

A New Computational Approach for Maximum Link Activation in Wireless Networks under the SINR Model

Antonio Capone, *Senior Member, IEEE*, Lei Chen, Stefano Gualandi, and Di Yuan, *Member, IEEE*

Abstract—A fundamental and computationally challenging optimization task in wireless networks is to maximize the number of simultaneous transmissions, subject to signal-to-noise-and-interference ratio (SINR) requirements at the receivers. The conventional approach guaranteeing global optimality is to solve an integer programming model with explicit SINR constraints. These constraints are however numerically very difficult. We develop a new integer programming algorithm based on a much more effective representation of the SINR constraints. Computational experiments demonstrate that the new approach performs significantly better in proving optimality.

Index Terms—wireless networks, optimization, link activation, SINR, integer programming.

I. INTRODUCTION

We consider the optimization problem of maximizing the number of parallel transmissions in a wireless network. Transmission on a link is subject to a signal-to-noise-and-interference ratio (SINR) threshold required for successful reception according to the physical model [1]. Differently than in asymptotic analysis where random networks are considered [1], this problem is commonly considered in performance engineering of wireless networks with arbitrary topology and propagation coefficients [2]. Notably, a weighted version of this problem is the key subproblem in many radio resource management aspects, such as link scheduling [3], [4], where different sets of compatible parallel transmissions need to be selected, and routing [6], [7], where transmissions over multiple links are required.

The problem we consider has its root in link admission/activation and Spatial Time Division Multiple Access (STDMA) [8]. It has attracted a significant amount of research attention (see e.g. [9], [10] and the references therein). Its relevance within the theory of wireless networks is related to benchmarking and performance analysis of protocols and systems, but it also has practical validity in static all-wireless networks (like Wireless Mesh Networks) where propagation conditions are almost constant and a centralized management of radio resources is applicable [7], [11].

The problem is NP-hard with nodes distributed in Euclidean space even under uniform node power [9], and even if the background noise is neglected [10]. Approximation algorithms with ratio dependent on the number of connections [11], or on

geometric characteristics [9], [10] are available. Algorithms with constant approximation guarantee have been proposed [2], [12] under the uniform power assumption. Recently, a constant-factor approximation algorithm for the general case of variable power has been developed [5].

In contrast, very little work has been done for approaching global optimality [7]. Yet, exact algorithms are of great interest for performance evaluation and designing resource optimization strategies in order to assess the actual potential improvement of heuristic schemes. Moreover, solving the problem to optimum allows for making fair comparisons of transmission, medium access, and routing schemes exploiting parallel transmissions and cooperation among nodes.

To our knowledge, the conventional approach in previous works for guaranteeing global optimality is to solve an integer linear model with explicit SINR constraints. These constraints are however numerically very difficult, mainly because propagation gains vary significantly in magnitude. In this letter, we propose a new integer programming algorithm based on a numerically equivalent but much more effective representation of the SINR constraints. We show that the new approach performs significantly better in proving optimality or confining the optimum within a tight interval.

II. PRELIMINARIES

Let V denote the node set of a wireless network. The nodes use fixed transmit power. The algorithm we present does not assume uniform power, although the networks used in the experiments have this characteristic. We use the following notation: P_i is the power of $i \in V$, g_{ij} is the gain between nodes i and j , η is the noise effect, and γ is the SINR threshold. Simultaneous transmissions are subject to several constraints: 1) a node can be a sender or receiver, but not both simultaneously, 2) a node can send to at most one receiver or receive from at most one sender, and 3) all links in transmission (i.e., activated links) must have the SINR threshold satisfied. Clearly, i can send to j , only if the signal-to-noise ratio (without any interference) satisfies $\frac{P_i g_{ij}}{\eta} \geq \gamma$. Let arc set A denote the resulting collection of links. When multiple links are activated, the SINR requirement of (i, j) reads $\frac{P_i g_{ij}}{\sum_{k \in I \setminus \{i\}} P_k g_{kj} + \eta} \geq \gamma$, where I is the set of active senders.

