graph. In the first step, instead of placing a pair of vertices for each vertex $v \in V$, we now place a triplet of vertices v_f, v_b, v_r (see Figure 3), and connect the pairs of vertices v_f, v_b and v_b, v_r with an edge. In the second step, for each edge $uv \in E$, we connect the vertices u_f, v_f through a sequence of $(K-1)/2$ edges and $(K-3)/2$ vertices. We denote the resulting graph as G' . We now have

$$\begin{aligned} |V'| &= 3|V| + \left(\frac{K-3}{2}\right)|E| = O(|E||V|), \\ |E'| &= 2|V| + \left(\frac{K-1}{2}\right)|E| = O(|E||V|). \end{aligned} \quad (6)$$

Similarly, we can check that the graph G' has a K -valid matching of cardinality no smaller than the cardinality of the maximum independent set of G . Suppose $\{v^1, v^2, \dots, v^m\}$ constitutes an independent set of vertices in G . We have $d_h(v^i, v^j) \geq 2$ for all $i, j \in \{1, 2, \dots, m\}$ with $i \neq j$ in G . Then by the construction of G' , we have $d(v_b^i, v_r^j) \geq 2 + 2((K-1)/2) = K + 1 > K$ for all $i \neq j$, and the result follows.

We show that given a K -valid matching in G' , we can obtain an independent set of vertices in G with the same cardinality in polynomial time. The construction algorithm is the same as Algorithm 1, except for the following three lines:

```

(1)  $L \leftarrow \emptyset$ 
(2) while  $M \neq \emptyset$  do
(3)   Pick an edge  $e \in M$ 
(4)   if  $e$  is of the form  $v_b v_f$  then
(5)      $L \leftarrow L \cup v$ 
(6)   else if  $e$  is of the form  $u v_{u,v}(1)$  then
(7)      $L \leftarrow L \cup u$ 
(8)   else if  $e$  is of the form  $V_{u,v}((K-2)/2)v$  then
(9)      $L \leftarrow L \cup v$ 
(10)  else if  $e$  is of the form  $V_{u,v}(i)V_{u,v}(i+1)$  then
(11)    if  $i \leq (K-2)/4$  then
(12)       $L \leftarrow L \cup u$ 
(13)    else
(14)       $L \leftarrow L \cup v$ 
(15)    end if
(16)  end if
(17)   $M \leftarrow M - e$ 
(18) end while
    
```

ALGORITHM 1: Constructing an independent set L in G from a K -valid matching M in G' , when K is even ($G' = (V', E'), M, L$).

- (i) Line 4: **if** e is of the form $v_b v_f$ or $v_r v_b$ **then**
- (ii) Line 8: **else if** e is of the form $V_{u,v}((K-3)/2)v$ **then**
- (iii) Line 11: **if** $i \leq (K-3)/4$ **then**

We check that the resulting set L is an independent set in G as follows. Suppose that L is not an independent set. Then there exist two vertices $v, u \in L$ such that $d_h(u, v) = 1$. Then by the construction of G' , there must exist two edges $e_1, e_2 \in M$ such that $d(e_1, e_2) \leq \max((K-1)/2 + 2((K-3)/4), (K-1)/2 + 2) < K$ for $K \geq 5$, which contradict our assumption that M is a K -valid matching. The running time of the algorithm is also bounded above by a polynomial in $|V|$ and $|E|$.

For $K = 3$, we can construct the graph G' as in $K = 4$ case, and prove the corresponding results accordingly. We omit the details. \square

From $K = O(|V|)$ and $|E| = O(|V|^2)$ in the above proof, the following result follows from Theorem 5.

Corollary 6. *MKVMP (and hence, MWKVMP) for $K \geq 2$ is not approximable within $|V|^{1/6-\epsilon}$ for any $\epsilon > 0$, unless $\text{NP} = \text{P}$. Further, it is not approximable within $|V|^{1/3-\epsilon}$ for any $\epsilon > 0$, unless $\text{NP} = \text{ZPP}$.*

Corollary 6 gives a lower bound on the approximation ratio of any polynomial time approximation algorithm for MWKVMP or MKVMP. The next result we have is opposite in flavor: it shows that there exists a polynomial time algorithm for MWKVMP whose approximation ratio is no worse than $\Theta(|E|/(\log |E|)^2)$.

Theorem 7. *MWKVMP can be approximated within a factor of $\Theta(|E|/(\log |E|)^2)$.*

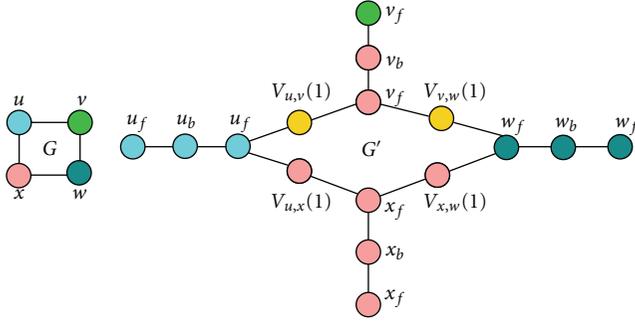


FIGURE 3: A graph G along with the graph G' constructed as specified in the proof of Theorem 5 for $K = 5$.

The following Corollary is an immediate consequence of Theorem 7.

Corollary 8. *MKVMP can be approximated within a factor of $\Theta(|E|/(\log |E|)^2)$.*

We define the Vertex Weighted Maximum Independent Set Problem (VWMISP), which is the following variation of the Maximum Independent Set Problem (MISP). Let $w(v)$ denote the weight of vertex v . VWMISP is to find an independent set L of vertices that maximizes $\sum_{v \in L} w(v)$. It is shown in [26] that VWMISP is approximable within $\Theta(|V|/(\log |V|)^2)$. We now proceed to the proof of Theorem 7.

Proof. Given a network graph $G = (V, E)$, we construct a graph G' from G in polynomial time, and approximately solve VWMISP in G' using the results of [26]. We can then obtain the corresponding K -valid matching in G' from the independent set in G .

We first construct $G' = (V', E')$ from $G = (V, E)$ as follows. For each edge $e \in E$, we generate a vertex v_e in V' with weight $w(v_e) = w(e)$. If two edges $e_1, e_2 \in E$ satisfy $d(e_1, e_2) \leq K - 1$, we connect the corresponding vertices v_{e_1}, v_{e_2} with an edge. The resulting graph G' is often called the *conflict graph* of G . Clearly, we have $|V'| = |E|$, and we can construct the conflict graph in polynomial time. From the construction, it is clear that for a K -valid matching in G , there exists an independent set of vertices in G' with the same weighted sum, and vice versa.

