

# Multi-Dimensional Conflict Graph Based Computing for Optimal Capacity in MR-MC Wireless Networks

Hongkun Li, Yu Cheng, Chi Zhou

Department of Electrical and Computer Engineering  
Illinois Institute of Technology  
{hli55, cheng, zhou}@iit.edu

Pengjun Wan

Department of Computer Science  
Illinois Institute of Technology  
wan@cs.iit.edu

**Abstract**—Optimal capacity analysis in multi-radio multi-channel wireless networks by nature incurs the formulation of a mixed integer programming, which is NP-hard in general. The current state of the art mainly resorts to heuristic algorithms to obtain an approximate solution. In this paper, we propose a novel concept of multi-dimensional conflict graph (MDCG). Based on MDCG, the capacity optimization issue can be accurately modeled as a linear programming (LP) multi-commodity flow (MCF) problem, augmented with maximal independent set (MIS) constraints. The MDCG-based solution will provide not only the maximum throughput or utility, but also the optimal configurations on routing, channel assignment, and scheduling. Moreover, the MDCG-based optimal capacity planning can exploit dynamic channel swapping, which is difficult to achieve for those existing heuristic algorithms. A particular challenge associated with the MDCG-based capacity analysis is to search exponentially many possible MISs. We theoretically show that in fact only a small set of critical MISs, termed as critical MIS set, will be scheduled in the optimal resource allocation. We then develop a polynomial computing method, based on a novel scheduling index ordering (SIO) concept, to search the critical MIS set. Extensive numerical results are presented to demonstrate the efficiency of the MDCG-based resource allocation compared to well-known heuristic algorithm presented in [1], and the efficiency of SIO-based MIS computing compared to the widely adopted random algorithm for searching MISs.

## I. INTRODUCTION

The multi-radio multi-channel (MR-MC) wireless network remains hot in the past few years. With MR-MC communications, the network capacity can be significantly improved by simultaneously exploiting multiple non-overlapping channels through different radio interfaces and mitigating interferences through proper network configuration.

Compared to the traditional single-radio single-channel (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 *channel assignment, scheduling, and routing* [1]–[8], with the objective to optimize network capacity; the resource allocation issue is also termed as *optimal capacity planning*. However, the optimal multi-dimensional resource allocation in MR-MC networks by nature leads to a mixed integer programming problem (which is NP-hard), involving binary variables to describe channel assignment and radio interface constraints [1], [2]. Due to such

inherent hardness, the state of the art of MR-MC networks has been constrained to either adopting linear programming (or convex optimization in general) relaxation to obtain an upper bound of the network capacity [1]–[4], or developing heuristic resource allocation methods [1], [2], [5]–[8] to obtain a lower bound.

In this paper, we aim to find an efficient computing methodology to optimize the capacity of MR-MC networks in the multi-dimensional resource space. Our inspiration comes from that the single-radio single-channel (SR-SC) networks can be optimized by formulating a linear programming (LP) multi-commodity flow (MCF) problem, augmented with constraints derived from the *link conflict graph* or *contention graph* [9], [10]. The conflict graph tool did not achieve the similar popularity in MR-MC networks as in SR-SC networks. The reason is that the link conflict graph is not sufficient to describe the complex conflict relations in competing for both the radio interfaces and channels. We show in this paper that by generally interpreting each vertex in the conflict graph as a basic resource point for scheduling, we could discover a generic methodology to extend the conflict graph tool to the MR-MC networks: representing each resource point in the MR-MC multi-dimensional space (i.e., each vertex in the conflict graph) as a *radio-link-channel tuple* (RLC-tuple). The conflict graph constructed in this manner is termed as *multi-dimensional conflict graph* (MDCG).

The new insight of conflict-graph based computing in a multi-dimensional resource space brings a chance to reshape the computing methodology in MR-MC networks. Based on the MDCG, the optimal resource allocation in MR-MC networks can now be transformed from the integer programming regime to the linear programming regime, and has a clear direction towards the optimal resource allocation, i.e., MCF formulation augmented with the MIS constraints [9]. The MDCG-based MCF solution will provide not only the maximum throughput or utility, but also the optimal channel assignment, scheduling and routing to achieve it. Moreover, the MDCG-based optimal capacity planning can exploit dynamic channel swapping, which is very hard to achieve for those existing heuristic algorithms. Numerical results are presented to demonstrate the efficiency of the MDCG-based capacity planning, with comparison to the well-known heuristic algorithm presented in [1].

A particular challenge associated with MDCG based capacity analysis is that finding all maximal independent sets in a conflict graph is NP-complete [12]. Thus it is impractical (although theoretically possible) to include all the MISs as augmented constraints in the MCF formulation. A random algorithm for MIS search is proposed in [9] and widely adopted in the literature, which provides a framework while more MISs can be obtained with more rounds of computation. However, we find that random search algorithm is quite inefficient for a large size MDCG (which is normally incurred by a MR-MC network with a large number of nodes, multiple channels and multiple radio interfaces), where the random search may result in redundant search (i.e., getting a MIS already found) with a high chance. In this paper, we theoretically show that in fact only a small set of critical MISs, termed as *critical MIS set*, will be scheduled in the optimal resource allocation, although exponentially many MISs are possible in a conflict graph. The important finding motivates us to develop an efficient algorithm to identify the critical MIS set, while the random algorithm could be interpreted as an aimless approach to cover the critical MIS set by throwing a huge number of MISs.

Another main contribution in this paper is that we develop a polynomial heuristic algorithm to intelligently compute a set of MISs to better cover the critical MIS set, compared to the aimless random search algorithm. Our basic idea is to exploit the network topology and network flow information to infer those communication links that might be scheduled with high possibility, and define a *scheduling index* metric to differentiate the scheduling priority of links. The tuple with a larger value of SI indicates the chance of scheduling with higher priority. The scheduling index ordering (SIO) allows us to develop a systemic way to approximately computing the critical MIS set by covering higher priority links (and the associated tuples) with more MISs; the algorithm is termed as *SIO-based MIS computing*. Numerical results show that the SIO-based MIS computing can well locate the critical MIS set, leading to higher network capacity with much less computing complexity. In the Section of numerical results, we will show that the SIO-based computing steadily performs with one order of reduced complexity fewer to achieve the same level of capacity compared to the random algorithm. In summary, this paper has four-fold main contributions:

- (1) We propose a novel multi-dimensional conflict graph, by which the optimal resource allocation in MR-MC networks can now be transformed from integer programming regime to the linear programming regime.
- (2) Based on MDCG, we formulate a MCF problem augmented with MIS constraint to compute the optimal capacity over a MR-MC network. The solution provides not only the maximum capacity, also the optimal routing, scheduling and channel assignment. Further, dynamic channel assignment is achieved as well.
- (3) We for the first time (to the best of our knowledge) reveal an important concept of critical MIS set, and develop an efficient polynomial SIO-based method to approximately

compute the critical MIS set.