The optimization problem is to admit as many links as possible, i.e., to find a subset of A with maximum cardinality, such that the SINR threshold of all links in the

A. Capone and S. Gualandi are with Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.

L. Chen and D. Yuan are with the Department of Science and Technology, Linköping University, Sweden.

subset are simultaneously fulfilled. Denote by L^* the optimal value, and LB and UB lower and upper bounds of L^* , i.e., $L^* \in [LB, UB]$. An *exact algorithm* always guarantees global optimum L^* . *Heuristics* give LB of unknown quality. An *approximation algorithm* with a constant approximation ratio k produces, for any instance, LB such that $L^* \leq UB = k \cdot LB$. Thus the ratio k has to be derived for the theoretically worst case.

Obviously, heuristics and approximation algorithms can deal with much larger networks than exact algorithms. To *accurately* assess the *numerical performance limit* of maximum link activation, as well as to evaluate heuristics, however, finding L^* or obtaining a *numerically* tighter interval than that of an approximation algorithm, is of great importance. Our contribution is a new *exact algorithm* that significantly improves the scalability of the current one in delivering global optimum L^* , or *numerically* computing a tight interval $[LB, UB]$.

III. THE CONVENTIONAL APPROACH

Let binary variables x_{ij} and y_i denote, respectively, if link (i, j) is activated and if node i is transmitting. Solving the following integer linear model is the conventional approach for obtaining global optimum of link activation.

$$[M1] \quad L^* = \max \sum_{(i,j) \in A} x_{ij} \quad (1)$$

$$\text{s. t.} \quad \sum_{j \in V: (i,j) \in A} x_{ij} + \sum_{j \in V: (j,i) \in A} x_{ji} \leq 1, i \in V \quad (2)$$

$$\sum_{j \in V: (i,j) \in A} x_{ij} = y_i, i \in V, \quad (3)$$

$$P_i g_{ij} x_{ij} + M_{ij} (1 - x_{ij}) \geq \gamma \left(\sum_{k \neq i} P_k g_{kj} y_k + \eta \right), (i, j) \in A, \quad (4)$$

$$x_{ij} \in \{0, 1\}, (i, j) \in A, \quad (5)$$

$$y_i \in \{0, 1\}, i \in V. \quad (6)$$

The objective (1) maximizes the number of admitted links. Inequalities (2) state that, among the incoming and outgoing links of a node, at most one can be activated. This corresponds to the first two constraints in Section II. The y -variables, although can be eliminated by (3), are kept for the sake of clarity. Inequalities (4) formulate the SINR requirement. If $x_{ij} = 1$, (4) constrains SINR to be at least γ . For $x_{ij} = 0$, the constraint is always satisfied for a sufficiently large number M_{ij} ; this is the case if M_{ij} equals the right-hand side of (4), with all y -variables set equal to one.

The difficulty of solving M1 stems from (4). Solving an integer linear model relies heavily on the bound from the continuous relaxation. The big number M_{ij} , known as big-M in integer programming, makes the continuous relaxation very weak. Moreover, the gain values in (4) vary significantly in magnitude and cause numerical difficulties in problem solution.

IV. A NEW EXACT ALGORITHM

A. A Reformulation

If link (i, j) is active, (4) reduces to a *knapsack* constraint: $\sum_{k \neq i} b_{kj} y_k \leq r_{ij}$, where $b_{kj} = P_k g_{kj}$ and $r_{ij} = \frac{P_i g_{ij}}{\gamma} - \eta$. We reformulate this knapsack constraint by using *cover* inequality-type cutting planes. A set $C \subseteq V \setminus \{i\}$ is called a cover, if $\sum_{k \in C} b_{kj} > r_{ij}$. The SINR of (i, j) threshold requires that at most $|C| - 1$ nodes in C may transmit. We obtain the most basic form of SINR cover inequality: $\sum_{k \in C} y_k \leq |C| - x_{ij}$. Note that the inclusion of x_{ij} in the right-hand side restricts the activation of links in C to at most $|C| - 1$, only if link (i, j) itself is active.

Using the cover-type inequality to reformulate the SINR requirements yields the following new integer model.

$$[M2] \quad \max \sum_{(i,j) \in A} x_{ij}$$

$$\text{s. t.} \quad (2), (3), (5), (6)$$

$$\sum_{k \in C} y_k \leq |C| - x_{ij},$$

$$(i, j) \in A, C \subseteq V : \sum_{k \in C} b_{kj} > r_{ij}. \quad (7)$$

Note that the explicit SINR constraints (4) are *completely eliminated* in M2. What's more, unlike (4) with big-M and gain values, (7) does not contain any numerically difficult coefficient.

