EURO J Comput Optim (2014) 2:1–15
DOI 10.1007/s13675-013-0015-y

ORIGINAL PAPER

Optimizing compatible sets in wireless networks through integer programming

Yuan Li · Michał Pióro · Di Yuan · Jinshu Su

Received: 17 December 2012 / Accepted: 26 September 2013 / Published online: 7 November 2013

Abstract In wireless networks, the notion of compatible set refers to a set of radio links that can be simultaneously active with a tolerable interference. Finding a compatible set with maximum weighted revenue from the parallel transmissions is an important optimization subproblem for resource management in wireless networks. For this subproblem, two basic ways of expressing the signal-to-interference-plus-noise-ratio within integer programming are used, differing in the number of variables and the quality of the upper bound given by their linear relaxations. To our knowledge, there is no systematic study comparing the effectiveness of these two approaches to the compatible set optimization problem. The contribution of the article is twofold. First, we present a comparison of the two basic approaches, and, second, we introduce matching inequalities that improve the upper bounds achievable with the two basic approaches. The matching inequalities are generated within the branch-and-cut process using a minimum odd-cut generation procedure based on the Gomory–Hu tree algorithm. The article presents results of a numerical study illustrating our statements and findings.

Y. Li (✉) · M. Pióro
Department of Electrical and Information Technology, Lund University, Lund, Sweden
e-mail: yuan.li@eit.lth.se

M. Pióro
Institute of Telecommunications, Warsaw University of Technology, Warsaw, Poland
e-mail: michal.pioro@eit.lth.se

D. Yuan
Department of Science and Technology, Linköping University, Linköping, Sweden
e-mail: di.yuan@liu.se

J. Su
School of Computer, National University of Defense Technology, Changsha, China
e-mail: sjs@nudt.edu.cn

Keywords SINR · Branch-and-cut · Matching polytope · Gomory–Hu tree · Minimum odd-cut generation

Mathematics Subject Classification (2000) 90C11 Mixed integer programming · 90B18 Communication networks · 68R10 Graph theory

Introduction

In radio communications, a compatible set (CS) is defined as a group of radio links that can be simultaneously active (i.e., they can transmit in parallel) so that the signal-to-interference-plus-noise-ratio (SINR) constraint is satisfied for each of the active links. The use of CS is a powerful approach to various kinds of optimization problems in wireless networks involving transmission scheduling. For example, compatible sets are applied in such papers for optimizing routing, power control and traffic throughput as (Capone et al. 2010a; Elbatt and Ephremides 2004; Viswanathan and Mukherjee 2006; Li and Ephremides 2007). In Pióro et al. (2011) a max-min fair flow allocation problem in wireless mesh networks is considered, combining link rate adaptation with transmission scheduling—the CS subproblem is embedded as the pricing problem in a branch-and-price algorithm for solving the overall problem. Our previous work described in Li et al. (2012) develops a simulated annealing algorithm to generate a sub-optimal solution consisting of several CSs for the problem of joint link rate assignment and transmission scheduling. The papers (Björklund et al. 2004; Capone et al. 2010b) also treat generating compatible sets through a pricing problem in column generation for various applications in wireless networks.

The problem of finding a CS that maximizes a weighted sum of its links—maximum weighted CS problem (MWCSP)—has attracted a considerable research attention. For example, the papers (Goussevskaia et al. 2007, 2009) study the complexity of the problem and point out that it is $\mathcal{NP}$-hard, also when the background noise is ignored. The paper (Brar et al. 2006) investigates MWCSP in wireless mesh networks, proposing heuristic and approximate algorithms; other approximate algorithms for MWCSP can be found in Xu and Tang (2009) and Kesselheim (2011). The paper (Andrews and Dinitz 2009) presents a distributed algorithm for MWCSP based on game theory. As pointed out in Capone et al. (2011), there is not much work on finding exact solutions for MWCSP. The authors, however, present an exact MWCSP algorithm based on knapsack lifted-cover inequalities, and the scheme is especially effective when the link weights are all set to 1.

In this article, we first summarize two different integer programming models for MWCSP. Then, we strengthen the presented MWCSP models using the results of matching theory, and make numerical comparisons. The numerical study shows that the linear relaxation of the enhanced model gives an improvement in terms of the upper bound. In addition, it turns out that embedding a matching cut generation procedure in branch-and-bound results in a branch-and-cut algorithm that is more efficient than applying branch-and-bound to the standard models with no matching inequalities.

Problem description

Network model and notation

A wireless network can be represented as a directed graph $\mathcal{H} = (\mathcal{V}, \mathcal{A})$ with the set of nodes $\mathcal{V}$ and the set of directed radio links (arcs) $\mathcal{A}$. The originating and terminating nodes of an arc $a \in \mathcal{A}$ are denoted by $s(a)$ and $t(a)$, respectively, so for arc $a = (v, w)$ $(v, w \in \mathcal{V})$ we have $s(a) = v$ and $t(a) = w$. We assume that graph $\mathcal{H}$ is bi-directional, i.e., if arc $a = (v, w)$ is provided $(a \in \mathcal{A})$ then so is the opposite arc $a' = (w, v)$ $(a' \in \mathcal{A})$. The set of all arcs incident to node v is denoted by $\delta(v)$. The sets $\delta^+(v)$ and $\delta^-(v)$ denote the outgoing arcs and the incoming arcs of v, respectively, so that $\delta(v) = \delta^+(v) \cup \delta^-(v)$.