- (4) Extensive numerical results are presented to demonstrate the efficiency of the MDCG-based capacity analysis and SIO-based MIS computing.

The reminder of this paper is organized as follows. Section II reviews more related work. Section III is the system model; Section IV will highlight the most popular methodology in this area. Section V presents how to construct the MDCG. Section VI formulates the MDCG-based MCF problem for optimal capacity planning. Section VII presents the heuristic algorithm. Section VIII shows numerical results for performance evaluation. Section IX gives the concluding remarks.

## II. RELATED WORK

The MCF formulation augmented with the maximal independent set (MIS) constraints over a conflict graph can compute the optimal capacity planning to maximize the throughput or utility [9]. The *optimal scheduling* policy is that all the maximal independent sets take turns in grabbing the channel for data transmission, with the proportion of transmission time for each set determined from the MCF solution. The links with positive flow allocation constitute the *optimal route set* for the given commodity flows. Note that the MIS-based optimal scheduling over the MDCG also determines the *optimal channel assignment and radio interface assignment*, with each element in an MIS standing for a RLC-tuple.

It is due to the lack of a generic conflict graph in the multi-dimensional resource space that the elegant MIS-based MCF formulation for optimal capacity planning could not be generalized to MR-MC networks. Tang *et.al.* [3] propose to establish a multi-channel contention graph (MCCG) by representing link-channel pairs as vertices in the conflict graph, where only co-channel link-channel pairs may have conflict relations. However, the MCCG could not indicate conflicts in competing for radio interfaces, so the MCF formulation still needs to incorporate integer variables to describe the radio interface constraint, even with MCCG applied. In [11] and [13], a multi-radio conflict graph (MCG) and a multi-point link conflict graph (MPLCG) are proposed, respectively, to construct the conflict graph based on radio-link pairs. Nevertheless, computing the channel assignment based on such conflict graphs incurs the *list coloring problem* [14], which is NP-hard. The generic MDCG developed in this paper is expected to significantly advance the state of the art of MR-MC networking.

In a conflict graph, computing all MISs is NP-complete. Random search [9] is a popular approach to gradually compute MISs by randomly index the vertices in the conflict graph at each iteration. However, the random search may lead to a large number of redundant search (giving the same MIS in different rounds of search) in a conflict graph with a large number of vertices and high connectivity, which is the case for the MDCG. The inefficiency in MIS directly leads to the unnecessary computing complexity in solving the MCF problem and lower capacity result. A systemic approach computing MIS is introduced by [22] with polynomial delay between two

outputs. It sorts all vertices in a lexicographic order. However, the output is not suitable to construct MIS constraints, because with fixed number of iterations, the vertices with smaller indexes may be included in most of the output MISs while the vertices with larger indexes may be covered by just a small portion of the output MISs. Such unbalanced output may significantly limit the solution region for searching optimal scheduling. We call such an issue as *over-concentrated problem*. The MIS computing algorithm developed in this paper also needs to index and order the vertices in the MDCG, but makes necessary designs to alleviate the impact of the over-concentrated problem.

### III. SYSTEM MODEL

The wireless network is viewed as a directed graph  $G(\mathcal{N}, \mathcal{L})$  with node set  $\mathcal{N}$  and link set  $\mathcal{L}$ , and  $|\mathcal{N}|$  and  $|\mathcal{L}|$  be number of nodes and links, correspondingly. We use  $l_{uv}$  or  $(u, v)$  to denote a communication link from node  $u$  to node  $v$ . Consider that the whole spectrum available to the network can be divided into  $C$  frequency channels, represented as set  $\mathcal{C}$ . We use  $w_{uv}^c$  to denote the physical transmission capacity of link  $(u, v)$  on channel  $c$ . Moreover, let  $M_u$  denote the number of radio interfaces available at node  $u$ , and  $K = \max M_u, \forall u \in \mathcal{N}$ . At any given time, an interface can only tune to one channel, but it can switch channels dynamically at different time slots.

The interference in the wireless network can be defined according to a *protocol interference model* or a *physical interference model* [15]. With the protocol interference model, the conflict relationship between two links is determined by the specified *interference range*. The protocol interference model is adopted by most of the existing work, by which the interference over a network can be abstracted into a conflict graph. We also focus on the protocol interference model in this research. A very recent paper [16] discusses how to properly set the interference range so that a physical interference model can be accurately transformed to a protocol interference model.

### IV. MIXED INTEGER PROGRAMMING MODEL

In this section, we summarize the methodology taken by the well-known work [1] to compute the the scheduling, channel assignment, and routing issues in MR-MC networks. The work [1] resorts to a mixed integer programming model to formulate the optimal resource allocation problem, assuming a synchronous time-slotted system. The significant contributions in [1] are the construction of node radio constraint and the link congestion constraint. To describe the node radio constraint, a binary integer variables  $X_{l,c,\tau}$  is introduced, where  $X_{l,c,\tau} = 1$  indicates link  $l$  operates on channel  $c$  in time slot  $\tau$ . Thus, the node radio constraint for each node  $v \in \mathcal{N}$  can be expressed as:

$$\sum_{1 \leq c \leq C} \sum_{l=(u,v) \in \mathcal{L}} X_{l,c,\tau} + \sum_{1 \leq c \leq C} \sum_{l=(v,u) \in \mathcal{L}} X_{l,c,\tau} \leq M_v. \quad (1)$$

which indicates that the number of active links incident to a node is constrained by the number of available radio interfaces. To derive an interference-free scheduling, the following

equation must be satisfied for each link  $l$ :

$$X_{l,\tau} + \sum_{l' \in I(l)} X_{l',\tau} \leq c(q). \quad (2)$$

where  $c(q)$  is a constant depending on the ratio of interference range to transmission range. The condition denotes that the number of active links within link  $l$ 's interference range can not exceed  $c(q)$ . With some other basic flow balance constraints, the two conditions are able to give us the optimal solution, however, we need to solve the optimal problem once for every time slot  $\tau$ , which is impractical.