Theorem 1: The integer formulation M2 is correct, that is, its optimal point represents a feasible activation solution of L^* links.

Proof: Consider any link activation solution that is infeasible due to SINR requirement, and let the set of transmitters be T . The solution contains at least one link (i, j) , for which the total interference $\sum_{k \in T \setminus \{i\}} P_k g_{kj}$ exceeds the budget $\frac{P_i g_{ij}}{\gamma} - \eta$. The set $T \setminus \{i\}$ forms an inequality of (7) in M2. Therefore the optimum of M2 must be feasible. Moreover, since (7) is derived for SINR violation, the inequalities do not cut off any feasible solution. Hence the correctness. ■

A special case of (7) is a set C with cardinality one, i.e., knapsack violation due to one interfering sender:

$$y_k \leq 1 - x_{ij}, (i, j) \in A, k \neq i, j : b_{kj} > r_{ij}. \quad (8)$$

We remark that the SINR requirement can be represented using a linearization of the bi-linear constraint $\sum_{k \neq i} b_{kj} y_k x_{ij} \leq r_{ij}$. The linearization uses an auxiliary binary variable z_{ij}^k to replace the bi-linear product. Variable z_{ij}^k takes value one only if both link (i, j) and node k are active. This effect can be achieved by linear inequality $z_{ij}^k \geq y_k + x_{ij} - 1$. Computational experiments indicate, however, that the linearization does not perform better than M1. We therefore do not discuss the linearization further in the rest of the paper.

B. Stronger Inequalities

We can not solve M2 in its complete form, as the number of constraints of type (7), in general, grows exponentially

in $|V|$. On the other hand, optimum to knapsack problems can typically be reached by introducing a handful of cover inequalities. Thus the task is to identify those inequalities that are necessary to define the global optimum point. To this end, our algorithm integrates the following notions.

- 1) An iterative approach solving first a model containing a small number of constraints of type (7), such as (8), followed by repeatedly introducing new inequalities that are violated by the solution, which will then be cut off, and re-solving the model.
- 2) Identification of as strong inequalities as possible in each iteration.

Let (\bar{x}, \bar{y}) denote an integer solution satisfying (2), (3), and any subset of (7). Verifying whether or not (\bar{x}, \bar{y}) violates the SINR of any link (i, j) with $\bar{x}_{ij} = 1$ is straightforward. If the SINR threshold is exceeded, the nodes with value one in \bar{y} lead to a violated inequality of (7). However, (7) generated in this way is weak. To get substantially stronger inequalities, we generate (7) having a minimum number of elements in the left-hand side, corresponding to so called minimum cover for knapsack problems. Suppose $\sum_{k \neq i} b_{kj} \bar{y}_k > r_{ij}$. We minimize the number of interfering nodes we pick before the sum exceeds r_{ij} . Doing so amounts to sorting the elements in $\{b_{kj} : \bar{y}_k = 1\}$ in descending order, and following the sorted sequence until the accumulated sum goes above r_{ij} . Denoting the resulting index set by K . We obtain inequality $\sum_{k \in K} y_k \leq |K| - x_{ij}$. We strengthen it further by subtracting additional x -variables in the right-hand side, as described below. The procedure is called lifting in integer programming. An illustration of the links involved in the lifting procedure is shown in Figure 1.

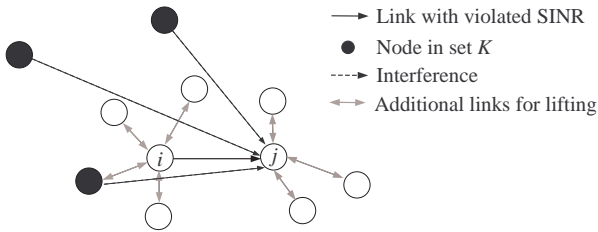


Fig. 1. An illustration of cut generation.

