A Systematic Study of the Delayed Column Generation Method for Optimizing Wireless Networks

Yu Cheng
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, IL 60616, USA
cheng@iit.edu

Xianghui Cao
Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, IL 60616, USA
xcao10@iit.edu

Xuemin (Sherman) Shen
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo Ontario N2L 3G1
Canada
xshen@bbcr.uwaterloo.ca

Devu Manikantan Shila
United Technologies Research Center
Hartford, CT 06108 USA
manikad@utrc.utc.com

Hongkun Li
InterDigital
King of Prussia, PA 19406
USA
hongkun.li@interdigital.com

ABSTRACT

The main-thread approach for optimizing the throughput capacity over a multihop wireless network is to apply a multi-commodity flow (MCF) formulation, augmented with a scheduling constraint derived from the conflict graph associated with the network. A fundamental issue with the conflict graph based MCF formulation is that finding all independent sets (ISs) for scheduling is NP-hard in general. If we express the MCF formulation in a matrix format, the constraint matrix will contain a very large number of columns, with each IS being associated with one column. According to the linear programming theorem, such a type of problem can be addressed with the delayed column generation (DCG) method. Unfortunately, applications of the DCG in wireless networks have not received much attention. To the best of knowledge, none of the existing work conducted theoretical studies of the performance of DCG in wireless networks. In this paper, we study the DCG method in the context of a general network flow problem. With a protocol interference model, we rigorously prove that searching an entering column in the DCG operation is equivalent to a maximum weighted independent set (MWIS) problem. A prominent theoretical contribution of this paper is the theorem that: if an MWIS approximation algorithm with the approximation ratio β (< 1) is applied in the DCG method, the maximum flow solved will be at least β of the optimal solution. Furthermore, the DCG method is also applied to the multi-radio multi-channel (MR-MC) networks. With extensive numerical results comparing to the existing methods, we show that the DCG method achieves the most preferred tradeoff between computation complexity and network capacity and maintains good scalability when addressing large-scale networks, particularly in the complex MR-MC context.

Categories and Subject Descriptors

C.m [**Computer-Communication Networks**]: Miscellaneous

Keywords

Network capacity optimization; delayed column generation; MWIS; approximation ratio; multi-radio multi-channel networks

1. INTRODUCTION

The *network capacity* of fundamental importance to a multihop wireless network is the total throughput traversing the given set of ingress/egress edge nodes. The maximum network capacity is normally coupled with optimal routing and scheduling to form a *network dimensioning* issue [2, 3]. A fundamental issue in the multihop wireless networks is that the neighboring hops along a path, when transmitting over the same spectrum, have to contend for the channel access and can not transmit at the same time. The *conflict graph* is the popular tool to model the interference among different wireless links [4–6]. The main-thread approach for wireless network dimensioning is to apply a *multi-commodity flow* (MCF) formulation, augmented with constraints derived from the conflict graph [2–4, 18].

The conflict-graph constraints are normally based on the concept of independent set [4]: at a moment, the links that can have successful transmissions simultaneously in the network form an independent set over the conflict graph associated with the network. The MCF formulation with the independent set constraints can generate an optimal scheduling (under the assumption of a synchronized slotted system): the maximal independent sets (MISs) take turns in grabbing the channel for data transmission, with the proportion of transmission time for each set determined by the MCF solution. The recent work in [9] develops a generic multi-dimensional conflict graph (MDCG) technique, which enables the conflict graph based MCF formulation for a multi-radio multi-channel (MR-MC) wireless network.

While the conflict graph based MCF formulation is a linear programming problem, finding all ISs to be used is NP-hard in general. A random algorithm for MIS search is proposed in [4] and widely adopted in the literature, which provides a framework while more MISs can be obtained with more rounds of computation. However, in a large scale network (especially for the MR-MC case with the

MDCG involved), the random search method incurs a heavy computation overhead to find a large enough number of MISs for a good approximate MCF solution. The work in [9] develops a heuristic algorithm to search a smaller set of critical MISs, which can considerably reduce the overhead in MIS search compared to the random method. The overhead of searching critical MISs is still significant anyhow in a large-scale network. Moreover, those heuristic MIS search algorithms can not provision a guaranteed capacity region of the MCF solutions.

Regarding the conflict graph based MCF linear programming (LP) problem, if we express all the constraints in a matrix format, the constraint matrix will have a very large number of columns where each IS will be associated with one column.[1] Both searching all the columns and storing the columns for MCF solutions incur high complexity. According to the LP theorem, such a type of problem can be addressed with the delayed column generation (DCG) method [11], where a new column will be iteratively searched online based on the current feasible solution for the most reduced cost till the optimal solution is reached. Unfortunately, applications of the DCG in wireless networks have not received enough attention. We only noticed a couple of studies, which focus on very specific cases. The work in [12] applies the DCG method to study a minimum scheduling problem, with the objective to assign at least one time slot to each node such that the total number of time slots is minimized. In [13, 14], the DCG method is applied to study joint routing, link scheduling and power control in wireless networks. The fundamental issue is that none of the existing studies, to the best of knowledge, conducted theoretical studies of the performance of DCG in wireless networks. In a wireless network with a certain interference model, searching an entering column according to the DCG will be a maximum weighted independent set (MWIS) problem [16] (to be proved rigorously in this paper). Thus, in practice, an approximation algorithm needs to be used to find an entering independent set with a reasonable computation overhead. *A prominent theoretical contribution of this paper is that we prove the following theorem: if an MWIS approximation algorithm with the approximation ratio β (< 1) is applied in the DCG method, the maximum flow solved under such an approximation algorithm will be at least β of the optimal solution.* More specifically, this paper has four-fold contributions.

- We study the DCG method in the context of a general network flow problem, and rigorously prove that searching an entering column in the DCG operation is equivalent to an MWIS problem.

- We conduct theoretical analysis to obtain the lower bound of the network flow, when the MWIS problem involved in the DCG method (which is NP-hard) is addressed with an approximation algorithm. Specifically, if an MWIS approximation algorithm with the approximation ratio β (< 1) is used, the maximum flow solved will be at least β of the optimal solution. Furthermore, we show such a lower bound also applies to the multi-commodity flow problem and the network flow problem with a fairness constraint.

- The DCG method is also applied to the MR-MC network, exploiting the tuple-based network model and multi-dimensional conflict graph technique developed in [9].

- We present extensive numerical results to demonstrate the performance of the DCG method in terms of computation overhead and the achieved network flow capacity, with comparison to that with the random search of MISs [4] and that with the heuristic search of critical MISs [9]. It turns out the DCG method achieves the most preferred tradeoff between computation complexity and network capacity, and maintains good scalability when addressing large-scale MR-MC networks. We also examine the performance of DCG method under different MWIS approximation algorithms.

[1]Although it is sufficient to just involve maximal independent set (MIS) in the scheduling [4], in this paper, we generally represent the scheduling involving all ISs for the convenience of theoretical analysis.

The remainder of this paper is organized as follows. Section 2 reviews more related work. Section 3 describes the system model. Section 4 conducts a theoretical study of the performance of DCG in solving various network flow problems. In Section 5, the DCG method is applied to MR-MC networks. Section 6 presents extensive numerical results to evaluate the performances of DCG, with comparison to some existing methods. Section 7 gives the concluding remarks.

2. RELATED WORK

The MR-MC wireless networks have received substantial attention in the last few years, especially in the context of wireless mesh networks (WMNs) [1,10,20]. Such a networking model can significantly increase network capacity by simultaneously exploiting multiple non-overlapping channels through different radio interfaces. Compared to the traditional SR-SC networks, MR-MC networking takes place in a multi-dimensional resource space, with dimensions defined by radio interfaces, links, and channels. The central issue of resource allocation in such a multi-dimensional space is to find solutions for a set of coupled problems including *scheduling, channel and radio assignment, and routing* [1, 19], with the objective to optimize the network capacity. The conflict graph tool however did not achieve the similar popularity in MR-MC networks as in SR-SC networks [6, 7], mainly because the link conflict graph is not sufficient for MR-MC networks to describe the extra conflict relations in competing for radio interfaces and channels. The multi-dimensional conflict graph technique developed in [9] fill the gap between optimizing SR-SC and MR-MC networks. Each vertex in the MDCG represents a link-radio-channel (LRC) tuple, a basic resource point in the multi-dimensional resource space. The LRC-tuple based MISs can then be searched over the MDCG to specify the scheduling constraints in the MCF formulation.