Now, using the results of [26], we can approximate VWMISP in polynomial time and obtain an independent set in G' with weight at least $\Theta((\log |V'|)^2/|V'|)$ times the weight of an optimal independent set. From the independent set in G' , we can reconstruct a K -valid matching in G with the same weight due to $|V'| = |E|$, in polynomial time. \square

4. MWKVMP for Geometric Unit-Disk Graphs

In this section, we focus on the MWKVMP problem for an important class of network graphs. In particular, we are interested in *geometric unit-disk graphs*, under which the connectivity and the interference constraints are determined

by the location of vertices. Specifically, the vertices are placed on a plane, two vertices are connected if and only if their distance is no greater than 1, and also interfere with each other if and only if their distance is no greater than K . Geometric graphs have been used extensively in the literature to model the connectivity of wireless networks [27, 28]. In this section, we show that MWKVMP can be approximated within a constant factor in case of unit-disk graphs. We also note that the results can also be extended to some other geometric graphs including the quasi-unit-disk graphs [29].

We start with redefining K -valid matching in geometric graphs. Let $\tilde{d}_S(u, v)$ denote the Euclidean distance between two nodes $u, v \in V$. We define the distance between edges and K -valid matching accordingly as (2), for $e_a = u_1 u_2, e_b = v_1 v_2 \in E$, we let

$$\tilde{d}(e_a, e_b) \triangleq \min_{i, j \in \{1, 2\}} \tilde{d}_S(u_i, v_j). \quad (7)$$

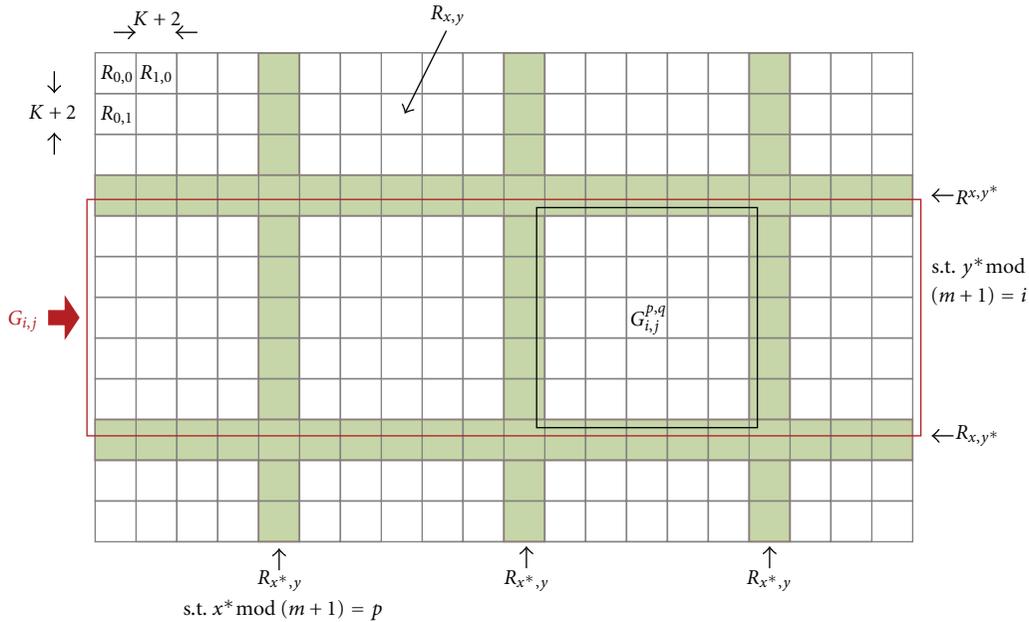
A set of edges M is said to be a “ K -valid matching” if for all $e_a, e_b \in M$ with $e_a \neq e_b$, we have $\tilde{d}(e_a, e_b) > K$. We also denote the set of K -valid matchings of the graph G by S_K .

4.1. PTAS for MWKVMP. Several NP-complete problems are known to admit PTAS when restricted to planar or geometric graphs. In [30], PTASs are developed for various NP-complete problems restricted to planar graphs. NC-approximation schemes for various NP-Hard and PSPACE-Hard problems restricted to geometric graphs are developed in [31]. Following the approach in [31], we now show that MWKVMP and, therefore, MKVMP admits a constant factor PTAS when restricted to geometric graphs. We present the polynomial time approximation algorithm for the completeness.

Consider a geometric graph $G = (V, E)$ with $r = 1$, specified using the coordinates of its vertices in the plane. We now present an algorithm that yields a K -valid matching with weight at least $(1 + \epsilon)^{-1}$ times the weight of an optimal K -valid matching in polynomial time, where $\epsilon > 0$ is a constant, and can be chosen to be arbitrarily small.

The basic technique is the following. Given any $\epsilon > 0$, we calculate the smallest possible m that satisfies $((m + 1)/m)^2 \leq 1 + \epsilon$. We divide the plane into grids of width $(K + 2)$ and height $(K + 2)$, and denote each grid by $R_{x, y}$ as shown in Figure 4. Each grid is left (top) closed and right (bottom) open. For each $i \in \{0, 1, \dots, m\}$, we partition the set of edges E into $s_i \geq 1$ disjoint sets $E_{i, 1}, \dots, E_{i, s_i}$ by removing edges whose two end-vertices are within R_{x, y^*} such that $y^* \bmod (m + 1) = i$. For $1 \leq j \leq s_i$, let $V_{i, j}$ be the smallest subset of V such that all edges in $E_{i, j}$ are of the form uv for some $u, v \in V_{i, j}$. Also, let $G_{i, j} = (V_{i, j}, E_{i, j})$, $1 \leq j \leq s_i$, and let $G_i \triangleq \bigcup_{j=1}^{s_i} G_{i, j}$. For each subgraph $G_{i, j}$, we find a K -valid matching of size at least $m/(m + 1)$ times the size of the optimal K -valid matching in $G_{i, j}$. Observe that since each subgraph has been separated by $(K + 1)$, the union of K -valid matchings for subgraphs $G_{i, 1}, \dots, G_{i, s_i}$ is a K -valid matching for the graph G . Using arguments similar to [31, 32], we then show that each iteration returns a K -valid matching with