Since the mixed integer programming is NP-hard, a LP relaxation of original problem is formulated to obtain an upper bound solution. Although the LP relaxation solution may generate infeasible channel assignment, it is "optimal" in terms of minimizing the interference for each channel. In this paper, we also use the LP relaxation results as computed in [1] as an upper bound of capacity analysis. To make the LP relaxation solution feasible, a 3-phase channel assignment algorithm is then developed to adjust the flow allocation on the flow graph to ensure a feasible channel assignment. The algorithm first allocates traffic to the different radio interfaces of a node to ensure that the traffic assigned to any node is at most minimum number of radios among all nodes; phase 2 assigns channels to nodes to derive small intra-component interference; phase 3 continues to modify the channel assignment for all nodes and radios so that the intra-channel interference is minimized. In addition, post processing and flow scaling are performed to redistribute the flow allocation to minimize the maximum interference over all channels, and eliminate all interference for all channels. *Note that the final channel assignment scheme is static, which significantly impact the resource utilization; in this paper, the MDCG based capacity analysis reveal that the optimal resource allocation by nature incurs dynamic channel swapping.*

### V. MULTI-DIMENSIONAL CONFLICT GRAPH

In this section, we will introduce new multi-dimensional conflict graph. The central issue of resource allocation in a MR-MC network is to find solutions for a set of coupled problems including channel assignment, scheduling, and routing [1]–[8], with the objective to optimize network capacity. In this paper, the optimal capacity planning of a MR-MC network is to be computed based on a multi-dimensional conflict graph. In the perspective of resource allocation, we interpret the MR-MC networks as a multi-dimensional resource space, with dimensions defined by radio interfaces, links, and channels. The MDCG is to described the conflict relationship among the resource points, each represented as a radio-link-channel tuple. Specifically, a RLC tuple  $p$  is defined in the format:

$$\text{Radio-link-channel tuple: } ((x_u, x_v), (u, v), c). \quad (3)$$

The tuple indicates that the link  $(u, v)$  operates on channel  $c$ , which involves the radio interfaces  $x_u \in M_u, x_v \in M_v$  at the sending node  $u$  and receiving node  $v$ , respectively. According to the RLC tuple definition, we can systematically list all the

possible resource allocations to enable a communication link by considering all the possible combinations of activated radio interfaces at source/destination nodes and the working channel. Thus, a link  $(u, v)$  can be mapped to  $M_u \times M_v \times C$  RLC tuples in the MDCG. Let  $\mathcal{P}$  denote the tuple set containing all tuples in a MDCG. Given the number of links  $|\mathcal{L}|$  in original graph, the number of tuples in a MDCG is at most  $K^2 C |\mathcal{L}|$ , in this case every node is equally equipped with  $K$  radios.

The conflict relationship among the RLC tuples in a multi-dimensional resource space is much more complex than that among the links in a single-dimensional resource space. On one side, the conflict relations within the interference range can be mitigated by parallel transmissions over different channels; on the other side, multi-channel simultaneous transmissions will be constrained by the available radio interfaces. To identify all the possible conflict relations in a MDCG, we define three types of events:

- (E) Two different RLC tuples are associated with nodes being located within each other's interference range, according to the protocol interference model.
- (F) Two different RLC tuples are associated with the same channel.
- (G) Two different RLC tuples share common radio interfaces at one or two nodes.

Give two different RLC tuples, a conflict relationship exist between them if the condition  $EF\bar{G} \cup G$  is true. In particular, the condition  $EF\bar{G}$  indicates that con-channel transmissions within the interference range conflict with each other, termed as *interference conflict* relationship. The condition  $G$  indicates the *interface conflict* relationship that a single radio interface can not support multiple transmissions (which might be associated with different channels) at a moment. Note that the interface conflict is the special issue induced by the MR-MC networking. There is no explicit interface conflict in a SR-SC network, because the constraint that a half-duplex radio interface can not transmit and receive simultaneously is ensured by the interference constraint. According to the decision criteria, we could figure out all the interfering neighbors for each tuple, and let  $I(p)$  denote these tuples. These tuples are called neighbors of  $p$ , and the rest are non-neighbors of  $p$ .

We use an example to illustrate the MDCG based on the conflict relationship among RLC tuples, as shown in Fig. 1. The left side of Fig. 1 shows a small network consisting of two directional links, where node  $A$  has two radio interfaces, and nodes  $B$  and  $C$  each have one interface. There are 2 available channels. Thus, both links  $(A, B)$  and  $(A, C)$  can be mapped to  $2 \times 1 \times 2 = 4$  RLC tuples, respectively. For instance, the tuple  $((1, 1), (A, B), 1)$  indicates that the transmission from node  $A$  to node  $B$  uses the radio interface 1 at node  $A$  and the one available radio interface at node  $B$ , and both radio interfaces tune to channel 1. Given the tuples, all possible interference/interface conflict relations among them can then be identified (according to the condition  $EF\bar{G} \cup G$ ) to form the MDCG, as shown in the right side of Fig. 1.

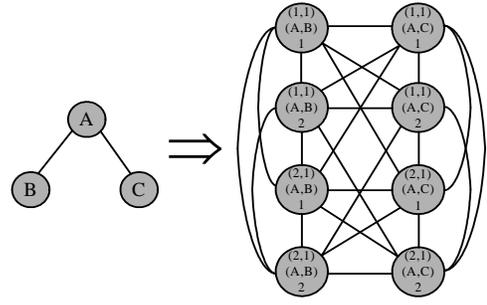


Fig. 1. Conflict relationship in MDCG.

## VI. MDCG BASED OPTIMAL CAPACITY PLANNING

The MDCG enables a linear programming approach to compute the optimal resource allocation in MR-MC networks, that is, formulating a MCF problem augmented with MIS constraints over the MDCG. In particular, it is selected sets of RLC tuples that constitute the MISs in MDCG; such MIS-based scheduling solved from the MCF formulation can jointly indicate the optimal scheduling of link transmissions, channel assignment and radio interface assignment to the active transmissions, and the optimal routing.

Define that each commodity is associated with a source destination pair  $(\nu, \eta)$ , and let  $\Lambda$  denote the set of all source/destination pairs,  $|\Lambda|$  the number of flows. Each flow has a rate requirement  $r_{(\nu, \eta)}$ . Let  $f_{(\nu, \eta)}(u, v)$  denote the flow associated with commodity  $(\nu, \eta)$  traversing the link  $(u, v)$ . The MCF problem can be formulated as a linear optimization or convex optimization problem, depending on how the objective function is selected. A common formulation is to maximize the total throughput over all source destination pairs, but such an objective function may lead to starvation of some commodity flows. Considering the fairness issue and the convenience for performance comparison with the resource allocation approach presented in [1], we seek to maximize  $\lambda$  where at least  $\lambda r_{(\nu, \eta)}$  amount of throughput can be ensured for each commodity flow. Such an objective function is also taken in [1]. We term  $\lambda$  as *network capacity*. Specifically, the basic MCF formulation can be expressed as:

$$\max \lambda \quad (4)$$

Subject to:

$$\sum_{l_{\nu v} \in \mathcal{L}} f_{(\nu, \eta)}(\nu, v) = \lambda r_{(\nu, \eta)}, \forall (\nu, \eta) \in \Lambda \quad (5)$$

$$\sum_{l_{uv} \in \mathcal{L}} f_{(\nu, \eta)}(u, v) = \sum_{l_{vu} \in \mathcal{L}} f_{(\nu, \eta)}(v, u), \quad \forall (\nu, \eta) \in \Lambda \text{ and } v \in \mathcal{N} / \{n_\nu, n_\eta\} \quad (6)$$

$$\sum_{l_{uv} \in \mathcal{L}} f_{(\nu, \eta)}(u, v) = 0, \forall (\nu, \eta) \in \Lambda \quad (7)$$

$$\sum_{l_{\eta v} \in \mathcal{L}} f_{(\nu, \eta)}(\eta, v) = 0, \forall (\nu, \eta) \in \Lambda \quad (8)$$

$$\sum_{(\nu, \eta) \in \Lambda} f_{(\nu, \eta)}(u, v) \leq B_{uv}, \forall l_{uv} \in \mathcal{L} \quad (9)$$

$$f_{(\nu, \eta)}(u, v) \geq 0, \forall l_{uv} \in \mathcal{L} \text{ and } (\nu, \eta) \in \Lambda \quad (10)$$