Searching all independent sets is however NP-hard in general. While the random search algorithm [4] and the critical MIS search algorithm [9] can reduce the computation overhead in some degree, neither of the two methods can provide performance guarantee of the MCF capacity. With an LP formulation, the independent set induces a large number of columns to the constraint matrix of the MCF problem. The delayed column generation method is expected to be decomposition method addressing such a problem. While the existing studies [12,13] are limited to special scenarios without theoretical analysis, this paper provides a systematic study of the DCG method in the general context of network flow problem. We show that searching an entering column in the DCG method is equivalent to an MWIS problem. The MWIS is a classic problem in combinatorial optimization, which is NP-hard in general and by nature calls for approximation algorithms. For example, [16] studies a simple greedy algorithm which iteratively removes the vertex (and its neighbors) having minimal weighted degree each time from the remaining graph. This method is combined with a linear programming to form an LP-based algorithm [16]. A minimum weighted vertex cover based approximation algorithm is proposed in [17].

In this paper, we compare the performances of various embedded MWIS approximation algorithms in our DCG method, and the results show the greedy algorithm is the most efficient one.

3. SYSTEM MODEL

In this section, we mainly present the system model in the SR-SC context. Compared to the MR-MC case, an SR-SC network provides a less entangled environment for theoretically studying the performance of DCG. In Section 5, we do show that the theoretical results obtained in the SR-SC context can be extended to the MR-MC case.

Consider an SR-SC wireless network as a directed graph $G(\mathcal{N}, \mathcal{L})$ with node set $\mathcal{N}$ and physical link set $\mathcal{L}$. A directed link in a graph is also termed as an arc. Let $|\mathcal{N}|$ and $|\mathcal{L}|$ denote the number of nodes and links, respectively. A link $\ell \in \mathcal{L}$ is also denoted by the pair of transmitter and receiver (a, b) for which the capacity is $w_{(a,b)}$. To form a link, nodes a and b should be within each other's *communication range*. We consider the nodes are indexed from 1 to $|\mathcal{N}|$ and links indexed from 1 to $|\mathcal{L}|$.

We adopt the *protocol interference model* defined by an *interference range* [21]. We assume that every node has the same interference range. Two nodes interfere with each other if they are within each other's interference range. Two links conflict with each other if they have interfering nodes. Under such an interference model, a link could have a successful transmission only when both the sender and receiver are free of interference. The interference-free requirement at the sender side ensures that it can receive the acknowledgement returned by the receiver. The interferences among the links can be described by an undirected conflict graph. Each vertex in the conflict graph represents a link in the original network. An edge between two vertices indicates a conflict relationship between the corresponding links in the original network. An independent set over the conflict graph then represents a set of links that can transmit simultaneously in the original network.

We consider a slotted system, where time is divided into slots of unit length. Different ISs share the time for transmission, forming a *scheduling*. Let $\mathcal{I}$ denote an IS over the conflict graph and $\mathcal{M}$ the set of all the ISs. The convex hull of all the ISs, $Co(\mathcal{M})$, defines the network capacity region under scheduling. Let $r_{(a,b)}$ denote a feasible flow allocation over link (a, b) and $\mathbf{r}$ the flow rate vector over all the links. We have $\mathbf{r} \in Co(\mathcal{M})$.

In the sequel, we use boldface letters in lower case and uppercase to denote vectors and matrices, respectively. A vector by default is a column vector. The operator $(')$ indicates the transpose of a vector or matrix. We use $\mathbf{0}$ and $\mathbf{1}$ to denote the vectors of 0's and 1's, respectively, whose dimensions are clear in the context.

4. DELAYED COLUMN GENERATION

In this section, the theoretical performance of DCG is first studied based on a standard network flow problem with a single flow. We then extend the results to some other important network flow problems, such as maximum flow, multi-commodity flow, and MCF with fairness. Before we present the network flow formulation, let us look at the basic flow conservation constraint and the IS based scheduling constraint.

4.1 Flow Conservation Constraint

Consider a network with a single flow f from the source node $s(f)$ to the destination node $d(f)$. Let $\mathbf{g} = (g_1, g_2, \cdots, g_{|\mathcal{N}|})$ denote the input vector, with $g_i = 0$ for $i \neq s(f)$ and $g_{s(f)} = -g_{d(f)}$. Define the *node-arc incidence matrix* $\mathbf{H}$ as follows: its dimensions are $|\mathcal{N}| \times |\mathcal{L}|$ (each row corresponds to a node and each column corresponds to an arc (i.e., a link)) and its (i, k)th entry h_{ik} is associated with the ith node and the kth link. We let

$$h_{ik} = \begin{cases} 1, & \text{if } i \text{ is the start node of the } k\text{th arc,} \\ -1, & \text{if } i \text{ is the end node of the } k\text{th arc,} \\ 0, & \text{otherwise.} \end{cases}$$

With the node-arc incidence matrix, the flow conservation constraint can be expressed as

$$\mathbf{Hr} = \mathbf{g} \tag{1}$$

Note that the rows of the matrix $\mathbf{H}$ sum to the zero vector and are therefore linearly dependent. In fact, the last flow conservation constraint at node $|\mathcal{N}|$ is a consequence of the flow conservation constraints at the other nodes, and can be omitted without affecting the feasible set. A *truncated node-arc incident matrix* $\tilde{\mathbf{H}}$ can be defined, which consists of the first $|\mathcal{N} - 1|$ rows of the matrix $\mathbf{H}$. Let $\tilde{\mathbf{g}} = (g_1, \cdots, g_{|\mathcal{N}|-1})$. According to [11], the flow conservation constraints without redundancy can be expressed as

$$\tilde{\mathbf{H}}\mathbf{r} = \tilde{\mathbf{g}} \tag{2}$$

We can index the destination node as the last node and truncate it, so $\tilde{\mathbf{g}}$ consists of only zero and positive values. For convenience, we later just use $\mathbf{H}$ and $\mathbf{g}$ to represent the trunked node-arc incidence matrix and the associated input vector.

4.2 Scheduling Constraint

In wireless network, the link capacity under scheduling is constrained by the convex hull of the independent sets. The total flow allocation over link (a, b) should satisfy its capacity constraint.

$$r_{(a,b)} \leq \sum_{m:(a,b)\in\mathcal{I}_m} \alpha_m w_{(a,b)} \tag{3}$$

$$\sum_{m=1}^{|\mathcal{M}|} \alpha_m = 1 \tag{4}$$

where the real number α_m $(0 \leq \alpha_m \leq 1)$ represents the proportion of time scheduled to the mth IS.

LEMMA 1. *There exists a scheduling* $(\alpha_1, \cdots, \alpha_u)$, *so that any feasible flow allocation within the capacity region, i.e.,* $\mathbf{r} \in Co(\mathcal{M})$, *has an exact scheduling as*

$$r_{(a,b)} = \sum_{m:(a,b)\in\mathcal{I}_m} \alpha_m w_{(a,b)}, \quad \forall (a,b) \in \mathcal{L} \tag{5}$$

$$\sum_{m=1}^{|\mathcal{M}|} \alpha_m = 1. \tag{6}$$

PROOF. Consider feasible flow allocation within the convex hull. A link (a, b) is termed as *over-scheduled* if

$$r_{(a,b)} < \sum_{m:(a,b)\in\mathcal{I}_m} \alpha_m w_{(a,b)}. \tag{7}$$

We are going to show that the scheduling can be adjusted so that every over-scheduled link becomes exactly scheduled, and the originally exactly scheduled links maintain intact.