- (1) Find the smallest m such that $((m+1)/m)^2 \leq 1 + \epsilon$.
- (2) Divide the plane into grids of width $(K+2)$ and height $(K+2)$. Each grid is denoted by $R_{x,y}$.
- (3) **for** $i = 0$ to m **do**
- (4) Partition E into s_i disjoint sets $\{E_{i,1}, E_{i,2}, \dots, E_{i,s_i}\}$ by removing edges whose two end-vertices are within R_{x,y^*} such that $y^* \bmod (m+1) = i$.
- (5) Let $G_{i,j}(V_{i,j}, E_{i,j})$ denote the subgraph induced by $E_{i,j}$ with $j \in \{1, 2, \dots, s_i\}$, and let $G_i \triangleq \bigcup_{j=1}^{s_i} G_{i,j}$.
- (6) **for** $j = 1$ to s_i **do**
- (7) **for** $p = 0$ to m **do**
- (8) Partition $E_{i,j}$ into $s_{i,j}$ disjoint sets $\{E_{i,j}^{p,1}, E_{i,j}^{p,2}, \dots, E_{i,j}^{p,s_{i,j}}\}$ by removing edges whose two end-vertices are within $R_{x^*,y}$ such that $x^* \bmod (m+1) = p$.
- (9) Let $G_{i,j}^{p,q}(V_{i,j}^{p,q}, E_{i,j}^{p,q})$ denote the subgraph induced by $E_{i,j}^{p,q}$ with $q \in \{1, 2, \dots, s_{i,j}\}$, and let $G_{i,j}^p \triangleq \bigcup_{q=1}^{s_{i,j}} G_{i,j}^{p,q}$.
- (10) **for** $q = 0$ to $s_{i,j}$ **do**
- (11) Obtain an optimal K -valid matching $KVM(G_{i,j}^{p,q})$.
- (12) **end for**
- (13) $KVM(G_{i,j}^p) \leftarrow \bigcup_{1 \leq q \leq s_{i,j}} KVM(G_{i,j}^{p,q})$
- (14) **end for**
- (15) $KVM(G_{i,j}) \leftarrow KVM(G_{i,j}^{p^*})$, where $p^* = \arg \max_{0 \leq p \leq m} w(KVM(G_{i,j}^{p,q}))$
- (16) **end for**
- (17) $KVM(G_i) \leftarrow \bigcup_{1 \leq j \leq s_i} KVM(G_{i,j})$
- (18) $KVM(G) \leftarrow KVM(G_{i^*})$, where $i^* = \arg \max_{0 \leq i \leq m} w(KVM(G_i))$
- (19) **end for**
- (20) $M \leftarrow KVM(G)$

 ALGORITHM 2: A $(1+\epsilon)$ -approximation scheme for $MWKVMP$ ($G' = (V, E), M$).

 FIGURE 4: Graph partition at iteration i in Algorithm 2.

weight at least $(m/(m+1))^2$ times the weight of an optimal K -valid matching in G . Our algorithm is described in detail in Table 2, and achieves $(1 + \epsilon)$ of the optimal performance. For the detailed analysis, we refer to our technical report [22].

Algorithm 2 has complexity of $|V|^{O(m^2)}$ (see [22]). Hence, even for a small m , the complexity is likely to be a high-order polynomial of $|V|$ and becomes a major obstacle to its implementation in practice. In the next subsection, we

show that a natural greedy algorithm with a lower complexity can approximate $MWKVMP$ within a constant factor under geometric unit-disk graphs.

4.2. Greedy Approach for $MWKVMP$. We study the performance of the greedy scheduling scheme illustrated in Algorithm 3. Note that the algorithm is greedy in the sense that it schedules links in decreasing order of the weight.

- (1) Arrange edges of E in descending order of weight as $e_1, e_2, \dots, e_{|E|}$
- (2) $M \leftarrow \phi$
- (3) **for** $i = 1$ to $|E|$ **do**
- (4) **if** $M \cup e_i$ is a K -valid matching **then**
- (5) $M \leftarrow M \cup e_i$
- (6) **end if**
- (7) **end for**

ALGORITHM 3: Greedy weighted K -valid matching algorithm ($G = (V, E), w : E \rightarrow \mathbb{R}, M$).

Some other works use the term “greedy” for a simpler scheme that schedules a set of links that no other links can be added to without violating the interference constraints. In this paper, we denote such a scheme by “maximal scheduling”, and differentiate from our greedy algorithm. It is well known that this greedy approach yields a 2-approximation algorithm for $MWMP$ in general network graphs under the 1-hop interference model [33], and a constant approximation algorithm in planar graphs under the 2-hop interference model [34]. However, the performance can be much worse for $K > 2$. In this section, we characterize the performance of the greedy approach in geometric unit-disk graphs by providing a lower bound for “all K .” We begin with some definitions.

Definition 9. The K -hop interference set of an edge $e \in E$, denoted by $I_K(e)$, is the set of edges $l \in E$ such that $\tilde{d}(e, l) \leq K$.

Definition 10. The K -hop interference degree of an edge $e \in E$, denoted by $\Delta_K(e)$, is defined as

$$\Delta_K(e) \triangleq \max_{M \in \mathcal{S}_K} |M \cap I_K(e)|. \quad (8)$$

Definition 11. The K -hop interference degree of the graph $G = (V, E)$, denoted by $\Delta_K(G)$, is defined as

$$\Delta_K(G) \triangleq \max_{e \in E} \Delta_K(e). \quad (9)$$

The following is the main result of this subsection.