The constraints (5) to (10) respectively states: 1) The throughput achieved between each source/destination pair is proportional to the commodity rate requirement. 2) At every node, except the source and the destination, the amount of incoming flow associated with a certain ingress/egress pair equals the amount of outgoing flow for that ingress/egress pair; 3) For each ingress/egress pair, the incoming flow to the source node is 0; 4) The outgoing flow from the destination node is 0; 5) The total amount of flow on a link can not exceed the link throughput capacity; 6) The amount of flow allocation should be non-negative. We would emphasize that the link capacity  $B_{uv}$  is the average throughput capacity of link  $(u, v)$  under the scheduling. In the MR-MC network, a transmission link might operate on different channels at different time, and the physical link capacity might depend on the operating channel as  $w_{uv}^c$ . The relationship between the throughput capacity and the physical capacity is to be shown in the below.

In a wireless network, the basic MCF formulation needs to be augmented with conflict graph constraints to take the interferences among different links into account [9]. Given a conflict graph, an *independent set*  $\mathcal{I}$  is a set of vertices, such that there is no edge between any two of the vertices, which equivalently indicate the links (the RLC tuples in a MDCG) that can transmit simultaneously in the network. If adding any one more vertex into an independent set  $\mathcal{I}$  results in a non-independent set,  $\mathcal{I}$  is defined as a *maximal independent set*. Let  $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_K$  denote all the  $K$  maximal independent sets in the MDCG, and  $\alpha_m, 0 \leq \alpha_m \leq 1$  denote the fraction of time allocated to the independent set  $\mathcal{I}_m$ . Considering that only one maximal independent set can be active at a time, we can add the two constraints given in (11) and (12) to the basic MCF formulation to solve the optimal capacity planning of a MR-MC wireless network, that is,

$$\sum_{m=1}^K \alpha_m \leq 1 \quad (11)$$

$$B_{uv} = \sum_{m: l_{uv} \in \mathcal{I}_m} \alpha_m w_{uv}^{c(l_{uv}, \mathcal{I}_m)}, \forall l_{uv} \in \mathcal{L} \quad (12)$$

where  $c(l_{uv}, \mathcal{I}_m)$  denotes the operating channel of link  $(u, v)$  when it is activated in the MIS  $\mathcal{I}_m$ . The factors  $\alpha_m$  ( $m = 1, \dots, K$ ) solved from the MCF problem represents the optimal scheduling: assuming a large scheduling period of  $T$  slots, all the MISs take turns in accessing the channel for transmission, with MIS  $\mathcal{I}_m$  ( $m = 1, \dots, K$ ) allocated transmission time of  $\alpha_m T$  slots. Based on the scheduling factors, the average throughput capacity is determined by (12).

The MDCG based capacity planning has significant advantages in three aspects. 1) The MIS based scheduling can jointly indicate optimal transmission scheduling, channel assignment, radio interface assignment, and routing. Note that an active MIS at a slot consisting of RLC tuples that can operate simultaneously. The links associated with these RLC tuples represent transmissions scheduling. According to (3), each active RLC also indicates the radio interfaces allocated to the sender/receiver nodes of the link, and the operating channel.

Over the network, the links with positive flow allocation constitute the *optimal route set* for the given commodity flows. 2) The MIS based scheduling could exploit dynamic channel swapping for higher capacity [17]. A link is associated with multiple RLC tuples on different channels, and these tuples could be activated in different slots under the MIS-based scheduling, generating the effect of dynamic channel swapping on the transmission link. The benefit of dynamic channel swapping is to be demonstrated by numerical results in Section VII. 3) The MIS based scheduling can conveniently exploit the channel diversity to optimally utilize high-quality channels (with higher rate  $w_{uv}^c$ ) for the maximum network capacity, according to (12).

Since finding all MISs is NP-complete, this may lead to unacceptable complexity for constructing the constraint (11) (12). Further, we observe that even if more than  $10^5$  MISs are involved, the optimal solution only uses a few of them ( $< 50$ ). Due to these two aspects, we are motivated to design a heuristic algorithm to smartly compute the MISs to improve the efficiency and the performance.

## VII. SIO-BASED MIS COMPUTING

In this section, we first theoretically demonstrate that only a relative small set of MISs needs to be involved in the optimal MIS based scheduling, although exponentially many MISs are associated with a given conflict graph. The set of MISs being scheduled by the optimal solution is termed as a *critical MIS set*. Then, we develop a polynomial heuristic algorithm to intelligently compute a set of MISs with the objective to cover the critical MIS set as much as possible. Our basic idea is to exploit the network topology and network flow information to infer those communication links that might be scheduled with high possibility, and define a *scheduling index* metric to differentiate the scheduling priority of links. Although in the following we develop a scheduling index ordering (SIO) based algorithm for MIS computing with the focus on the MDCG, the algorithm is generally applicable to any given conflict graph and of an interesting contribution itself.

### A. Critical MIS set

We have the following theorem regarding the cardinality of a critical MIS set.

**Theorem 1:** The number of maximal independent sets scheduled by optimal solution, i.e., the cardinality of a critical MIS set, is upbounded by  $(|\mathcal{N}| + 1)|\Lambda| + |\mathcal{L}|$ .

**Proof:** The LP MDCG-based MCF formulation, as indicated by expressions (4)–(12), can be reduced to a standard form LP problem by transforming the inequality constraints indicated in (9) and (11) into equality constraints with introducing necessary slack variables [24]. Let the vector  $\mathbf{x}$  denote the decision variables, which consists of  $\lambda$ , all the flow variables  $f_{\nu, \eta}(u, v)$ , the scheduling factors  $\alpha_m$  for all the MISs, and the induced slack variables. The feasible region of the standard form LP can be written as  $\{\mathbf{x} | \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$ , where the matrix  $\mathbf{A}$  compactly representing all the equality constraints. In such a standard form LP problem, the number of

decision variables is much larger than the number of equality constraints.