Consider link (i, h) with $h \neq j$, and assume first $h \notin K$. If link (i, h) can tolerate interference from at most m nodes in K and $m < |K|$, we can strengthen the inequality to $\sum_{k \in K} y_k \leq |K| - x_{ij} - a_{ih}^K x_{ih}$, where $a_{ih}^K = |K| - m$. To see the validity of the new inequality, note that x_{ij} and x_{ih} can not be one simultaneously, which is ensured by constraints (2). Thus, if $x_{ij} = 1$, then $x_{ih} = 0$, and the inequality reduces to $\sum_{k \in K} y_k \leq |K| - x_{ij}$. If $x_{ih} = 1$, then $x_{ij} = 0$, and the inequality allows at most m transmitters in K to be active. To find out m , we sort the nodes in K in ascending order in their signal strengths at h , and determine how many nodes in the sorted sequence can be active before the accumulated interference exceeds r_{ih} . The discussion so far applies to $h \notin K$. If $h \in K$, the calculation of m is adjusted for $K \setminus \{h\}$, and the coefficient $a_{ih}^K = |K| - 1 - m$ if $m < |K| - 1$. A similar lifting procedure applies to any incoming link (h, i) having i

```

1:  $(Status, UB, (\bar{x}, \bar{y})) \leftarrow Solve((1), (2), (3), (5), (6), (8))$ 
2:  $LB \leftarrow LinkRemoval(\bar{x}, \bar{y}); L = LB + 1; Q \leftarrow \emptyset$ 
3: repeat
4:    $Q \leftarrow Q \cup Generate(\bar{x}, \bar{y})$ 
5:    $(Status, \bar{L}, (\bar{x}, \bar{y})) \leftarrow Solve((1), (2), (3), (5), (6), (8), Q, L)$ 
6:   while  $Status = Solution\ exists\ and\ (\bar{x}, \bar{y})\ feasible$  do
7:      $LB \leftarrow LB + 1; L \leftarrow LB + 1$ 
8:      $(Status, \bar{L}, (\bar{x}, \bar{y})) \leftarrow Solve((1), (2), (3), (5), (6), (8), Q, L)$ 
9:   end while
10:  if  $Status = Solution\ exists$  then
11:     $LB \leftarrow \max\{LB, LinkRemoval(\bar{x}, \bar{y})\}; UB \leftarrow \min\{UB, \bar{L}\}$ 
12:  end if
13: until  $Status = No\ feasible\ solution$ 

```

Fig. 2. Algorithm description.

as the receiver. Next, we make the observation that at most one of the outgoing and incoming links of i can be active. Hence the strengthening terms of all links of i , except (i, j) , can all be part of the new inequality. The lifted inequality has the following general form.

$$\sum_{k \in K} y_k \leq |K| - x_{ij} - \sum_{h \neq j: (i, h) \in A} a_{ih}^K x_{ih} - \sum_{h: (h, i) \in A} a_{hi}^K x_{hi}. \quad (9)$$

Inspecting Figure 1, one can derive a second inequality based on (i, j) and the links of j . The inequality is given below, and the calculation of its coefficients is analogous to (9).

$$\sum_{k \in K} y_k \leq |K| - x_{ij} - \sum_{h: (j, h) \in A} a_{jh}^K x_{jh} - \sum_{h \neq i: (h, j) \in A} a_{hj}^K x_{hj}. \quad (10)$$

C. Algorithm Summary

Figure 2 summarizes the algorithm. The first line solves the model defined by maximizing (1), subject to (2), (3), and (8), to optimality by a solver. The last set of constraints are the cover-type inequalities having one interfering transmitter. The optimum initializes UB . Assume the solution is not SINR-feasible (otherwise $L^* = UB$). A link-removal heuristic (*LinkRemoval*) is used to derive a feasible set of active links. Link removal considers the subset of links having infeasible SINR. Among them, the one causing largest total interference to all remaining links with violated SINR is removed first, and the SINR of the other links are updated. This is repeated until feasibility is reached. The *LinkRemoval* procedure results in the initial value of LB (line 2). Also, the pool of inequalities of (9) and (10), denote by Q , is initialized.