Theorem 12. *The weight of the matching returned by Algorithm 3 is always within a factor $\Delta_K(G)$ of the weight of an optimal matching. Further, there exists a graph G for which the above ratio is tight.*

Proof. Let e_1 be the edge added to the matching during the first step by the greedy algorithm. Then, we have $w(e_1) \geq w(e)$ for all $e \in E$. Now, the optimal matching can contain at most $\Delta_K(G)$ edges belonging to $I_K(e_1)$, each with a weight no larger than $w(e_1)$. Let e_2 be the edge added to the matching during the second step by the greedy algorithm. Then, we have $w(e_2) \geq w(e)$ for all $e \in E \setminus I_K(e_1)$, where $A \setminus B$ denotes the set consisting of elements of A that are not in B . Moreover, the optimal matching can contain at most $\Delta_K(G)$

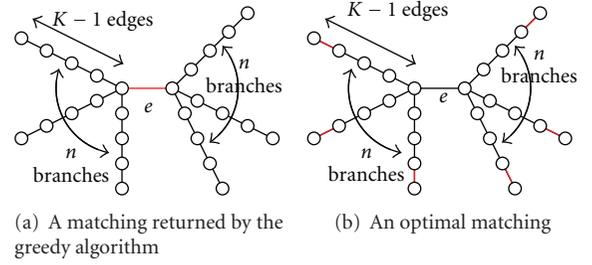


FIGURE 5: Comparison between a matching returned by the greedy algorithm and an optimal matching. All links in the graph have the same weight, and links included in each matching are marked in red. The greedy algorithm may schedule link e at the center of the graph while it is possible to schedule n links at the same time.

edges belonging to $I_K(e_2) \setminus I_K(e_1)$, each with a weight no larger than $w(e_2)$.

For $i \geq 1$, let $L_K(e_i) = I_K(e_1) \cup \dots \cup I_K(e_i)$. Arguing as above, it can be shown that during the i th step, the greedy algorithm adds an edge e_i to the matching that satisfies

$$w(e_i) = \max_{e \in E \setminus L_K(e_{i-1})} w(e), \quad (10)$$

and the optimal matching contains no more than $\Delta_K(G)$ edges belonging to $I_K(e_i) \setminus L_K(e_{i-1})$. Let e_m denote the last edge added to the matching by the greedy algorithm, and let \mathcal{O} denote the optimal matching. From the above discussion, it is clear that for $1 \leq i \leq m$, we have

$$\sum_{e \in \mathcal{O} \cap I_K(e_i) \setminus L_K(e_{i-1})} w(e) \leq \Delta_K(G) w(e_i). \quad (11)$$

Note that by convention $L_K(e_0) = \emptyset$. Summing over i , we obtain that

$$\sum_{e \in \mathcal{O}} w(e) \leq \Delta_K(G) \sum_{i=1}^m w(e_i), \quad (12)$$

proving the first part of Theorem 12.

To prove the second part, we consider a network graph G as shown in Figure 5. Let e denote the link at the center. In this example, we have $\Delta_K(G) = \Delta_K(e) = 2n$. One possible matching obtained using the greedy algorithm is shown in Figure 5(a). Note that the weight of this matching is 1. However, the weight of an optimal matching is $2n$ as shown in Figure 5(b). Thus, the greedy algorithm may return a matching whose weight is off by a factor of $\Delta_K(G)$ in comparison to the weight of an optimal matching. \square

Clearly, Figure 5 shows that $\Delta_K(G)$ can be of the order of $|E|$ in general graphs, and correspondingly, the performance of Algorithm 3 can be arbitrarily small when compared with the optimal performance. However, if the network graphs are governed by some geometric constraints, we can show that Algorithm 3 approximates the optimal scheduler by a constant.

Theorem 13. *The weight of the matching returned by Algorithm 3 is within a factor of 49 of the weight of an optimal matching in case of geometric unit-disk graphs.*

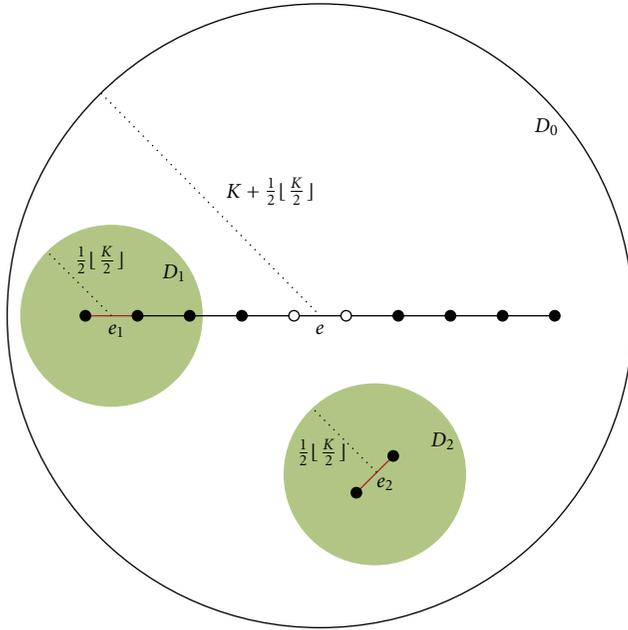


FIGURE 6: For an edge $e \in E$, we draw a large disk D_0 of radius $K + (1/2)\lfloor K/2 \rfloor$ centered at e . Then for each edge $e_x \in M^* \cap I_K(e)$, where M^* is a K -valid matching, we can draw a small disjoint disk D_x of radius $(1/2)\lfloor K/2 \rfloor$. By counting the number of small disks within D_0 , we can estimate a bound on the K -hop interference degree $\Delta_K(e)$.

Proof. From Theorem 12, it suffices to show that $\Delta_K(G) \leq 49$ for any geometric unit-disk graph $G = (V, E)$. To this end, we show that $\Delta_K(e) \leq 49$ for all edges $e \in E$.

At a time slot, let M^* denote a K -valid matching chosen by Algorithm 3. We consider the set of edges $M^* \cap I_K(e)$ for an edge $e \in E$. For each edge $e_x \in M^* \cap I_K(e)$, we draw a disk D_x of radius $(1/2)\lfloor K/2 \rfloor$ centered at the mid-point of e_x . Let e_i, e_j denote two edges in $M^* \cap I_K(e)$ with $i \neq j$. If there are no such pair of edges, then we have $\Delta_K(e) \leq 1$. Otherwise, it is clear from $d(e_i, e_j) > K$, two disks D_i and D_j do not intersect with each other as shown in Figure 6.

Now we consider a large disk D_0 of radius $K + (1/2)\lfloor K/2 \rfloor$ centered at the mid-point of e . Since we have $d(e, e_x) \leq K$ for all edges $e_x \in M^* \cap I_K(e)$, all disks D_x should be contained in D_0 . However, since no two disks intersect, the number of disks D_x in D_0 is bounded by

$$\frac{\pi(K + (1/2)\lfloor K/2 \rfloor)^2}{(\pi/4)\lfloor K/2 \rfloor^2} \leq 49, \quad (13)$$

for all $K \geq 2$. Hence, we have $|M \cap I_K(e)| \leq 49$ for all edge $e \in E$, which implies that $\Delta_K(G) \leq 49$. \square

Note that Algorithm 3 has complexity of $O(|E|)$ and can be implemented in a distributed manner [35].