The situation of relevance here is that the optimal solution takes practical finite value. For such a case, the LP theorem tells that the optimal solutions should be a basic feasible solution, where the number of variables taking positive values (the basic variables) is upbounded by the number of linearly independent equality constraints [24]. Let's count the number of equality constraints in the standard form problem, which is exactly the same as the original LP problem. The conditions (5), (7), and (8) generate  $|\Lambda|$  constraints respectively, thus  $3|\Lambda|$  in total. The condition (6) denotes the flow balance at each node for each flow except the source and destination nodes, giving  $|\mathcal{N}||\Lambda| - 2|\Lambda|$  constraints. For the MIS constrains, (12) and (9) could be incorporated, producing  $|\mathcal{L}|$  constraints. The condition (11) then gives one more constraint. Altogether, we have  $(N + 1)|\Lambda| + |\mathcal{L}| + 1$  equality constraints in the transformed standard form problem, i.e., the matrix  $\mathbf{A}$  has  $(N + 1)|\Lambda| + |\mathcal{L}| + 1$  rows. For a conservative estimation of the number of MISs to be scheduled, that is, those  $\alpha_m$  taking positive values, we assume that all the equality constraints are linearly independent. Thus, at most  $(N + 1)|\Lambda| + |\mathcal{L}| + 1$  decision variables can take positive values. Since  $\alpha_m$  are just part of the decision variables, the positive  $\alpha_m$  values which indicate the number of MISs scheduled are therefor upbounded by  $(N + 1)|\Lambda| + |\mathcal{L}| + 1$ . ■

It is noteworthy that the upper bound of the number of critical MISs is quite loose. Given some  $\alpha_m > 0$ , there must be a set of scheduled links with  $f_{(\nu,\eta)}(u, v) > 0$  correspondingly. In other words, it is impossible that all the basic variables are  $\alpha_i$ . The significance of theorem 1 lies on that it indicates that the number of critical MISs involved in optimal scheduling is very limited, a smart method to search the critical MIS set has the great potential to improve both the computation and resource allocation efficiency, compared to randomly computing the MISs in an exponentially large space. Thus, we are going to develop an intelligent algorithm to approximate the critical MIS set, which are used by optimal solution of MCF. Our target is to design a polynomial algorithm to compute a set of MISs with acceptable complexity. Meanwhile, with the limited number of MISs, we could constructing MIS constraints in MDCG-based MCF problem to derive better network capacity, compared to that from [1] and the random search [9].

### B. Algorithm description

We observe that the set of active tuples depends on the network topology and the source destination nodes associated with the network flows. For example, if the link of a tuple is a cut edge between source and destination, this tuple will be scheduled with high probability (not definitely, because a link is associated with multiple tuples in a MR-MC network). Another example, the links on the shortest path between a source destination pair is more likely to be scheduled. In particular, we define a *scheduling index* (SI) for tuple  $p$  to quantitatively indicate the possibility that the tuple  $p$  will be used. The tuple with a larger value of SI indicates the chance

of scheduling with higher priority. The scheduling index ordering allow us to develop a systemic way to approximately computing the critical MIS set: Since the tuples with higher SI are more possible to be scheduled and thus more possible to be included in a critical MIS; our strategy is therefore to search those MISs covering the higher priority tuples first until all tuples are covered. Specifically, our algorithm consists of three phases:

*Phase 1): Calculating all shortest paths.* The length of shortest path is fixed, however, there may be two or more shortest paths; We can use the following algorithm to calculate the shortest paths from  $u$  to  $v$ , which is a modified form of the standard breadth-first search [23].

- (1) Assign the destination node  $v$  distance zero, to indicate it is zero hop away from itself, and set  $d \leftarrow 0$ .
- (2) For each vertex  $i$ , which has been assigned distance  $d$ , follow every attached edge to vertex  $j$  at the other end of the link, if  $j$  has not been assigned, assign it distance  $d + 1$ . Mark  $i$  to be the predecessor of  $j$ .
- (3) If  $j$  has already been assigned distance  $d + 1$ , then there is no need to do anything but declare  $i$  is the predecessor of  $j$ .
- (4) Set  $d \leftarrow d + 1$
- (5) Repeat from step 2 until every vertex is assigned.

We could mark all shortest paths starting from  $u$  to its predecessor, and to the predecessor of each successive vertex until  $v$  is reached. Note that we apply the algorithm within the original topology, the algorithm will complete in time  $O(|\mathcal{L}|)$  according to [23], where  $|\mathcal{L}|$  is the number of edges in the graph. Now, we get every shortest path from source to destination for each flow injected into the network.

*Phase 2): Sort all tuples in the decreasing order of scheduling index.* By the end of phase 1, we get all shortest paths for all source-destination pairs. For each tuple, we can collect the number of shortest paths traversing it. Initially we can directly calculate this number for each link from phase 1, the number of each tuple is set to the number of its corresponding link. The tuple having largest value is assigned with the highest priority. If two tuples have the same value, we compare the summation of the value of their interfering neighbors. Let  $J$  be the total number of shortest paths resulting from phase 1,  $h(p)$  denote the number of shortest paths traversing tuple  $p$ ,  $sum(p)$  denote the summation of  $h$  counting all neighbors of tuple  $p$  (i.e.,  $sum(p) = \sum_{p' \in I(p)} h(p')$ ). We quantify scheduling index for tuple  $p$  as follows:

$$SI(p) = (JK^2C|\mathcal{L}|) \times h(p) + sum(p). \quad (13)$$

The SI defined in (13) could guarantee the tuple with larger  $h$  will be always assigned with larger SI, thus higher priority, since  $JK^2C|\mathcal{L}|$  is the upper bound of  $sum(p)$ . It is still possible (although with small probability) that two tuples have the same SI value based on (13). In such a case, we will break the tie by arbitrarily setting one with higher priority and adding its SI a fractional value  $0 < \theta < 1$ . The output of our sorting algorithm is that all tuples are arranged according

to SI values in the decreasing order.<sup>1</sup> In addition, we use the expressions  $p_1 \succ p_2$  and  $p_1 \prec p_2$  to denote the relations if  $SI(p_1) > SI(p_2)$  and  $SI(p_1) < SI(p_2)$  for convenience, respectively.

```

Input : All MDCG tuples in the decreasing order of
          scheduling index
Output : A set of maximal independent sets
Initialize: set the ordered  $p \in \mathcal{P}$  as anchor tuples;
               $S[p] = \emptyset$  for all anchor tuples  $p$ ;
              /* set of ISs associated with anchor tuples */
              to-be-covered set  $\mathcal{T} = \mathcal{P}$ ;
▷ while 1 do
  pick a new anchor tuple  $p$  in decreasing order;
  if  $I^c(p) = \emptyset$  (i.e.,  $p$  has no non-neighbor tuple) then
    add the MIS  $\{p\}$  into  $S[p]$ ;
    if  $p \in \mathcal{T}$  then
       $\mathcal{T} = \mathcal{T} / \{p\}$ 
    end
  else
    * for  $p' \in I^c(p)$  &  $p' \prec p$  in decreasing order do
      for every IS  $s \in S[p]$  do
        if  $p'$  can be added into  $s$  then
          add  $p'$  into  $s$ ;
          break;
        end
      end
      if  $p'$  can not be inserted then
        create a new IS  $\{p, p'\}$ ;
        add  $\{p, p'\}$  into  $S[p]$ ;
      end
      if  $p' \in \mathcal{T}$  then
         $\mathcal{T} = \mathcal{T} / \{p'\}$ 
      end
    end
  end
  repeat the “*” for-loop for  $p' \in I^c(p)$  &  $p' \succ p$  in
  decreasing order;
  if  $\mathcal{T} = \emptyset$  (i.e., all tuples have been covered) then
    break;
  end
end
for every IS found above do
  if IS is not maximal then
    add some tuples to make it maximal;
  end
end

```

**Algorithm 1:** Phase 3: Computing maximal independent sets

<sup>1</sup>The sorting algorithm considered here is based on shortest path information computed in Phase 1. In a more general scenario, Phase 1 can take longer paths into account and assign different weights to paths with different lengths, based on which Phase 2 sorting may generate more accurate ordering. For illustration purpose, we consider only shortest path in this paper without loss of generality.