Pick an over-scheduled link (a, b) and assume there are n independent sets, with index $(i_1, i_2, \cdots, i_n)$, which have positive time allocation in the scheduling (7). We take the ratio

$$\beta = \frac{r_{(a,b)}}{\sum_{m:(a,b)\in\mathcal{I}_m} \alpha_m w_{(a,b)}} = \frac{r_{(a,b)}}{\sum_{j=1}^{n} \alpha_{i_j} w_{(a,b)}}. \tag{8}$$

Let i' index the independent set that contains the links in $\mathcal{I}_i/(a,b)$, that is, the links of $\mathcal{I}_i$ excluding link (a,b). Then we adjust the scheduling as

$$\tilde{\alpha}_{i_j} = \beta\alpha_{i_j}, \tag{9}$$

$$\tilde{\alpha}_{i'_j} = \alpha_{i'_j} + (1-\beta)\alpha_{i_j} \tag{10}$$

for $j = 1, \cdots, n$. We call the operations in (9) and (10) as a splitting procedure. With a splitting procedure, it can be seen that link (a,b) will become exactly scheduled. The scheduled capacities of other links in $\mathcal{I}_i/(a,b)$ do not change, for which the splitting procedure just redistributes the scheduling time between $\mathcal{I}_i$ and $\mathcal{I}_{i'}$ but maintains the total time allocation. Note that if an independent set includes only the link (a,b), the splitting procedure will lead to a positive time allocation to the empty set, indexed as $i'_j = 0$.

Using the $(\tilde{\alpha}_{i_1}, \tilde{\alpha}_{i'_1}, \cdots, \tilde{\alpha}_{i_n}, \tilde{\alpha}_{i'_n})$ to update the corresponding time allocation in the original scheduling, we get another scheduling denoted as $\tilde{\boldsymbol{\alpha}}$. For $k \notin \{i_1, i'_1, \cdots, i_n, i'_n\}$, we have $\tilde{\alpha}_k = \alpha_k$. Further, we can remove the empty set (if it has positive time allocation) from the scheduling with the normalization operations:

$$\tilde{\alpha}_k = \frac{\tilde{\alpha}_k}{1-\tilde{\alpha}_0}, \quad \text{for } k = 1, \cdots, n.$$

We can then move on to pick another over-scheduled link and apply the splitting procedure to make it exactly scheduled. The scheduling $\tilde{\boldsymbol{\alpha}}$ is then updated according to the splitting. Such a procedure of splitting and schedule updating can be iteratively applied till all over-scheduled links become exactly scheduled. As a result, the final scheduling is still valid. $\square$

The exact scheduling lemma has benefits in two aspects. a) It allows a standard linear programming formulation of the network flow problem over a wireless network, which will facilitate the theoretical analysis of DCG performance. b) It reduces computation complexity in the DCG algorithm. In a non-standard LP problem, extra computation overhead will be required to handle slack variables associated with those inequality constraints [11], if we want to search a new entering column only over the IS columns.

4.3 Delayed Column Generation

Let $\mathbf{u}$ denote a vector of link costs. Consider the standard network flow problem $\mathcal{P}_1$

$$\text{minimize} \quad \mathbf{u}'\mathbf{r} \tag{11}$$

subject to:

$$\mathbf{Hr} = \mathbf{g} \tag{12}$$

$$\mathbf{r} = \mathbf{W}\boldsymbol{\alpha} \tag{13}$$

$$\mathbf{1}'\boldsymbol{\alpha} = 1 \tag{14}$$

$$\mathbf{r} \geq \mathbf{0} \tag{15}$$

$$\boldsymbol{\alpha} \geq \mathbf{0}. \tag{16}$$

where the matrix $\mathbf{W}$ consists of all the independent sets (excluding the empty set), with each column as one different independent set. Consider that there are M non-empty independent sets can be used for the optimization problem. Each independent set $\mathbf{w}_m$ is a vector of $|\mathcal{L}|$ elements, with

$$\mathbf{w}_m(i) = \begin{cases} w_i, & \text{if link } i \text{ is in this set} \\ 0, & \text{otherwise} \end{cases}$$

The matrix $\mathbf{W}$ can be expressed as

$$\mathbf{W} = \begin{bmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_M \\ | & & | \end{bmatrix}. \tag{17}$$

In the problem $\mathcal{P}_1$, the decision variables are $\mathbf{r}$ and $\boldsymbol{\alpha}$. The objective function is equivalent to $\mathbf{u}'\mathbf{r} + \mathbf{0}'\boldsymbol{\alpha}$. To get a standard LP problem, we redefine the decision variables as $\mathbf{x} = (\mathbf{r}', \boldsymbol{\alpha}')'$. The cost vector $\mathbf{c} = (\mathbf{u}', \mathbf{0}')'$. Further we rewrite the condition (13) as

$$\mathbf{r} - \mathbf{W}\boldsymbol{\alpha} = \mathbf{0} \tag{18}$$

The problem $\mathcal{P}_1$ can then be reformulated as a standard LP problem $\mathcal{P}_2$

$$\text{minimize} \quad \mathbf{c}'\mathbf{x} \tag{19}$$

subject to:

$$\mathbf{Ax} = \mathbf{b} \tag{20}$$

$$\mathbf{x} \geq \mathbf{0} \tag{21}$$

where

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{H}_{(|\mathcal{N}|-1)\times|\mathcal{L}|} & \mathbf{0}_{(|\mathcal{N}|-1)\times M} \\ \hline \mathbf{I}_{|\mathcal{L}|\times|\mathcal{L}|} & -\mathbf{W}_{|\mathcal{L}|\times M} \\ \hline \mathbf{0}_{1\times|\mathcal{L}|} & \mathbf{1}_{1\times M} \end{array}\right] \tag{22}$$

$$\mathbf{b} = (\mathbf{g}'_{1\times(|\mathcal{N}|-1)}, \mathbf{0}'_{1\times|\mathcal{L}|}, 1). \tag{23}$$

Note that the constraint (20) presents in a compact manner the constraints (12), (13), and (14). For the convenience of presentation, we term the columns in $\mathbf{A}$ associated with all the ISs as *right sub-matrix* and the other part as *left sub-matrix*.

The challenge to solve the problem $\mathcal{P}_2$ is that the number of independent sets is in the order of $O(2^{|\mathcal{L}|})$, and it is known that searching all independent sets is an NP-hard problem. Even if all the independent sets can be searched, the number of columns is so large that it is impossible to generate and store the entire matrix $\mathbf{A}$ in memory.

Experience with large problems indicates that, usually, most of the columns never enter the basis, and we can therefore afford not to ever generate those unused columns. The *delayed column generation* method [11] is a sequence of iterations. At the beginning of an iteration, we have a basic feasible solution to the original problem, and an associated basis matrix. We search for a variable with negative reduced cost, possibly by minimizing $\bar{c}_i$ over all i, i.e., solving the problem

$$\text{minimize} \quad \bar{c}_i; \tag{24}$$

if none is found, the algorithm terminates. Suppose that we have found some j such that $\bar{c}_j < 0$. We then form a collection of columns $\mathbf{A}_i$, $i \in \mathcal{Q}$, which contains all of the columns that have been selected before and the new entering one. We then define the *restricted* problem

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in\mathcal{Q}} c_i x_i \\ \text{subject to} \quad & \sum_{i\in\mathcal{Q}} \mathbf{A}_i x_i = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

The solution from the restricted problem is another basic feasible solution to the original problem and serves as the start point for next iteration.

Selecting an initial basic feasible solution. We can select an initial feasible solution by the following algorithm:

Step 1: Randomly select a link, and then search an MIS starting from this link.

Step 2: Randomly select another link from the set of links that have not been included in any of the existing MISs, and then search an MIS starting from the selected link.

Step 3: Repeat Step 2 until all the links are covered.