Remark 14. The above results imply that PTAS of Algorithm 2 and the greedy algorithm of Algorithm 3 achieves a guaranteed fraction of weights. Let S_γ denote a class of scheduling policies such that at each time slot, the

weight of chosen schedule is no less than $\gamma \sum_{e \in \mathcal{O}} w(e)$. Then Algorithms 2 and 3 belong to S_γ with $\gamma = (m/(m+1))^2$ and $\gamma = 1/\Delta_K(G) \geq 1/49$, respectively. This property needs to be highlighted since distributed rate control algorithms that can deliver the performance of S_γ scheduling policy to end-users have been recently developed [8].

5. Throughput Guarantees of Scheduling Policies

Polynomial time algorithms developed in the earlier section can be used to construct scheduling policies that achieve a constant fraction of the capacity region. For example, it can be easily shown that a scheduling policy that belongs to S_γ achieves at least $\gamma\Lambda$, where Λ denotes the capacity region of the underlying network graph.

Although PTAS and the greedy algorithm achieve a guaranteed fraction of the capacity region, they require centralized control and/or a high complexity, which restrict their practical implementation within a limited class of wireless networks. In this section, we focus on throughput performance of scheduling policies. We show that even simpler scheduling policies that can be easily implemented in a distributed fashion have a provable throughput guarantee. Specifically, we show that the maximal scheduling policy of [8, 15] which is an S_0 scheduling policy can also achieve a guaranteed fraction of the capacity region in geometric unit-disk graphs, when all transmissions are carried out at certain fixed rates (i.e., rate control is not exercised).

5.1. Distributed Implementation for Geometric Unit-Disk Graphs. We start with the following definition of the maximal scheduling policy.

Definition 15. A subset M of edges is a *maximal schedule* if each edge $e \in M$ either has an empty queue, or satisfies $M \cap I_K(e) \neq \emptyset$. A scheduling policy is said to be a *maximal scheduling policy* if it chooses one of the maximal schedules for transmission at each time slot.

In words, the maximal scheduling policy ensures that if there are any packets waiting to be transmitted over an edge e , then either e or one of edges that interfere with e is scheduled for transmission. Note that an optimal solution to MWKVMP and the greedy algorithm are one of maximal scheduling policies while PTAS of Algorithm 2 is not a maximal scheduling policy.

Now we consider a network with single-hop fixed-rate sessions. Let Λ denote the capacity region of the network, that is, the set of session arrival rates for which the network can be stabilized under some scheduling policy. We have the following theorem.

Theorem 16. *In geometric unit-disk graphs under the K -hop interference model, any maximal scheduling policy can stabilize the network system for any set of session arrival rates within $(1/49)\Lambda$.*

Proof. It has been shown in [15] that any maximal scheduling policy achieves at least $1/\Delta_K(G)$ fraction of the capacity region. In other words, it stabilizes the network system for any set of arrival rates within $\Lambda/\Delta_K(G)$. From Theorem 13, we have that $\Delta_K(G) \leq 49$ in geometric unit-disk graphs, and hence, the result follows. \square

Note that a simple distributed maximal scheduling policy can be developed by extending the randomized maximal scheduling of [8] to the K -hop interference model. In this case, the complexity of the policy will be $O((\log |V|)^K)$.

Remark 17. Theorem 16 implies that the maximal scheduling policy can achieve $(1/49)\Lambda$ in the sense of *time average*. It can be contrasted with the results of PTAS and the greedy algorithm provided in Section 4, where they guarantee a constant fraction of weights *at each time slot*. Their average performance can be higher than the guaranteed fraction of weights. For example, it has been recently shown that the greedy algorithm achieves $(1/6)\Lambda$ [36].

5.2. Throughput Guarantees in Nongeometric Network Graphs.

The results provided in the previous section are encouraging as they indicate that one can develop simple distributed algorithms whose worst-case throughput is a nonvanishing fraction of the capacity region. Note that the results are obtained by admitting an arbitrarily small fraction of weights at a time slot, on the basis of geometric properties of unit-disk graphs. Although we have shown in Corollary 6 that *MWKVMP* cannot be approximated within a constant factor in general network graphs, it still remains unclear whether a simple distributed algorithm like the maximal scheduling policy can achieve a constant fraction of the capacity region in general network graphs. In the following, we show that the geometric constraints are indeed crucial to achieve the constant fraction of capacity region. To elaborate, we show that the greedy algorithm (and thus, the maximal scheduling policy as well) can achieve an arbitrarily small fraction of the capacity region in general network graphs.

We begin with some definitions. For a graph $G = (V, E)$, we consider a subset of edges $L \subset E$, and denote the set of all possible matching matchings on L by \mathcal{M}_L . Also let $\text{Co}(A)$ denote the convex hull of set A , that is,

$$\text{Co}(A) \triangleq \left\{ \sum_i \beta_i \vec{\alpha}_i \mid \beta_i \geq 0, \sum_i \beta_i = 1, \vec{\alpha}_i \in A \right\}. \quad (14)$$

Recently, it has been shown in [36, 37] that for a vector $\vec{v} \in \text{Co}(\mathcal{M}_L)$ and all $\epsilon > 0$, we can construct an arrival rate $\vec{\lambda} = (1 + \epsilon)\vec{v}$ such that the queues of all edges in L keep increasing under the greedy scheduling algorithm of Algorithm 3. Note that the optimal scheduler can serve the arrival rate $\vec{\lambda}$ if $\vec{\lambda} \in \text{Co}(\mathcal{M}_L)$. Therefore, in order to show that the greedy algorithm achieves no greater than $1/\sigma$, it suffices to find a subset L and two vectors $\vec{\mu}, \vec{v} \in \text{Co}(\mathcal{M}_L)$ such that $\sigma\vec{\mu} \geq \vec{v}$, where $\vec{a} \geq \vec{b}$ implies a component-wise inequality, that is, $a_i \geq b_i$ for all i .

Now we provide a systematic construction of network graphs such that we can find a subset of edges L and two

vectors $\vec{\mu}, \vec{v} \in \text{Co}(\mathcal{M}_L)$ satisfying $(1/\sigma)\vec{\mu} = \vec{v}$ with $\sigma \geq 3$. Once we find those two vectors, we have the upper bound $1/\sigma$ of the performance of the greedy algorithm.