*Phase 3): Iteratively compute maximal independent sets covering all tuples.* Algorithm 1 presents all details finding MIS based on the SI ordering generated in phase 2. The algorithm starts by considering all the ordered tuples  $p \in \mathcal{P}$  as anchor points. Each anchor tuple leads to a round of independent set (IS) computing. Given an anchor tuple  $p$ , we will compute a set of ISs that cover  $p$  and all its non-neighbor nodes in the conflict graph denoted as  $p' \in I^c(p)$ . The set of ISs computed will be stored in the set  $S[p]$ . Note that each IS in  $S[p]$  contains the tuple  $p$ , showing  $p$  as an anchor point. The algorithms will run recursively along the anchor points in decreasing SI order until all the tuples are covered by certain IS(s), which guarantees that the MCF formulation can generate a feasible solution. A to-be-covered set  $\mathcal{T}$  is defined as a set containing the tuples yet to be covered by ISs. The set  $\mathcal{T}$  is initialized as  $\mathcal{T} = \mathcal{P}$ , and each time a new tuple is covered by certain IS, the tuple is removed from  $\mathcal{T}$ . The operations to search IS stop when  $\mathcal{T}$  becomes empty. After the MIS searching stops, all the ISs obtained will be extended to MISs if they were not.

It is not difficult to see that in our algorithm, the tuples with larger SI tend to be covered by more MISs due to two facts: 1) The tuples with small SI values may not get the chance to be operated as anchor points if the to-be-covered set  $\mathcal{T}$  reaches empty at a higher-priority anchor point. 2) In each IS computing round, the anchor nodes are covered in all ISs maintained in  $S[p]$ , but other tuples are not. Moreover, a noteworthy implementation detail is that when we switch to next anchor tuple  $p$ , we start with the non-neighbor tuple  $p' \in I^c(p)$  &  $p' \prec p$  instead of the  $p' \in I^c(p)$  with the larger SI, and then come back to the half set with  $p' \succ p$ . The benefit of this operation is to reduce the chance of repeating an existing IS and avoid over-concentrating the ISs to high priority tuples.

### C. Complexity analysis

In this subsection, we give an analyze to the complexity of the proposed heuristic algorithm. As indicated earlier, Phase 1 of the algorithm labels all shortest path for one source-destination pair in time  $O(|\mathcal{L}|)$ . Given  $|\Lambda|$  flows, Phase 1 computations take the polynomial complexity of  $O(|\Lambda||\mathcal{L}|)$ . The sorting operations in Phase 2 is also polynomial; the commonly used sorting algorithms have the running time between  $O(n \log_2 n)$  and  $O(n^2)$ , where  $n$  is the number of nodes. In the MDCG context,  $n$  is the number of tuples, which is upper bounded by  $K^2C|\mathcal{L}|$ , but the larger value of  $n$  does not change the polynomial nature. Regarding the complexity of computations in Phase 3, we have the following theorem:

**Theorem 2:** The Phase 3 computations in Algorithm 1 have a polynomial complexity of  $O(K^8C^4|\mathcal{L}|^4)$ .

**Proof:** Note that the number of tuples is at most  $K^2C|\mathcal{L}|$  if there are  $|\mathcal{L}|$  links in original topology. Here, the operation comparing two tuples is define as one basic operation. Consider the **while**-loop labeled by the symbol ▷, which incurs at most  $K^2C|\mathcal{L}|$  of iterations. In each iteration, given an anchor tuple  $p$ , the number of non-neighbor tuples in  $I^c(p)$  is

upbounded by  $K^2C|\mathcal{L}|$ . To determine the number of iterations incurred by the **for**-loop, we consider the worst case that all non-neighbor tuples of  $p$  pair-wisely interfere with each other; in such a case, each tuple in  $I^c(p)$  incurs a new IS. Thus, the number of ISs in  $S[p]$  is upbounded by  $K^2C|\mathcal{L}|$  too. Obviously, the number of tuples contained in each IS is upbound by  $K^2C|\mathcal{L}|$ . Based on such analysis, it can be seen the number of basic operations incurred to determine whether a  $p' \in I^c(p)$  can be added into an existing independent set or not takes the order  $O((K^2C|\mathcal{L}|)^2)$ . Based on the number of iterations included in the “\*” **for**-loop and “▷” **while**-loop, the complexity to obtain the ISs contained in all  $S[p]$  sets take the order of  $O((K^2C|\mathcal{L}|)^4)$ . After obtaining all the candidate ISs, we also need to check the maximality of each set and make necessary extension to MISs. It can be seen that the number of candidate ISs contained in all  $S[p]$  sets is upbounded by  $O((K^2C|\mathcal{L}|)^2)$ , and it takes  $O((K^2C|\mathcal{L}|)^2)$  operations to each set for the maximality checking/exten. In summary, the total time complexity is  $O((K^2C|\mathcal{L}|)^4) + O((K^2C|\mathcal{L}|)^4)$ , that is,  $O(K^8C^4|\mathcal{L}|^4)$ . ■

Combining the complexity analysis of all the three phases, the proposed heuristic algorithm is a polynomial with a complexity of  $O(K^8C^4|\mathcal{L}|^4)$  (the complexity is dominated by Phase 3 operations).

## VIII. NUMERICAL RESULTS

In this section, we present some numerical results to demonstrate the efficiency of the MDCG-based resource allocation compared to the resource allocation algorithm developed in [1], and demonstrate the efficiency of SIO-based MIS computing compared to the random MIS searching algorithm proposed in [9]. We consider two network topologies: 1) a grid topology as shown in Fig 2, where 25 nodes are distributed within a  $1000m \times 1000m$  area and the area is divided into  $200m \times 200m$  square cells; 2) a random topology as shown in Fig 3, where 25 nodes are randomly placed in a  $1000m \times 1000m$  area to form a connected network. There are 3 commodity flows in each topology. The source and destination nodes for flow  $i$  ( $i = 1, 2, 3$ ) are denoted as  $S_i$  and  $D_i$  respectively. The transmission range and interference range of each node is set to 250m and 500m, respectively. For the convenience of performance comparison with [1], we assume the physical link capacity over each channel is the same, and we set all values of  $w_{uv}^c$  to a normalized link capacity of one rate unit. Each flow has a rate demand of 3 rate units. We develop C codes to implement our heuristic algorithm and the algorithms presented in [1], and use CPLEX [25] to solve the optimization problems involved in our MCF formulation and the algorithms in [1]. For the ease of presentation, we denote the MCF solutions under our SIO-based MIS computing as SIO-MDCG, and those under the random MIS computing [9] as RS-MDCG. We denote the capacity computed based on the heuristic algorithms in [1] as static approximation (S-App), and the upper bound indicates the capacity under the LP relaxation in [1] (denoted as  $\lambda^*$  there).