The bulk of the algorithm alternates between generating (9) and (10) to augment Q , of which the procedure is described in the previous section and denoted by *Generate* in line 4 of the figure, and optimizing the augmented model (line 5). For better efficiency, we do not require optimality in every iteration, but use a target value $L = LB + 1$. The solution process is halted, once the solver finds (\bar{x}, \bar{y}) with L active links, or proves that no such solution exists ($Status = no\ solution$). We use \bar{L} to denote the best upper estimation of optimum, originated from

the branch and bound process in the solver. As long as (\bar{x}, \bar{y}) happens to satisfy the SINR threshold, we update LB and the target value L (line 7), and resume the solution process (line 8). If (\bar{x}, \bar{y}) is not feasible, the link-removal procedure is applied, and LB and UB are updated in case of improvement as shown in line 11. This is followed by the next round of generating inequalities.

Theorem 2: The algorithm in Figure 2 converges to L^* within a finite number of iterations.

Proof: The algorithm terminates only if no solution with $L = LB + 1$ exists. At this stage, clearly $LB = L^*$. When this condition is not met, either LB increases, or new inequalities cutting off the current solution (\bar{x}, \bar{y}) are added. In the former case, the bounding interval becomes strictly tighter. In the latter case, at least one infeasible solution is excluded and will never be returned. These, together with the fact that L^* is integral, establish the result. ■

V. PERFORMANCE EVALUATION

We have used six network groups ranging from 50 to 100 nodes for performance comparison between the conventional but so far state-of-the-art approach of solving M1 directly by a solver, and the presented approach based on reformulation and mathematical programming concepts. Each network group has five instances¹. The nodes are randomly placed on a square area of 100 m². The gain parameter $g_{ij} = d_{ij}^{-3}$, where d_{ij} is the distance between i and j . The noise effect $\eta = 10^{-10}$ mW, and the SINR threshold $\gamma = 10$. The transmit power is uniformly set to 0.01 W. The link set A is defined by node pairs having a signal-to-noise ratio satisfying γ . On the algorithmic side, we have used the CPLEX (version 10.1, [13]) integer solver with its default options in implementing both solution approaches. Thus in the experiments, the solver generates its standard, built-in cutting planes, and applies branch-and-cut.

TABLE I
COMPUTING TIME (SECONDS) REQUIRED FOR REACHING GLOBAL OPTIMUM.

Netw.	Links	Convent. model	New algorithm	Netw.	Links	Convent. model	New algorithm
50-1	276	43	26	70-1	588	4161	126
50-2	280	43	13	70-2	630	10071	63
50-3	266	39	7	70-3	610	16333	196
50-4	288	37	3	70-4	560	6682	136
50-5	306	37	5	70-5	644	2751	546
60-1	404	502	55	80-1	732	>> 10 hours	815
60-2	412	758	24	80-2	826	>> 10 hours	11252
60-3	408	947	13	80-3	800	26163	4268
60-4	442	318	1	80-4	708	>> 10 hours	420
60-5	408	112	10	80-5	736	>> 10 hours	1763

In Table I, we compare the time required for reaching optimality (i.e., finding and proving L^*) for networks of up to 80 nodes. The time values are specified in seconds. The table also contains the number of links of each network. From the table, it is evident that the new algorithm based on inequalities from SINR cover substantially outperforms the conventional approach in proving and approaching global optimum. For smallest-sized networks of 50 nodes, the time difference is significant, even if not dramatic. For networks of 60 and 70

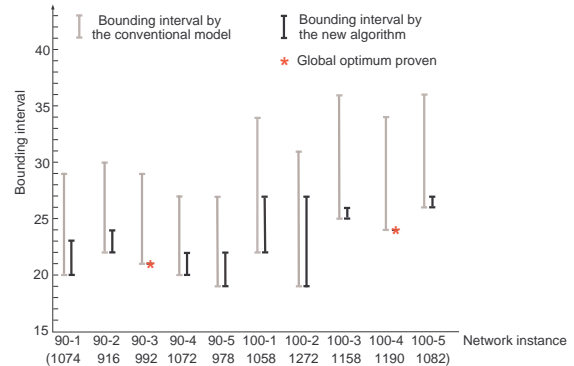


Fig. 3. A comparison of the bounding interval.

nodes, the speed-up becomes of one or even two magnitudes. For 80 nodes, optimum can no longer be found and proved by the conventional model after 10 hours of computation, except one case where the computing time is about 7 hours. (The notation “>> 10 hours” indicates that reaching optimality will most likely take days, provided that memory requirement is not the limiting factor.) The new algorithm is able to reach optimality, with a computing time ranging from minutes to at most two hours. These results demonstrate the benefit of the reformulation of the numerically-difficult SINR constraints.