Lemma 18. *There is a network graph $G_\sigma^K = (V, E)$ under the K -hop interference model with $K \geq 2$ such that two vectors $\vec{\mu}, \vec{v} \in \text{Co}(\mathcal{M}_L)$ with $L \subset E$ satisfy $(1/\sigma)\vec{\mu} \geq \vec{v}$ for $\sigma \geq 3$.*

Proof. We first describe our systematic construction of a graph, and then find two vectors in a subset of edges of the constructed network graph. Note that these two vectors should be a combination of maximal matchings in the subset of edges and one must be component-wise greater than the other by a factor of σ .

We construct the network graph G_σ^K with $K \geq 2$ and $\sigma \geq 3$ as follows.

Phase 1 (horizontal edges; see Figure 7(a) for an example of $K = 2$ and $\sigma = 3$). (1) Start with $N = 2\sigma K$ (or $2\sigma(K + 1)$ if K is odd) vertices. Place vertices on a cycle and name them in counter-clockwise order as $\{n_1^1, n_2^1, \dots, n_N^1\}$. Connect each vertex n_i^1 to its immediate neighbor $n_{i \oplus 1}^1$ for $1 \leq i \leq N$, where \oplus represents a modular addition by $N + 1$.

(2) Make the circle a centerless wheel by connecting each vertex n_i^1 to the opposite vertex $n_{i \oplus (N/2+1)}^1$ for $1 \leq i \leq N/2$. All vertices can be connected because N is an even number. Let W^1 denote the constructed wheel graph.

(3) Connect n_i^1 and $n_{i \oplus (K+[K/2])}^1$ using $\lfloor K/2 \rfloor$ -hop edges for $1 \leq i \leq N$. That is, for each i , add $(\lfloor K/2 \rfloor - 1)$ vertices between n_i^1 and $n_{i \oplus (K+[K/2])}^1$, say $\{\bar{n}_i(1), \bar{n}_i(2), \dots, \bar{n}_i(\lfloor K/2 \rfloor - 1)\}$, and connect them in sequence with edges $\bar{n}_i(k)\bar{n}_i(k+1)$ for $1 \leq k \leq \lfloor K/2 \rfloor - 2$. Also, add edges $n_i^1\bar{n}_i(1)$ and $\bar{n}_i(\lfloor K/2 \rfloor - 1)n_{i \oplus (K+[K/2])}^1$. If $K = 2$ or 3 , connect n_i^1 and $n_{i \oplus (K+[K/2])}^1$ directly.

(4) Repeat (3) with n_i^1 and $n_{i \oplus (jK+[K/2])}^1$ for $1 \leq j \leq \sigma - 2$.

(5) Construct another wheel graph W^K by duplicating W^1 , and name vertices on the wheel of W^K accordingly with superscript K .

Phase 2 (vertical edges; see Figure 7(b) for an example of $K = 2$ and $\sigma = 3$). (1) Connect n_i^1 and n_i^K using $(K - 1)$ -hop edges for all $1 \leq i \leq N$. That is, for each i , add vertices $\{n_i^2, n_i^3, \dots, n_i^{K-1}\}$ between n_i^1 and n_i^K , and connect them with edges $n_i^k n_i^{k+1}$ for $1 \leq k \leq K - 1$.

(2) Repeat (1) with n_i^1 and $n_{i \oplus jK}^K$ for $1 \leq j \leq \sigma - 2$.

As an example, all horizontal edges and a part of vertical edges of G_3^2 are shown in Figures 7(a) and 7(b).

Let $L \subset E$ be the set of edges *inside* two wheels, that is, $L = \{n_i^1 n_{i+N/2}^1, n_i^K n_{i+N/2}^K \mid 1 \leq i \leq N/2\}$. Let $L^1 = L \cap W^1$ and $L^2 = L \cap W^2$. Links in L are presented as solid black lines in Figure 7(a). Note that edges constructed in (3) and (4) of Phase 1 and in (1) and (2) of Phase 2 are designed to control interference among edges within and between wheels. If an edge l is active in L^1 (or in L^K), then edges constructed by (3) and (4) of Phase 1 allow at most $(\sigma - 1)$ other edges to be active in L^1 (or in L^2). Hence, we can activate at most σ edges in each wheel (see Figure 7(c)). However, the inter-wheel interference by vertical edges may reduce the number

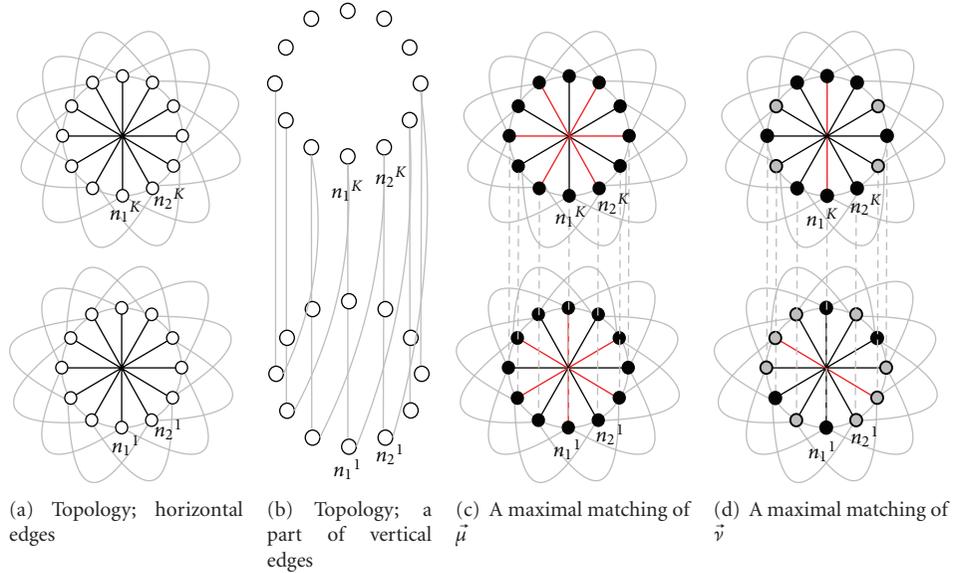


FIGURE 7: Example of network graph and matchings under the K -hop interference model, in which the greedy algorithm achieves no greater than $1/\sigma$ of the optimal performance ($K = 2$ and $\sigma = 3$). The subset L are the edges inside the cycles. (Solid black edges in (a).) An instance of maximal matching for $\vec{\mu}$ is shown in (c). Active edges are marked in red. By circulating the active edges in (c), we can obtain similar maximal matchings. Assume that $\vec{\mu}$ consists of those maximal matching with an identical weight. Similarly we can construct \vec{v} from maximal matchings like (d). Note that both $\vec{\mu}$ and \vec{v} serve all edges in L for the same amount of time, but a maximal matching of $\vec{\mu}$ has 3 times more active edges than a maximal matching of \vec{v} . Hence, it can be shown that $(1/\sigma)\mu_i \geq v_i$ for all $i \in L$. To make sure that the schedule of (d) is maximal, we color vertices interfered by the active edge in the upper wheel in black, and vertices interfered by the active edge in the lower wheel in gray.