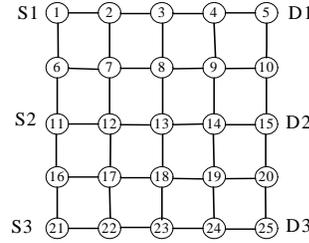


Fig. 2. Grid topology.

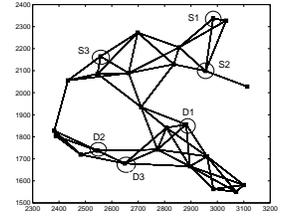
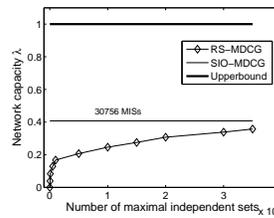
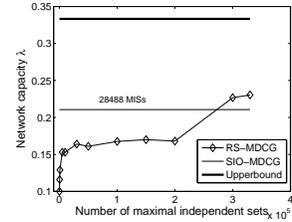


Fig. 3. Random topology.



(a) Grid topology with 4 radios, 8 channels.



(b) Random topology with 3 radios, 9 channels.

Fig. 4. Effectiveness of SIO-MDCG in computing MIS.

### A. Effectiveness of the SIO-based MIS computing.

The effectiveness of the proposed SIO-based MIS computing is to be demonstrated through comparing the SIO-MDCG results with the RS-MDCG results for both grid and random topologies. In Fig. 4, the network capacity  $\lambda$  versus the number of MISs constrains is computed under the RS-MDCG method; the upper bound is obtained according to the LP relaxation in [1]; the SIO-MDCG result is obtained with the number of MIS constrains determined by the SIO algorithm, and the number of constrains is indicated over the corresponding curves in Fig. 4.

We can have the following observations from Fig. 4. 1) The RS-MDCG curves shows that network capacity computed by the MCF formulation increases with the number of MIS constrains used. 2) Our SIO-based MIS computing algorithm outputs 28488 and 30756 MISs for the random and grid topologies, respectively. 3) In the grid topology, the SIO-MDCG results (with 30756 MIS constrains) steadily outperform the RS-MDCG result even when the RS-MDCG uses up to  $3 \times 10^5$  MIS constrains (around one order more than that used in SIO-MDCG). Looking at the case under the same number of MIS constrains (i.e., 30756), the capacity under SIO-MDCG is two times that under the RS-MDCG. Similar observations can be obtained in random topology, where the RS-MDCG needs to use as many as  $2.7 \times 10^5$  MIS constrains to catch the performance of SIO-MDCG which incurs only 28488 constrains. 4) In the grid topology, the RS-MDCG performance uniformly increases with the number of MISs as shown in Fig. 4(a), but the curve shows a step-increasing behavior in random topology as illustrated in Fig. 4(b). Such performance difference can be explained as follows. Note that the grid topology is a symmetric structure,

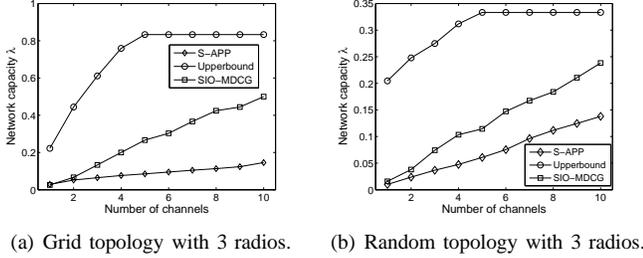


Fig. 5. The performance of SIO-MDCG.

where every link carries basically the same topological information (ignoring the borderline effect). In such a uniformed structure, the uniformed performance of random algorithm could be expected. However, in an arbitrary topology randomly configured, different links may be of different structure-level importance. For example, there might be some critical links (such as cut edge) between a given source-destination pair. In such a non-uniformed structure, if the random research misses some critical links, increasing the number of MISs will not provide much help in increasing the network capacity. An obvious capacity increase can only be achieved (shown as a jumping step in Fig. 4(b)) when those critical links are covered. The results in Fig. 4 clearly demonstrate that our SIO-MDCG algorithms derives satisfied performance in identifying those critical MISs, which further leads to efficient resource allocation.

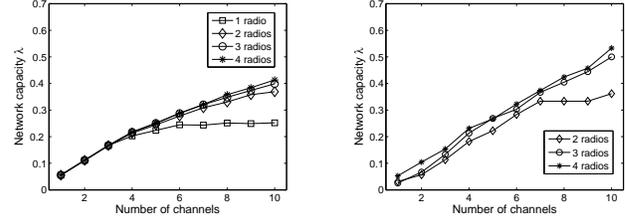
### B. MDCG-based resource allocation compared to the static heuristics in [1].

In this part, we are to demonstrate the efficiency of MDCG based resource utilization compared to the static heuristic algorithm developed in [1]. Based on the efficiency discussion in the above, the SIO-MDCG results are used to represent MDCG-based capacity. In this experiment, each node is equipped with 3 radio interfaces, and we investigate the performance by varying the number of channels. Fig. 5 presents the curves of  $\lambda$  versus the number of channels under the algorithms SIO-MDCG and S-App, respectively. It can be seen that the achieved network capacity under our algorithm steadily outperform that under the S-App algorithm. The more channels are available, the higher capacity increase can be achieved by our SIO-MDCG method, which clearly demonstrate the efficiency of SIO-MDCG in exploiting the available resources.

One fundamental reason for the capacity improvement with MDCG-based analysis is the scheduling with *dynamic channel swapping* as illustrated in Table I. The table presents the scheduling factors  $\alpha_m$  computed by SIO-MDCG for tuples associated with links (12, 13) in grid topology, for the instance of 2-interface nodes and 10 channels. The scheduling results clearly demonstrate that link (12, 13) dynamically swaps among different channels. For example, the link (12, 13) when operating over radio pair (1, 1) can switch over channels 2, 4, 6, and 9, although at a moment a link can only operate

TABLE I  
RLC-TUPLE BASED SCHEDULING: DYNAMIC CHANNEL SWAPPING FOR LINK (12, 13) IN GRID TOPOLOGY

Channel/ Radio-pair	(1,1)	(1,2)	(2,1)	(2,2)
Channel 2	0.04529	0.007246	0	0.006341
Channel 4	0.120471	0	0	0
Channel 6	0.067935	0	0	0
Channel 7	0	0.008152	0	0
Channel 8	0	0	0.203804	0
Channel 9	0.092391	0	0	0.038043
Channel 10	0	0.076993	0.088768	0



(a) RS-MDCG results,  $3 \times 10^5$  MIS searching rounds.