If n MISs have been selected, we can get a scheduling that assigns $1/n$ of the time to each MIS. Then each link will have the capacity at least w_l/n, with w_l the physical link capacity of link l. In fact, each link capacity under the scheduling could be exactly determined based on how many MISs it is involved in. Given the link capacity under scheduling, a feasible commodity flow can be easily constructed by setting it as the bottleneck capacity of a path between the source and the destination. The set $\mathcal{Q}$ can include all the columns in left sub-matrix and the columns associated with the MISs scheduled. As we already constructed a basic feasible solution over such $\mathcal{Q}$, we know that the restricted problem must have a solution to start the DCG iterations.

Theorem 1. *The DCG problem to find an entering column can be mapped to an MWIS problem.*

Proof. Note that when we apply the delayed column generation method, use matrix $\mathbf{B}$ to denote the matrix formed by columns associated with the basic feasible variables $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$. Define $\mathbf{p}' = \mathbf{c}'_B\mathbf{B}^{-1}$. To find a new column with improved solutions is to find a column

$$\bar{c}_i = c_i - \mathbf{p}'\mathbf{A}_i \le 0.$$

With delayed column generation, the candidate columns are only in the right sub-matrix of $\mathbf{A}$. For those columns, the associated cost variables are all 0. To pick the one with the largest value of cost reduction, the problem is

$$\text{minimize} \quad -\mathbf{p}'\mathbf{A}_j. \tag{25}$$

Let $\mathbf{w}_j$ be the independent set vector contained in $\mathbf{A}_j$. The minimization problem (25) is then equivalent to

$$\text{minimize} \quad \sum_{i=|\mathcal{N}|}^{|\mathcal{N}|+|\mathcal{L}|-1} p_i \mathbf{w}_j(i-|\mathcal{N}|+1) - p_{|\mathcal{N}|+|\mathcal{L}|}.$$

and further equivalent to

$$\text{maximize} \quad \sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i) \tag{26}$$

by defining $k_i = i + |\mathcal{N}| - 1$. $\square$

Remark: Note that an MWIS algorithm normally considers only non-negative weights. However, in the MWIS problem shown in (26), the weight associated with a link (i.e., $-p_j$) may take either negative or positive values. If some links have negative weights, we can replace the negative weights with a weight of zero and runs the MWIS algorithm. For the MWIS identified, we can exclude those links originally having negative weights from the MWIS and use the remaining set as our solution. It is not difficult to see that such a method indeed gives the MWIS to the original problem with negative weights. The problem that all links have negative weights is a trivial problem, where the solution is just the link with the largest weight. The proof of Theorem 2 in the below will show that the MWIS problem of importance in this paper indeed involves non-negative weights, so we will not have a trivial issue.

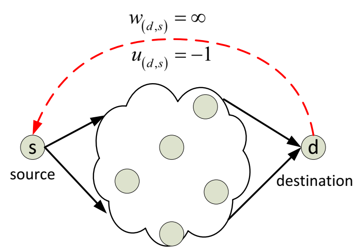


Figure 1: Reformulation of the maximum flow problem as a standard network flow problem.

4.4 DCG in the Maximum Flow Problem

The maximum flow problem can be reformulated as a standard network flow problem as follows. We let the cost of every arc be equal to zero and introduce a new infinite capacity arc (d, s) with cost $u_{(d,s)} = -1$, as shown in Fig. 1. Minimizing $\mathbf{c}'\mathbf{r}$ in the new network is the same as maximizing the flow $x_{(d,s)}$ on the new arc. Since the flow on the arc (d, s) must return from s to d through the original network, maximizing $x_{(d,s)}$ is the same as solving the original maximum flow problem.

In the transformed network, the problem $\mathcal{P}_\mathbf{2}$ needs to be updated as follows. The decision variables are now $\mathbf{x} = (r_{(d,s)}, \mathbf{r}', \boldsymbol{\alpha}')$, and the cost vector $\mathbf{c} = (-1, \mathbf{0}_{1\times(|\mathcal{L}|+M)})$. The truncated node-arc incidence matrix $\mathbf{H}$ needs to add one more column (specifically, as the first column) to include the new arc (d, s). The scheduling constraints do not need to be changed. The arc (d, s) is a virtual arc for modeling purpose. It does not interfere with any wireless link. Also the virtual link has an infinite capacity, so does not need an independent set based scheduling. Also, the external input rate is 0 in the reformulated problem, i.e., $\mathbf{g} = \mathbf{0}$. Thus, the problem $\mathcal{P}_\mathbf{2}$ is to be updated into the problem $\mathcal{P}_\mathbf{3}$, which has

$$\mathbf{A} = \left[\begin{array}{c|c} \mathbf{H}_{(|\mathcal{N}|-1)\times(|\mathcal{L}|+1)} & \mathbf{0}_{(|\mathcal{N}|-1)\times M} \\ \hline \hat{\mathbf{I}}_{|\mathcal{L}|\times(|\mathcal{L}|+1)} & -\mathbf{W}_{|\mathcal{L}|\times M} \\ \hline \mathbf{0}_{1\times(|\mathcal{L}|+1)} & \mathbf{1}_{1\times M} \end{array}\right] \tag{27}$$

$$\mathbf{b} = (\mathbf{0}'_{1\times(|\mathcal{N}|-1+|\mathcal{L}|)}, 1) \tag{28}$$

where

$$\hat{\mathbf{I}}_{|\mathcal{L}|\times(|\mathcal{L}|+1)} = \left[\begin{array}{c|c} \mathbf{0}_{|\mathcal{L}|\times 1} & \mathbf{I}_{|\mathcal{L}|\times|\mathcal{L}|} \end{array}\right]. \tag{29}$$

Theorem 2. *If we apply an approximation algorithm with the approximation ratio β to solve the MWIS problem when search for an entering column, the maximum flow solved under such an approximation algorithm will be at least β of the optimal solution.*

Proof. Note that we apply the delayed column generation method to the maximum flow problem formulated above. Given a basic feasible solution $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$, we then have $\mathbf{p}' = \mathbf{c}'_B\mathbf{B}^{-1}$. Further, $\mathbf{p}'\mathbf{b} = \mathbf{c}'_B\mathbf{B}^{-1}\mathbf{b} = \mathbf{c}'_B\mathbf{x}_B = p_{|\mathcal{N}|+|\mathcal{L}|} \le 0$, where the last step is due to $\mathbf{c} = (-1, \mathbf{0}_{1\times(|\mathcal{L}|+M)})'$ and a feasible solution $\mathbf{x} \ge \mathbf{0}$.

To find a new column with improved solutions is to find a column

$$c_i - \mathbf{p}'\mathbf{A}_i \le 0$$

With delayed column generation, the candidate columns are only in the right sub-matrix of $\mathbf{A}$. For those columns, the associated cost variables are all 0. To pick the one with the largest value of cost reduction, the problem is to

$$\text{minimize} \quad -\mathbf{p}'\mathbf{A}_j.$$

It is equivalent to

$$\text{minimize} \quad \sum_{i=1}^{|\mathcal{L}|} p_{k_i}\mathbf{w}_j(i) - p_{|\mathcal{N}|+|\mathcal{L}|}$$

and equivalent to

$$\text{maximize} \quad \sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i).$$

When the algorithm stops, we have

$$\mathbf{c}' - \mathbf{p}'\mathbf{A} \geq \mathbf{0}',$$

where $-p_{|\mathcal{N}|+|\mathcal{L}|}$ equals the network flow obtained. That is,

$$\sum_{i=1}^{|\mathcal{L}|} p_{k_i}\mathbf{w}_j(i) - p_{|\mathcal{N}|+|\mathcal{L}|} \geq 0 \quad \forall \text{ independent set}$$

equivalently,

$$\sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i) \leq -p_{|\mathcal{N}|+|\mathcal{L}|} \quad \forall \text{ independent set} \tag{30}$$

Thus,

$$\max_{j\in\mathcal{M}}\left(\sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i)\right) \leq -p_{|\mathcal{N}|+|\mathcal{L}|} \tag{31}$$