of active edges. In (1) and (2) of Phase 2, we have constructed $(\sigma - 1)$ vertical edges per each vertex of each wheel. Since the vertical edges have $(K - 1)$ -hop, an active edge in L^1 can interfere with $(\sigma - 1)$ edges in L^K and vice versa. Assume that edges l_1^1 and l_1^K are active in L^1 and L^K , respectively. We can have at most $(\sigma - 1)$ more active edges in each wheel, that is, $\{l_2^1, \dots, l_{\sigma}^1\}$ in L^1 and $\{l_2^K, \dots, l_{\sigma}^K\}$ in L^K . However, if we choose edges l_1^1 and l_1^K such that l_1^1 interferes with all edges of $\{l_2^K, \dots, l_{\sigma}^K\}$ in L^K , and l_1^K interferes with all edges of $\{l_2^1, \dots, l_{\sigma}^1\}$ in L^1 , then we have only two active edges as a maximal matching in L , that is, l_1^1 and l_1^K (two red lines in Figure 7(d)). We design the network graph carefully such that a maximal matching can include from 2σ active edges to two active edges.

Now, we find two convex combinations of maximal matchings in L that one is component-wise greater than the other by σ . Consider two sets of maximal matchings; one with maximal matchings of 2σ active edges and the other with maximal matchings of 2 active edges. We first let $\vec{\mu} = \sum_{i=1}^K \beta_i M_i$ where $\beta_i = 1/K$ and each maximal matching M_i with $1 \leq i \leq K$ includes σ active edges $n_{i \oplus jK}^1 n_{(i+N/2) \oplus jK}^1$ and $n_{(i+1) \oplus jK}^K n_{(i+1+N/2) \oplus jK}^K$ for all $0 \leq j \leq \sigma - 1$. For the other vector, let $\vec{v} = \sum_{k=1}^{N/2} v_k M_k$ where $v_k = 2/N$ and each maximal matching M_k with $1 \leq k \leq N/2$ includes only two edges $n_k^1 n_{k+N/2}^1$ and $n_{k \oplus (\sigma-1)K}^K n_{(k+N/2) \oplus (\sigma-1)K}^K$. Note that M_i 's and M_k 's are valid maximal matchings in L . All active edges in L are either activated or interfered, and all active edges satisfy the interference constraints. Figures 7(c) and 7(d) illustrate an instance of M_i and M_k in G_3^2 , respectively. Active

edges are colored in red. To clearly show the interference in Figure 7(d), we color a vertex in black if it is interfered by the active edge in the upper wheel, and in gray if it is interfered by the active edge in the lower wheel.

Using the scheduling of $\vec{\mu}$ or \vec{v} , each edge in L is served exactly once during a unit time for $1/K$ by $\vec{\mu}$ or for $2/N$ by \vec{v} . Since $N = 2\sigma K$ (or $2\sigma(K + 1)$ if K is odd), we obtain that $(1/\sigma)\mu_j \geq v_j$ for all edge $e_j \in L$ and thus, $(1/\sigma)\vec{\mu} \geq \vec{v}$. \square

Lemma 18 immediately implies the following proposition.

Proposition 19. *Under the K -hop interference model, Algorithm 3 can achieve an arbitrarily small fraction of the optimal throughput.*

Proof. From Lemma 18 and the techniques of [37], we can find a traffic arrival with $\vec{\lambda} = (1 + \epsilon)\vec{v}$ for all $\epsilon > 0$ such that the system is unstable under the greedy scheduling algorithm. However, the optimal scheduling policy can support $\vec{\mu} \geq (1/\sigma)\vec{v}$, which follows that the achievable rate of the greedy algorithm is not greater than $1/\sigma$. Since σ can be arbitrarily large from our graph construction, the performance ratio can be arbitrarily small. \square

Proposition 19 lets us know that it is hard, if possible, to characterize the performance limits of the greedy algorithm (and thus the maximal scheduling policy as well) in arbitrary network graphs under the K -hop interference model.

6. Concluding Remarks

We consider the problem of throughput-optimal scheduling in wireless networks subject to interference constraints, which are modeled using a family of K -hop interference models. Under the assumption that each node transmits at a fixed power level (which can be different for different nodes), the optimal scheduling problems are shown to be weighted matching problems with constraints determined by the K -hop interference model. These weighted matching problems are termed Maximum Weighted K -Valid Matching Problems (*MWKVMPs*).

For $K = 1$, *MWKVMP* corresponds to the well-studied Maximum Weighted Matching Problem (*MWMP*) in the literature, which can be solved in polynomial time. We show that *MWKVMP* is NP-Hard for all $K \geq 2$ and provided upper and lower bounds on its approximability.

By restricting the problem to geometric unit-disk graphs, under which connectivities are determined by geometric distance between nodes, we show that *MWKVMP* admits a PTAS within a factor arbitrarily close to 1, and the “natural” greedy matching algorithm yields a 49-approximation to the optimal solution for all K . We emphasize that both PTAS and the greedy scheduling schemes achieve a guaranteed fraction of *weights* at every time slot. Combining these with the results in [8], it follows that both can be used to develop a joint solution of scheduling and rate control with provable (end-to-end) performance guarantees with multihop traffics.

However, since PTAS and the greedy algorithm have a polynomial time complexity and require centralized control, their implementations in practice are restricted within a limited class of wireless networks. We complement these results by further focusing on the throughput performance of scheduling policies. Specifically, we show that the maximal scheduling policy that is amenable to distributed implementation achieves 1/49 fraction of the capacity region under a setting with single-hop traffic and fixed rate transmissions. These results are encouraging as they indicate that one can develop simple distributed algorithms whose worst-case throughput is a nonvanishing fraction of the optimal throughput in the case of a wide class of wireless networks. Finally, we highlight that the geometric constraints are crucial for the maximal scheduling policy to achieve the throughput guarantees. We show that even the greedy scheduling algorithm, in the worst case, can result in an arbitrarily small efficiency without these constraints.