(b) SIO-MDCG results.

Fig. 6. Impact of number of radios and channels, grid topology.

on one channel through a given radio interface. Instead, the algorithms in [1] statically assign a channel to a link operating over a given radio interface.

### C. Capacity versus the Number of Radios and Channels.

We also compute the optimal capacity under different number of radio interfaces and channels with both RS-MDCG and SIO-MDCG. Here, we present the results for the grid topology with an illustration purpose. We will have similar observations in a random topology. For RS-MDCG, we let the random MIS search algorithm run  $3 \times 10^5$  rounds. Such a number of searching rounds is large enough for the cases with small numbers of radio interfaces and channels, but insufficient for the large-scale cases. Thus, on one hand, the RS-MDCG results in the small-scale cases (which generates close-to-optimal results with enough MIS constraints) can be used to benchmark the SIO-MDCG results. On the other hand, the RS-MDCG results in the large-scale cases can further reflect the advantage of SIO-MDCG methods.

The results are shown in Fig. 6, where we can have the following observations. 1) In the small-scale cases less than 4 channels, the RS-MDCG results in Fig. 6(a) show that the same capacity  $\lambda$  is achieved when the number of radio interface changes from 1 to 4. In other words, one radio is enough to utilize all the four channels. We see that in the grid topology, a node at most has 4 neighbors. By proper radio interface assignment and dynamic channel swapping, the four channels can successfully resolve the interference among the four neighbors even under the 1-radio configuration. Such an effect is theoretically explained in [17]. 2) In Fig. 6(a), The increment of capacity  $\lambda$  basically stops when the number of channels exceeds 6 in 1-radio case. The reason is that full exploitation of a give number of channels requires a reasonable number of available radio interfaces. Thus is practice, a

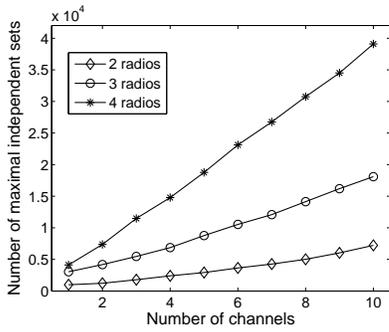


Fig. 7. Number of MIS computed by SIO-MDCG in grid topology.

good design should equip nodes with a proper number of radio interfaces to efficiently utilize the available channels. 3) Compared the results in Fig. 6(a) and Fig. 6(b), we can see that the SIO-MDCG results are very close to the RS-MDCG results in small-scale cases, showing again the good performance of SIO based computing in identifying critical MISs. Also, the comparison shows that the RS-MDCG results could not give accurate results in those large-scale cases with more number of radio interfaces and channels. First, the SIO-MDCG can give higher network capacity. Second, given 10 channels, the SIO-MDCG results show that 3 radio interfaces at each node are necessary to efficiently exploit all the channels (the 3-radio curve and 4-radio curve are very close), but the RS-MDCG results will suggest a number of 2 radios. This example again demonstrates that an efficient computing method is critical for properly planning the network capacity. 4) With the increase of number of radio interfaces and channels, the SIO-based computing will generate a larger number of MISs, which are shown in Fig. 7. We can observe from Fig. 7 that the complexity of SIO-based MIS computing in fact increases linearly, although the worst-case complexity analysis in Theorem 2 gives a very conservative bound.

## IX. CONCLUSION

In this paper, we develop a novel concept of multi-dimensional conflict graph to facilitate optimal capacity planning in MR-MC wireless networks through a LP MCF formulation. The MDCG-based MCF solution generates not only the maximum network capacity, but also the joint optimal solutions of link scheduling, radio interface and channel assignment, and routing. Moreover, the MDCG-based planning can conveniently exploit dynamic channel swapping and channel diversity for the maximum capacity. Further, we propose a heuristic algorithm to intelligently compute a small number of critical MISs to significantly reduce the computation complexity. To the best of our knowledge, this is the first work to solve the optimal capacity planning of MR-MC network in the LP regime with efficient computing methods.

## 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*, Aug. 2005, pp. 58–72.

[2] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proc. ACM MobiCom*, Aug. 2005, pp. 73–87.

[3] 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.

[4] Y. T. Hou, Y. Shi, and H. D. Sherali, "Spectrum sharing for multi-hop networking with cognitive radios," *IEEE J. Select. Areas Commun.*, vol. 26, no. 1, pp. 146–155, Jan. 2008.

[5] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2223–2234.

[6] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, Jan. 2006.

[7] H. Wu, F. Yang, K. Tan, J. Chen, Q. Zhang, and Z. Zhang, "Distributed channel assignment and routing in multiradio multichannel multihop wireless networks," *IEEE J. Select. Areas Commun.*, vol. 24, no. 11, pp. 1972–1983, Nov. 2006.

[8] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized algorithms for multi-channel wireless mesh networks," *ACM Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, Apr. 2004.

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

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

[11] K. N. Ramachandran, E. M. Belding, K. C. Almeroth, and M. M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proc. IEEE INFOCOM*, 2006.

[12] Garey, M. R., and Johnson, D. S, *Computers and Intractability: A guide to the theory of NP completeness*, W. H. Freeman and Company, 1979.

[13] M. Delakis and V. S. Siris, "Channel Assignment in a metropolitan wireless multi-radio mesh network," *Proc. 5th International Broadband Communications, Networks and Systems*, Sept. 2008, pp. 610–617.

[14] K. Rosen, *Discrete Mathematics and its applications*, McGraw Hill, 1999.

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

[16] Y. Shi, Y. T. Hou, and S. Kompella, "How to correctly use the protocol interference model for multi-hop wireless networks," in *Proc. ACM MobiHoc*, May 2009.

[17] 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. 43–57.

[18] X. Lin and S. Rasool, "A distributed joint channel-assignment, scheduling and routing algorithm for multi-channel ad hoc wireless networks," in *Proc. IEEE INFOCOM*, May 2007, pp. 1118–1126.

[19] S. Merlin, N. H. Vaidya, and M. Zorzi, "Resource allocation in multi-radio multi-channel multi-hop wireless networks" in *Proc. IEEE INFOCOM*, Apr. 2008, pp. 610–618.

[20] X. Wu and R. Srikant, "Bounds on the capacity region of multihop wireless networks under distributed greedy scheduling," in *Proc. IEEE INFOCOM*, Apr. 2006.

[21] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proc. ACM MobiHoc*, May 2005, pp. 68–77.

[22] D. S. Johnson and M. Yannakakis, "On generating all maximal independent sets," *Information Processing Letters*, vol. 27, no. 3, pp. 119–123, Mar. 1988.

[23] R. Sedgewick, *Algorithms*, Addison-Wesley, Reading, MA, 1988.

[24] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.

[25] Ilog cplex suite, <http://www.ilog.com/products/cplex/>