When the MWIS problem is solved by an approximation algorithm with an approximation ratio β (< 1), let $\tilde{\max}$ denote the maximum value obtained with the algorithm and we can have

$$\beta \max_{j\in\mathcal{M}}\left(\sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i)\right) \leq \tilde{\max_{j\in\mathcal{M}}}\left(\sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i)\right) \tag{32}$$

$$\leq -p_{|\mathcal{N}|+|\mathcal{L}|} \tag{33}$$

That is

$$\max_{j\in\mathcal{M}}\left(\sum_{i=1}^{|\mathcal{L}|}(-p_{k_i})\mathbf{w}_j(i)\right) \leq -\frac{1}{\beta}p_{|\mathcal{N}|+|\mathcal{L}|} \tag{34}$$

The dual problem of $\mathcal{P}_\mathbf{3}$ is

$$\begin{aligned} &\text{maximize} && \mathbf{p}'\mathbf{b} \\ &\text{subject to} && \mathbf{p}'\mathbf{A} \leq \mathbf{c}'. \end{aligned}$$

We now show that the vector $\hat{\mathbf{p}} = (p_1, \cdots, p_{|\mathcal{N}|+|\mathcal{L}|-1}, \frac{1}{\beta}p_{|\mathcal{N}|+|\mathcal{L}|})$ is a feasible solution to the dual problem. The vector $\mathbf{p}$ is associated with the optimal solution of the restricted problem, where all the columns involved are represented as set $\mathcal{Q}$. For a column $i \in \mathcal{Q}$ and associated with the basic variables, we have $c_i - \mathbf{p}'\mathbf{A}_i = 0$; for a column $i \in \mathcal{Q}$ and associated with the non-basic variables, we have $c_i - \mathbf{p}'\mathbf{A}_i \geq 0$ (if not, the solution associated with $\mathbf{p}$ could not be the optimal of the restricted problem). Given a vector $\mathbf{p}$ that stops the DCG algorithm, we can see that replacing the last element $p_{|\mathcal{N}|+|\mathcal{L}|}$ with $\frac{1}{\beta}p_{|\mathcal{N}|+|\mathcal{L}|}$ will not change the value of $\mathbf{p}'\mathbf{A}_i$ for left-submatrix columns, since $\mathbf{A}_i(|\mathcal{N}| + |\mathcal{L}|) = 0$ for those columns. Further the result in (34) ensures that $c_i - \hat{\mathbf{p}}'\mathbf{A}_i \geq 0$ for all the right-submatrix columns. In summary, $c_i - \hat{\mathbf{p}}'\mathbf{A}_i \geq 0$ for all columns of $\mathbf{A}$ and $\hat{\mathbf{p}}$ is a dual feasible solution. Then, by the weak duality, we have

$$\hat{\mathbf{p}}'\mathbf{b} = \frac{1}{\beta}p_{|\mathcal{N}|+|\mathcal{L}|} \leq \mathbf{c}'\mathbf{x}^* \tag{35}$$

That is

$$\text{max flow} = -\mathbf{c}'\mathbf{x}^* \leq -\frac{1}{\beta}p_{|\mathcal{N}|+|\mathcal{L}|} \tag{36}$$

The approximate solution

$$-p_{|\mathcal{N}|+|\mathcal{L}|} \geq \beta \times (\text{max flow}) \tag{37}$$

□

4.5 DCG in the MCF Problem

Consider there are F commodity flows over the network, with source-destination pairs (s_f, d_f) for $f = 1, \cdots, F$, respectively. The multi-commodity flow (MCF) problem is to maximize the summation of all the commodity flows. We can also transform the MCF problem to a standard network flow problem. Specifically, we induce new infinite capacity arcs (d_f, s_f) for $f = 1, \cdots, F$. We set the link costs $u_{(d_f,s_f)} = -1$ for $f = 1, \cdots, F$ and a cost of zero for all the other links.

We now need to determine the link flow allocation for every commodity. Thus we define for commodity flow f the vector

$$\mathbf{x}_f = (r_{(d_f,s_f)}, \mathbf{r}'_f) \tag{38}$$

where $\mathbf{r}_f$ indicates the flow allocation vector for commodity f over the original network, i.e., $\mathbf{r}_f$ is the vector of commodity f flow allocation $r^f_{(a,b)}$ for all links $(a, b) \in \mathcal{L}$. The decision variables are

$$\mathbf{x} = (\mathbf{x}'_1, \cdots, \mathbf{x}'_F, \boldsymbol{\alpha}'). \tag{39}$$

Define the cost vector $\mathbf{c}_f = (-1, \mathbf{0}_{1\times|\mathcal{L}|})$. The cost vector for the transformed network flow problem will be $\mathbf{c} = (\mathbf{c}'_1, \mathbf{c}'_2, \cdots, \mathbf{c}'_F, \mathbf{0}_{1\times M})$.

Each commodity flow allocation $\mathbf{r}_f$ needs to satisfy the flow conservation constraint. For flow f, the truncated node-arc incidence matrix $\mathbf{H}$ needs to add one column (specifically, as the first column) to include the newly induced arc (d_f, s_f); we use $\mathbf{H}^f$ to indicate this updated matrix. Since the induced links are not confined by the scheduling constraint, the matrix $\hat{\mathbf{I}}$ defined in (29) applies to every commodity flow.

Now the problem $\mathcal{P}_\mathbf{2}$ is to be updated into the problem $\mathcal{P}_\mathbf{4}$, which has

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}^1 & & & & \\ & \mathbf{H}^2 & & & \\ & & \ddots & & \\ & & & \mathbf{H}^F & \\ \hat{\mathbf{I}} & \hat{\mathbf{I}} & \cdots & \hat{\mathbf{I}} & -\mathbf{W}_{|\mathcal{L}|\times M} \\ & & & & \mathbf{1}_{1\times M} \end{bmatrix}$$

$$\mathbf{b} = (\mathbf{0}'_{1\times(F(|\mathcal{N}|-1)+|\mathcal{L}|)}, 1)$$

We define the first $(|\mathcal{L}| + 1)F$ columns of matrix $\mathbf{A}$ as left sub-matrix and the leftover M columns as right sub-matrix. It is not difficult to see that the DCG algorithm presented in Section 4.3 and the theoretical analysis regarding the approximated MWIS algorithm in Section 4.4 apply to the problem $\mathcal{P}_\mathbf{4}$ too. We thus have the following theorem.

THEOREM 3. *If we apply an approximation algorithm with the approximation ratio β to solve the MWIS problem when search for an entering column, the MCF solved under such an approximation algorithm will be at least β of the optimal solution.*

4.6 DCG in MCF with Fairness Constraint

Ensuring fairness in resource allocation is an important issue. According to [1], an MCF problem with fairness constraint can be formulated as maximizing a factor λ (≤ 1) where at least $\lambda \cdot g_{s_f}$ amount of throughput can be guaranteed for each commodity flow. Here, g_{s_f} is the traffic requirement of commodity flow f. We also transform the problem to a standard network flow problem similarly to the construction in Section 4.5. We induce new infinite capacity arcs (d_f, s_f) and set the link costs $u_{(d_f,s_f)} = -1$ for $f = 1, \cdots, F$, and set a cost of zero for all the other links. With the fairness constraint, we should have

$$r_{(d_f,s_f)} = \lambda g_{s_f} \quad \text{for } f = 1, \cdots, F. \tag{40}$$

We still adopt the vectors $\mathbf{x}_f$ and $\mathbf{x}$ defined in (38) and (39). Let $\mathbf{g}_F = (g_{s_1}, \cdots, g_{s_F})$ denote the vector of traffic requirements of all commodities. Define a $(|\mathcal{L}| + 1) \times 1$ vector $\mathbf{e} = (1, 0, \cdots, 0)$. The constraints in (40) can be expressed in a matrix format as

$$\begin{bmatrix} \mathbf{e}' & & & \\ & \mathbf{e}' & & \\ & & \ddots & \\ & & & \mathbf{e}' \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_F \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{E}_1 & \mathbf{E}_2 & \cdots & \mathbf{E}_F \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_F \end{bmatrix} = \lambda \mathbf{g}_F, \tag{41}$$

where each sub-matrix $\mathbf{E}_f$ is a $F \times (|\mathcal{L}| + 1)$ matrix, with only one non-zero element $\mathbf{E}_f(f, 1) = 1$.