Acknowledgments

This work has been supported in part by the NSF Awards CNS-0626703 and CNS-0721236, and the ARO MURI Award W911NF-08-1-0238, USA, and in part by the New Professor Research Program of KUT (2010), Korea.

References

[1] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” *IEEE*

Transactions on Automatic Control, vol. 37, no. 12, pp. 1936–1948, 1992.

[2] L. Xiao, M. Johansson, and S. P. Boyd, “Simultaneous routing and resource allocation via dual decomposition,” *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, 2004.

[3] M. J. Neely, E. Modiano, and C. E. Rohrs, “Power allocation and routing in multibeam satellites with time-varying channels,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 138–152, 2003.

[4] L. Tassiulas and A. Ephremides, “Jointly optimal routing and scheduling in packet radio networks,” *IEEE Transactions on Information Theory*, vol. 38, no. 1, pp. 165–168, 1992.

[5] R. L. Cruz and A. V. Santhanam, “Optimal routing, link scheduling and power control in multi-hop wireless networks,” in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 702–711, April 2003.

[6] F. P. Kelly, A. K. Maulloo, and D. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, 1998.

[7] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, “A game theoretic framework for bandwidth allocation and pricing in broadband networks,” *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 667–678, 2000.

[8] X. Lin and N. B. Shroff, “The impact of imperfect scheduling on cross-layer rate control in wireless networks,” in *Proceedings of the Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1804–1814, March 2005.

[9] X. Wu and R. Srikant, “Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks,” in *Proceedings of the 25th Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '06)*, April 2006.

[10] A. L. Stolyar, “Maximizing queueing network utility subject to stability: greedy primal-dual algorithm,” *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.

[11] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*, Prentice-Hall, 2000.

[12] B. Hajek and G. Sasaki, “Link scheduling in polynomial time,” *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.

[13] H. N. Gabow, “Data structures for weighted matching and nearest common ancestors with linking,” in *Proceedings of the Symposium on Discrete Algorithms (SODA '90)*, 1990.

[14] H. Balakrishnan, C. L. Barrett, V. S. A. Kumar, M. V. Marathe, and S. Thite, “The distance-2 matching problem and its relationship to the MAC-layer capacity of ad hoc wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1069–1079, 2004.

[15] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, “Throughput and fairness guarantees through maximal scheduling in wireless networks,” *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 572–594, 2008.

[16] E. Modiano, D. Shah, and G. Zussman, “Maximizing throughput in wireless networks via gossiping,” *Performance Evaluation Review*, vol. 34, no. 1, pp. 27–38, 2006.

[17] S. Sanghavi, L. Bui, and R. Srikant, “Distributed link scheduling with constant overhead,” in *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (Sigmetrics '07)*, pp. 313–324, June 2007.

- [18] G. Sharma, C. Joo, and N. B. Shroff, "Distributed scheduling schemes for throughput guarantees in wireless networks," in *Proceedings of the 44th Annual Allerton Conference on Communications, Control, and Computing*, September 2006.
- [19] X. Lin and S. B. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 231–242, 2009.
- [20] C. Joo and N. B. Shroff, "Performance of random access scheduling schemes in multi-hop wireless networks," in *Proceedings of the 26th Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '07)*, pp. 19–27, May 2007.
- [21] L. J. Stockmeyer and V. V. Vazirani, "NP-completeness of some generalizations of the maximum matching problem," *Information Processing Letters*, vol. 15, no. 1, pp. 14–19, 1982.
- [22] C. Joo, G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," Tech. Rep., Korea University of Technology and Education, 2010, <http://netlab.kut.ac.kr>.
- [23] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "Maximum weighted matching with interference constraints," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '06)*, pp. 70–74, March 2006.
- [24] J. Gill, "Computational complexity of probabilistic turing machines," *SIAM Journal on Computing*, vol. 6, no. 4, pp. 675–695, 1977.
- [25] J. Håstad, "Clique is hard to approximate within $n^{1-\epsilon}$," *Acta Mathematica*, vol. 182, no. 1, pp. 105–142, 1999.
- [26] M. M. Halldórsson, "Approximations of weighted independent set and hereditary subset problems," *Journal of Graph Algorithms and Applications*, vol. 4, no. 1, pp. 1–16, 2000.
- [27] S. O. Krumke, M. V. Marathe, and S. S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *Wireless Networks*, vol. 7, no. 6, pp. 575–584, 2001.
- [28] S. H. Teng, *Points, spheres, and separators, a unified geometric approach to graph separators*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, CMU-CS-91-184, Pittsburgh, Pa, USA, August 1991.
- [29] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Ad-hoc networks beyond unit disk graphs," in *Proceedings of the 1st ACM Joint Workshop on Foundations of Mobile Computing (DIALM-POMC '03)*, pp. 69–78, San Diego, Calif, USA, September 2003.
- [30] B. S. Baker, "Approximation algorithms for NP-complete problems on planar graphs," *Journal of the ACM*, vol. 41, no. 1, pp. 153–180, 1994.
- [31] H. B. Hunt III, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, "NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs," *Journal of Algorithms*, vol. 26, no. 2, pp. 238–274, 1998.
- [32] D. S. Hochbaum and W. Maass, "Approximation schemes for covering and packing problems in image processing and VLSI," *Journal of the ACM*, vol. 32, no. 1, pp. 130–136, 1985.
- [33] D. Avis, "A survey of heuristics for the weighted matching problem," *Networks*, vol. 13, no. 4, pp. 475–493, 1983.
- [34] S. Ramanathan and E. L. Lloyd, "Scheduling algorithms for multihop radio networks," *IEEE/ACM Transactions on Networking*, vol. 1, no. 2, pp. 166–177, 1993.
- [35] J.-H. Hoepman, "Simple distributed weighted matchings," <http://arxiv.org/abs/cs/0410047v1>.
- [36] C. Joo, X. Lin, and N. B. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proceedings of the Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '08)*, pp. 1103–1111, April 2008.
- [37] C. Joo, X. Lin, and N. B. Shroff, "Performance limits of greedy maximal matching in multi-hop wireless networks," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 1128–1133, December 2007.