For networks of 90 and 100 nodes, proving global optimum becomes time-consuming. For these networks, we illustrate and compare in Figure 3 the bounding interval $[LB, UB]$ after 5 hours of computation. The numbers of links are shown within parentheses. The results show that the new algorithm delivers significantly tighter interval, and is also able to prove optimality in two cases. Note that the two approaches find identical LB (i.e., feasible link activation) in all cases, and the tightening is due to better UB .

We remark that our algorithmic construction of repeatedly generating violated constraints/cuts not only applies to M2, but also generalizes to the conventional model M1. In the latter case, the generation of cover-type inequality is replaced by adding successively violated SINR constraints (4). Experiments show that, although this approach leads sometimes to improvement in computing time of M1, the performance of M2 remains far more superior. Thus the huge gain in solution time is mostly due to our new way of modeling the SINR requirement.

VI. CONCLUSIONS

We have presented a new exact algorithm for maximum link activation. Our notion is to reformulate the SINR constraint using more effective inequalities. The new algorithm substantially outperforms the approach of solving the conventional formulation. A future work is to develop an algorithm extension to the more general form of the objective, where each link is associated with a non-negative weight. For this problem generalization, the current algorithm can be applied by skipping the use of the target value L . A refined approach is to, in the branch-and-cut tree, add violated cover-type cutting planes each time an integer solution with better objective value is found. Further investigations include the extension of our

¹The data sets can be downloaded from <http://www.itn.liu.se/~diyua>

exact (and numerically stable) approach to adjustable transmit power, and the study of its potential benefit in link scheduling.

REFERENCES

- [1] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. on Information Theory*, 46:388–404, 2000.
- [2] O. Goussevskaia, M. M. Halldórsson, R. Wattenhofer, and E. Welzl. Capacity of arbitrary wireless networks. In *Proc. of IEEE INFOCOM '09*, 2009.
- [3] P. Björklund, P. Värbrand, and D Yuan. A column generation method for spatial TDMA scheduling in ad hoc networks. *Ad hoc networks*, 2:405–418, 2004.
- [4] A. Capone and G. Carello. Scheduling optimization in wireless mesh networks with power control and rate adaptation. In *Proc. of IEEE SECON*, 2006.
- [5] T. Kesselheim. A constant-factor approximation for wireless capacity maximization with power control in the SINR model. *Proc. of ACM-SIAM SODA '11*, 2011.
- [6] Y. Li and A. Ephremides. A joint scheduling, power control, and routing algorithm for ad hoc wireless networks. *Ad Hoc Networks*, 5:959–973, 2007.
- [7] A. Capone, G. Carello, I. Filippini, S. Gualandi, and F. Malucelli. Solving a resource allocation problem in wireless mesh networks: a comparison Between a CP-based and a classical column generation. *Networks*, 55:221–233, 2010.
- [8] R. Nelson and L. Kleinrock. Spatial-TDMA: A Collision free Multihop Channel Access Protocol. *IEEE Trans. on Communication*, 33:934–944, 1985.
- [9] O. Goussevskaia, Y. A. Pswald and R. Wattenhofer. Complexity in geometric SINR. In *Proc. of ACM MobiHoc '07*, 2007.
- [10] M. Andrews and M. Dinitz. Maximizing capacity in arbitrary wireless networks in the SINR model: complexity and game theory. In *Proc. of IEEE INFOCOM '09*, 2009.
- [11] G. Brar, D.M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proc. of ACM MobiCom '06*, 2006.
- [12] X. Xu and S. Tang. A constant approximation algorithm for link scheduling in arbitrary networks under physical interference model. In *Proc. ACM FOWANC '09*, 2009.
- [13] *ILOG CPLEX, User's Manual*, 2006.