With λ included, the new decision variable is defined as

$$\tilde{\mathbf{x}} = (\lambda, \mathbf{x}'_1, \cdots, \mathbf{x}'_F, \boldsymbol{\alpha}'), \tag{42}$$

with the corresponding cost vector $\mathbf{c} = (-1, 0, \cdots, 0)$. The problem $\mathcal{P}_\mathbf{2}$ can then be updated to problem $\mathcal{P}_\mathbf{5}$ with

$$\mathbf{A} = \begin{bmatrix} -\mathbf{g}_F & \mathbf{E}_1 & \mathbf{E}_2 & \cdots & \mathbf{E}_F & \\ & \mathbf{H}^1 & & & & \\ & & \mathbf{H}^2 & & & \\ & & & \ddots & & \\ & & & & \mathbf{H}^F & \\ & \hat{\mathbf{I}} & \hat{\mathbf{I}} & \cdots & \hat{\mathbf{I}} & -\mathbf{W}_{|\mathcal{L}|\times M} \\ & & & & & \mathbf{1}_{1\times M} \end{bmatrix}$$

$$\mathbf{b} = (\mathbf{0}'_{((|\mathcal{N}|-1)(F+1)+|\mathcal{L}|)\times 1}, 1).$$

Again, the DCG algorithm and the theoretical analysis regarding the approximated MWIS algorithm presented in previous cases apply to the problem $\mathcal{P}_\mathbf{5}$ too. We thus have the following theorem.

THEOREM 4. *If we apply an approximation algorithm with the approximation ratio β to solve the MWIS problem when search for an entering column, the MCF problem with fairness constraint solved under such an approximation algorithm will be at least β of the optimal solution.*

5. DCG IN MR-MC NETWORKS

The DCG method can be applied to MR-MC networks based on the multi-dimensional conflict graph technique developed in [9]. To facilitate understanding, we here briefly summarize the MDCG technique presented in [9]. We use C to denote the number of available channels and c a specific channel. Although c is used as a link cost in previous sections, there will be no confusions of notation due to the clear context in this section. We use M_a to denote the number of radios available at node a.

An MR-MC network can be interpreted as a multi-dimensional resource space, with *dimensions* defined by links, radios, and channels. The MDCG is to describe the conflict relationships among the resource points, each represented as a link-radio-channel tuple. Specifically, an LRC tuple p is defined in the format:

$$\text{Link-radio-channel tuple:} \quad ((a, b), (y_a, y_b), c). \tag{43}$$

The tuple indicates that link (a, b) operates on channel c, using radios y_a and y_b at nodes a and b, respectively. Note that link (a, b) can be mapped to $M_a \times M_b \times C$ LRC tuples in the MDCG.

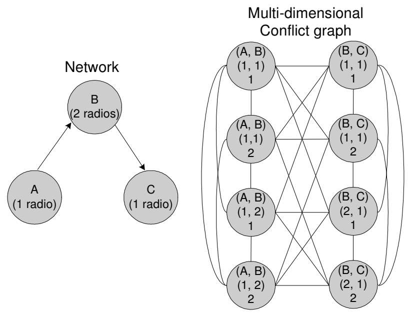


Figure 2: An illustration of MDCG construction [9].

There are two types of conflict relationships among LRC tuples in the MDCG. One is *interference conflict* indicating that co-channel transmissions (geometrically separated) conflict with each other within the interference range. The other is *radio conflict* indicating that multiple transmissions (possibly over different channels) contend for the same radio.

Consider an example of the MDCG. The left side of Fig. 2 shows a small network with directed links and 2 available channels. The right side illustrates the MDCG constructed. With the MDCG, the IS-based scheduling could jointly give the solutions of link scheduling, routing, channel and radio assignment [9].

For a network flow formulation over an MR-MC network, the flow conservation constraint presented in Section 4.1 still applies. The scheduling constraint presented in Section 4.2 needs to be updated according to the MDCG. Let $\mathcal{I}^p_m$ denote the mth independent set over an MDCG and $\mathcal{M}^p$ the set of all of the independent sets over the MDCG. The scheduling constraint is updated as:

$$r_{(a,b)} \le \sum_{m:(a,b)\in\mathcal{I}^p_m} \alpha_m w^{c((a,b),\mathcal{I}^p_m)}_{(a,b)} \tag{44}$$

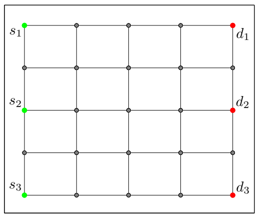
$$\sum_{m=1}^{|\mathcal{M}^p|} \alpha_m = 1 \tag{45}$$

where $c((a, b), \mathcal{I}^p_m)$ denotes the operating channel of link (a, b) when it is activated in the IS $\mathcal{I}^p_m$, and $w^c_{(a,b)}$ denotes the channel-dependent capacity of link (a, b) in the multiple-channel context. Note that the scheduling constraint (44) can conveniently exploit the channel diversity to optimally utilize high-quality channels for the maximum network capacity [9].

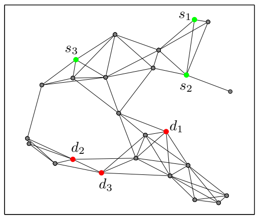
With the updated scheduling constraints (44) and (45), it can be easily checked that all the analysis and theorems developed in Section 4 remain valid in the MR-MC context. An implementation detail is that each column of the matrix $\mathbf{W}$ now represents a tuple-link based IS over the MDCG. In the implementation, we assume all nodes have the same communication range and interference range over all channels.

6. NUMERICAL RESULTS

In this section, we present numerical results to demonstrate the performance of DCG. For the convenience of comparison with existing methods, we focus on the MCF problem with fairness constraint in an MR-MC network which is considered in [1] and [9]. Our goals are three-fold. 1) We study the efficiency of the DCG algorithm by comparing its performance with three existing algorithms — the joint routing, channel assignment and link scheduling (RCL) algorithm proposed in [1], the random search of ISs [4] over the MDCG (RS-MDCG), and the MDCG based scheduling index ordering (SIO-MDCG) algorithm proposed in [9]. The performance is measured in terms of achieved network capacity λ^*



(a) Grid topology



(b) Random topology

Figure 3: Network topologies.

and the total computation time. 2) We then study the scalability of our DCG algorithm versus the number of available channels, the number of radios and the network size. 3) We compare the performance of three MWIS approximation algorithms (i.e., the greedy algorithm in [16], the LP-based algorithm that integrates the greedy algorithm with a relaxed linear programming in [16], and the SRA algorithm in [17]) in affecting the performance of the DCG algorithm in Section 6.3.

In the following, all our results with the DCG algorithm are based on the greedy MWIS algorithm [16], except in Section 6.3 where we explicitly compare the performance of different approximate MWIS algorithms. The DCG algorithm is terminated if the following two events happen: 1) no new entering column is found by the MWIS algorithm; 2) the network capacity improvement in one iteration has held less than 10^{-6} for 100 iterations. The second condition stems from our observation that, after a number of iterations, the algorithm begins continuously finding new entering columns which introduce very limited reduced cost (or have little impact on the capacity). We set the RS-MDCG algorithm stop searching for new MISs when it has already found 2×10^5 MISs.

We develop Matlab programs for each of the above algorithms. We use CPLEX [15] within the Matlab environment to solve all the LPs involved in the above algorithms. In the comparisons, the computation time of each algorithm is measured as the total time including independent set searching (not applicable to the RCL algorithm) and LP optimizations. We run the program on a machine with an 8-core and 3.4GHz CPU and 12GB RAM. Besides, the codes are optimized to further avoid memory leak when solving large-scale LPs. For instance, instead of storing the sparse independent set matrix $\mathbf{W}$, which is very memory expensive, we only store the non-zero elements along with their associated positions in the matrix. Intermediate variables are immediately cleared after last use in order to save memory.

6.1 Performance Comparison

We consider two networks as shown in Fig. 3, each of which has 25 nodes deployed within a 900m × 900m area. The communication range and interference range of each node are 250m and 500m, respectively. In the grid topology, each node is equipped with 4 radios and there are totally 8 orthogonal channels available to them; in the random topology, 9 orthogonal channels are available and each node has 3 radios. We assume that the channel capacities of all the tuples are the same and are normalized to 1 rate unit. As marked in both figures, there are 3 commodity flows from sources s_1, s_2, s_3 to destinations d_1, d_2, d_3, respectively. Each flow demands a throughput of 3 rate units, i.e., the input to the source nodes $g_i = 3$. As a reference value, an upper bound of λ^* is also calculated by solving an LP relaxation problem proposed in [1]. Note that this upper bound may not be practically feasible. For the RS-MDCG algorithm, in order to achieve comparable capacity, we fix the total number of independent sets to be searched at 2×10^5.

Table 1: Performance comparisons.

Network 1 with 8 channels and each node having 4 radios			
	λ^*	computation time (seconds)	number of independent sets used
Upper bound	1.0	—	—
RCL	0.179	10.6	—
RS-MDCG	0.345	41771.0	2×10^5
SIO-MDCG	0.404	3485.28	31406
DCG	0.435	138.1	1008 (223 iterations)
Network 2 with 9 channels and each node having 3 radios			
	λ^*	computation time (seconds)	number of independent sets used
Upper bound	0.333	—	—
RCL	0.125	23.68	—
RS-MDCG	0.230	46313.22	2×10^5
SIO-MDCG	0.191	2620.25	28718
DCG	0.300	117.32	1118 (285 iterations)

Table 1 summarizes the comparison results. Generally, our algorithm achieves the most preferable tradeoff between network capacity and computation time for both topologies. First, we observe that the DCG algorithm achieves the best capacities with the least computation time among all the three MDCG based algorithms. In fact, both RS-MDCG and SIO-MDCG first find a large amount of maximal independent sets, and then input them into the LP problem $\mathcal{P}_5$ which solves the optimal capacity and scheduling jointly. During searching for independent sets, RS-MDCG just applies random search while the SIO-MDCG does so in a more managed way by taking critical independent sets into account [9]. However, the importance of each independent set to the problem has not been fully explored. Alternatively, the DCG algorithm searches for a new entering independent set in each round which can definitely improve the objective. As the table shows, the DCG algorithm uses only around 1000 ISs (among them 785 and 833 sets are initial ISs) in both networks, respectively. These numbers are significantly smaller than those used by the RS-MDCG and SIO-MDCG algorithms. Second, although the RCL algorithm takes the shortest time, its final capacity is significantly low (less than 50% of the capacity achieved by the DCG algorithm).

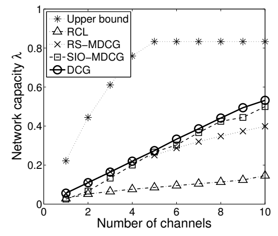
6.2 Scalability Evaluation

6.2.1 *The impact of number of channels*

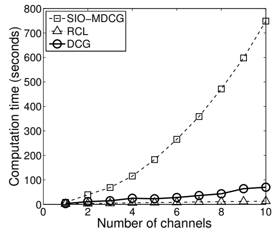
Consider both networks shown in Fig. 3 with the number of available channels ranging from 1 to 10. From the results shown in Fig. 4, we have the following observations. 1) For all the four algorithms, the network capacity basically increases proportionally with the number of available channels. 2) The DCG algorithm always achieves the highest capacity. Moreover, as shown in Fig. 4(b) and 4(d), the computation time of DCG is significantly lower than that of RS-MDCG (its computation time is longer than 6000 seconds and thus omitted from the figures) and of SIO-MDCG. Although the RCL algorithm takes the shortest time, its achieved capacity is significantly lower than all the other three algorithms. 3) The computation time of the SIO-MDCG algorithm is exponential in the number of channels. However, roughly speaking, both the DCG and RCL algorithms run in time polynomial in the number of channels. Such polynomial relationship for the RCL algorithm has been proved in [1]. For the DCG algorithm, we further explore it in Section 6.2.3.

6.2.2 *The impact of number of radios*

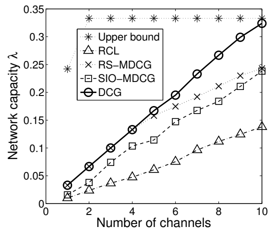
Based on the observations above, only the SIO-MDCG algorithm is comparable to our SIO-MDCG algorithm in terms of both



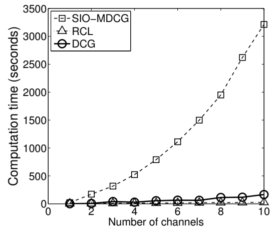
(a) Network capacity λ.



(b) Computation time.

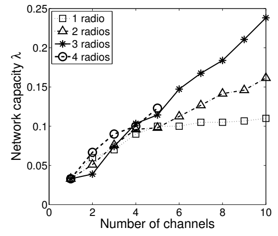


(c) Network capacity λ.

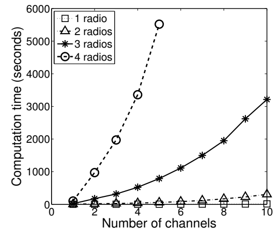


(d) Computation time.

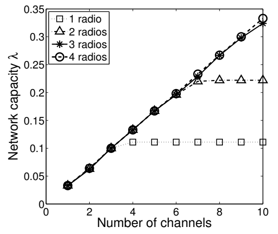
Figure 4: Impact of multi-channels. (a-b) Grid topology with 4 radios; (c-d) random topology with 3 radios.



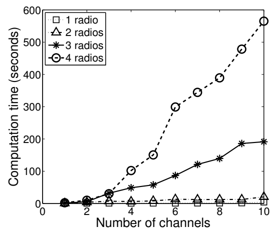
(a) Network capacity λ.



(b) Computation time.



(c) Network capacity λ.



(d) Computation time.

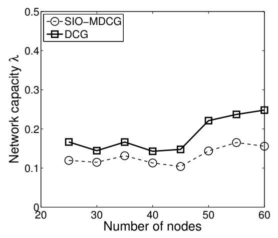
Figure 5: Impact of multi-radios. (a-b): SIO-MDCG algorithm; (c-d): DCG algorithm.

network capacity and computation complexity. Therefore, we only compare the performance of the DCG and SIO-MDCG algorithms in the following. We consider the random topology with the number of radios varying from 1 to 4. The results for the grid topology are similar and hence are omitted. In addition, in Fig. 5(a) and 5(b), the results for SIO-MDCG with 4 radios and more than 5 channels are omitted because the algorithm's computation time is excessively high while the resultant network capacities are very close to the curve corresponding to 3 radios.

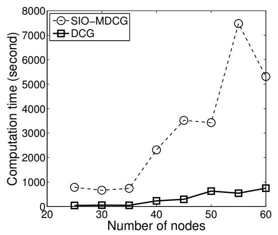
In Fig. 5, we can observe for both algorithms that when there are less than 4 channels available, the number of radios does not have significant impact on the network capacity; in other words, all the four channels can be exploited even with just one radio. The explanation is that in that range, the optimal resource allocation for the highest network capacity is to assign the links within an interference neighborhood each a different channel [20]. When there are still more channels available than the number of links in an interfering neighborhood, which can then be exploited by the extra available channels. In Fig. 5, there are also ranges that the capacity curves become flat, where all the available radios are fully used and can not exploit the extra available channels. Furthermore, the exponential growth of the computation time of the MIO-MDCG algorithm is clearly shown in Fig. 5(b). In contrast, the computation complexity of the DCG algorithm is polynomial in the number of radios as shown in Fig. 5(d).

6.2.3 The impact of network size

In this evaluation, we generate a number of networks with nodes randomly deployed within linearly expanded areas, where the node density maintains the same as that of the random network shown in Fig. 3(b). There are 5 channels available and each node has 3 radios. We consider 3 commodity flows where their sources and destinations are randomly chosen in each of the networks. From the results shown in Fig. 6(a), the network capacities by the two algorithms are almost flat versus the network sizes, but the DCG always performs better than the SIO-MDCG. Such flat shapes are reasonable since the network density is fixed. As the network scale grows, the computation complexity of the SIO-MDCG algorithm climbs much faster than that of the DCG algorithm, as shown in Fig. 6(b).



(a) Network capacity λ.



(b) Computation time.

Figure 6: Scalability against network size. Random topologies with 3 radios and 5 channels.

Since the tuple-based independent sets have been used in the MR-MC networks, we here more directly demonstrate the algorithm complexity versus the number of tuples. We collect all the data for the DCG algorithm in Fig. 4(b) and 5(d) and re-present them in Fig. 7 as a function of the square of the number of tuples. We can see that the computation time is almost linear to the square of tuple numbers. In fact, fixing the network topology, the number of tuples is proportional to the product of the number of channels (C) and the square of the number of radios ($(\max\{M_a\})^2$). Therefore, these results show that our DCG algorithm is polynomial in time, and the complexity is in the order of $O\left(C^2(\max\{M_a\})^4\right)$.

6.3 Comparison of MWIS Algorithms

With thorough examinations, we find that the major amount of the computation time in DCG is spent by the embedded MWIS algorithm. Therefore, in this part, we study the impact of the MWIS algorithm on DCG by comparing the performance with three MWIS approximation algorithms: the greedy algorithm [16], the LP-based algorithm [16], and the support ratio algorithm (SRA) [17]. We use the random network as shown in Fig. 3(b) with 9 available channels and 3 radios. From Fig. 8, we observe that the simple greedy algorithm converges extremely faster than the other two without sacrificing the network capacity. This is reasonable since both SRA and LP-based algorithms have higher computation complexity than the greedy algorithm. Another observation is that by the three algorithms, the network capacity curves all experience a sharp increasing period.

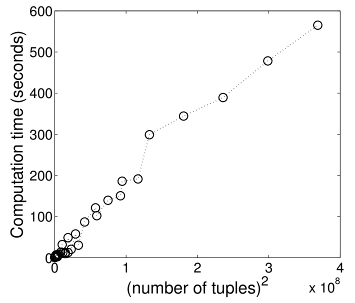


Figure 7: Computation time as a function of square of tuples number.

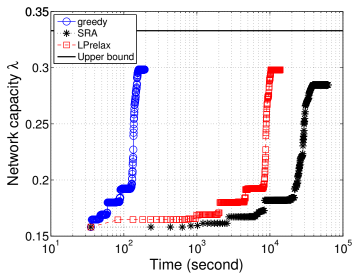


Figure 8: Performance comparison among MWIS algorithms.

7. CONCLUSION

The network capacity of a multi-hop wireless network is fundamentally determined by independent set based scheduling. Searching all ISs over a conflict graph for scheduling is NP-hard in general. In this paper, we systematically study the performance of the delayed column generation method in solving network flow problems augmented with IS-based scheduling, and contribute some original theoretical analysis of the performance of DCG. Extensive numerical results have been presented to demonstrate that the DCG algorithm achieves the most preferred tradeoff between network capacity and computation time among the RCL, RS-MDCG and SIO-MDCG algorithms, and is also scalable versus the number of channels, the number of radios, and the network size. In addition, as for the embedded MWIS algorithm in DCG, it has been shown that the greedy approximation algorithm is a good option for both the performance and the complexity.

8. ACKNOWLEDGMENTS

This work was supported in part by the NSF under grants CNS-1053777 and CNS-1320736.

9. REFERENCES

[1] M. Alicherry, R. Bhatia, and L. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proc. ACM MobiCom*, pp. 58–72, Aug. 2005.

[2] W. Wang, X. Liu, and D. Krishnaswamy, "Robust routing and scheduling in wireless mesh networks", in *Proc. IEEE SECON*, pp. 471–480, 2007

[3] L. Badia, A. Erta, L. Lenzini, and M. Zorzi, "A general interferenceaware framework for joint routing and link scheduling in wireless mesh networks", *IEEE Network*, vol. 22, no. 1, pp. 32–38, Jan./Feb. 2008

[4] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multihop wireless network performance", in *Proc. ACM MobiCom*, pp. 66–80, 2003.

[5] W. Li, X. Cheng, T. Jing, Y. Cui, K. Xing, and W. Wang, "Spectrum assignment and sharing for delay minimization in multi-hop multi-flow CRNs, " *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2483–2493, Nov. 2013.

[6] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multiradio wireless mesh networks", in *Proc. IEEE INFOCOM*, pp. 1–12, 2006.

[7] J. Tang, S. Misra and G. Xue, "Joint spectrum allocation and scheduling for fair spectrum sharing in cognitive radio wireless networks," *Computer Networks*, vol. 52, no. 11, pp. 2148-2158. 2008.

[8] H. Li, Y. Cheng, X. Tian, and X. Wang, "A generic framework for throughput-optimal control in MR-MC wireless networks," in *Proc. IEEE INFOCOM*, Orlando, Florida, Mar. 25–30, 2012

[9] H. Li, Y. Cheng, C. Zhou, and P. Wan, "Multi-dimensional conflict graph based computing for optimal capacity in MR-MC wireless networks", in *Proc. IEEE ICDCS*, Genoa, Italy, June 21–25, 2010.

[10] I. Ho, P. Lam, P. Chong, and S. Liew, "Harnessing the High Bandwidth of Multi-radio Multi-channel 802.11 n Mesh Networks," *IEEE Trans. Mobile Computing*, vol. 13, no. 2, pp. 448–456, Feb. 2014.

[11] D. Bertsimas and J. N. Tsitsiklis. "Introduction to linear optimization", *Athena Scientific*, 1997.

[12] P. Bjorklund, P. Varbrand, and D. Yuan. "Resource optimization of spatial TDMA in ad hoc radio networks: A column generation approach". In *Proc. IEEE INFOCOM*, 2003.

[13] S. Kompella, J. E. Wieselthier, and A. Ephremides. "Multi-hop routing and scheduling in wireless networks subject to SINR constraints.", *46th IEEE Conference on Decision and Control*, 2007.

[14] J. Luo, C. Rosenberg, A. Girard, "Engineering wireless mesh networks: joint scheduling, routing, power control, and rate adaptation," *IEEE/ACM Trans. Netw.*, vol. 18, no. 5, pp. 1387–1400, Oct. 2010.

[15] IBM ILOG CPLEX Optimizer. `http://www.ilog.com/products/cplex/`.

[16] A. Kako, T. Ono, T. Hirata, and M. M. Halldórsson. "Approximation algorithms for the weighted independent set problem in sparse graphs." *Discrete Applied Mathematics*, vol. 157, no. 4, pp. 617–626, 2009.

[17] S. Balaji, V. Swaminathan, and K. Kannan. "Approximating Maximum Weighted Independent Set Using Vertex Support.", *World Academy of Science, Engineering and Technology*, 35, 2009.

[18] V. S. Anil Kumar, M. V. Marathe, and S. Parthasarathy, "Algorithmic aspects of capacity in wireless neworks," in *Proc. ACM SIGMETRICs*, pp. 133–144, 2005.

[19] X. Lin and S. Rasool, "Distributed and provably-efficient algorithms for joint channel-assignment, scheduling and routing in multi-channel ad hoc wireless networks," *IEEE/ACM Trans. Networking*, vol. 17, no. 6, pp. 1874–1887, Dec. 2009.

[20] P. Kyasanur and N. H. Vaidya, "Capacity of multi-channel wireless networks: Impact of number of channels and interfaces," in *Proc. ACM MobiCom*, Aug. 2005, pp. 43Ű57.

[21] P. Gupta and P. R. Kumar, "The cpacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.