Let $\mathcal{S}$ $(\mathcal{S} \subseteq \mathcal{A})$ denote the set of all active arcs at an arbitrary time instant of network operation. P_{vw} represents the power signal received by node w when node v is transmitting. The arcs in $\mathcal{S}$ can be active simultaneously only if an appropriate SINR condition is satisfied for each arc in $\mathcal{S}$, as formulated in (1) where $\mathcal{V}_{\mathcal{S}}$ is the set of active transmitting nodes in $\mathcal{S}$: $\mathcal{V}_{\mathcal{S}} = \{s(a) : a \in \mathcal{S}\}$. For a given arc a, the SINR condition implies that the ratio between the signal received at the end node $t(a)$ to the sum of the noise N plus the signals received from the nodes other than $s(a)$ should meet a certain threshold denoted by γ.

$$\frac{P_{s(a)t(a)}}{N + \sum_{v \in \mathcal{V}_{\mathcal{S}} \setminus \{s(a)\}} P_{vt(a)}} \geq \gamma \quad a \in \mathcal{S} \tag{1}$$

To simplify the analysis, we assume that all nodes in the considered network use the same frequency. At a time instant, a node v can either transmit or receive, and only one arc incident to v can be active. For any set of active arcs we thus have

$$|\mathcal{S} \cap \delta(v)| \leq 1 \quad v \in \mathcal{V} \tag{2}$$

A subset $\mathcal{C} \subseteq \mathcal{A}$ is called a CS if it satisfies constraints (1) and (2).

Problem formulations

MWCSP is given in formulation (3). It consists in finding a CS $\mathcal{C}$ that maximizes $\sum_{a \in \mathcal{C}} c_a$ for the given arc weights $c_a \geq 0, a \in \mathcal{A}$. Our model for MWCSP uses binary variables $Y = (Y_a, a \in \mathcal{A})$ and $X = (X_v, v \in \mathcal{V})$ to represent, respectively, whether or not arc a and node v are active.

$$\max \quad \sum_{a \in A} c_a Y_a \tag{3a}$$

$$\sum_{a \in \delta(v)} Y_a \leq 1 \qquad v \in \mathcal{V} \tag{3b}$$

$$\sum_{a \in \delta^+(v)} Y_a = X_v \qquad v \in \mathcal{V} \tag{3c}$$

$$\frac{P_{s(a)t(a)}}{N + \sum_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)} X_v} \geq \gamma Y_a \qquad a \in \mathcal{A} \tag{3d}$$

$$Y \text{ binary}, \quad X \text{ continuous}. \tag{3e}$$

The objective is to find a maximum weighted sum of arcs. If $c_a = 1$ for all $a \in \mathcal{A}$, then the objective is to maximize the cardinality of a CS. Constraint (3b) assures that at most one arc a incident to node v is active. Constraint (3c) states that node v is active if, and only if, one of its outgoing arc is active. Constraint (3d) represents the SINR condition which is active only when arc a is active, i.e., when $Y_a = 1$. In the model, Y are binary variables while variables X are forced to be binary by constraints (3b) and (3c). Clearly, the CS $\mathcal{C}$ resulting from an optimal solution Y^* of (3) is defined as $\mathcal{C} = \{a \in \mathcal{A} : \ Y_a^* = 1\}$.

Constraint (3d) is nonlinear. There are (at least) two ways to linearize it (Pióro et al. 2011; Capone et al. 2011). One is to introduce variables Z_{av} that express the product $Y_a \cdot X_v$ (we will call the resulting model the Z-model), while the other is to introduce a big constant M (the M-model).

In the Z-model, the following constraints are used to make the additional variables Z_{av} express the products $Y_a \cdot X_v$:

$$Z_{av} \geq Y_a + X_v - 1 \qquad a \in \mathcal{A}, v \in \mathcal{V} \tag{4a}$$

$$Z_{av} \leq Y_a, \ Z_{av} \leq X_v, \ Z_{av} \geq 0 \qquad a \in \mathcal{A}, v \in \mathcal{V} \tag{4b}$$

where Y_a and X_v are binary. Constraints (4a) and (4b) imply that $Z_{av} = Y_a X_v$, i.e., $Z_{av} = 1$ if both Y_a and X_v are equal 1, and 0 otherwise. Then the SINR constraint (3d) is expressed in a linear way as:

$$\gamma \left(N Y_a + \sum_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)} Z_{av} \right) \leq P_{s(a)t(a)} \quad a \in \mathcal{A} \tag{5}$$

The Z-model is obtained when (3d) in model (3) is replaced by (4a), (4b) and (5).

Another way is using a "big M" to express the SINR condition:

$$\gamma \left(N + \sum_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)} X_v \right) \leq P_{s(a)t(a)} + M_a (1 - Y_a) \quad a \in \mathcal{A} \tag{6}$$

where $M_a = \gamma (N + \sum_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)}) - P_{s(a)t(a)}, \ a \in \mathcal{A}$. When $Y_a = 1$, inequality (6) expresses the SINR condition for arc $a \in \mathcal{A}$. For $Y_a = 0$, the inequality holds trivially due to the use of "big M" on the right-hand side of (6). Replacing constraint (3d) in model (3) by (6) results in the M-model.

Note that setting $M_a = \gamma (N + \sum_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)}) - P_{s(a)t(a)}$ accounts for the worst possible interference condition when all nodes $\mathcal{V} \setminus \{s(a)\}$ generate interference at $t(a)$. This is not a very likely case as the transmissions originated at these nodes are subject to the constraints of MWCSP as well. Finding the minimum and a valid value of M_a for the M-model is however not practical, as it requires solving one MWCSP per arc.

As will be illustrated in Sect. 4, inequality (5) is in general stronger than inequality (6) in terms of the upper bound produced by the corresponding linear relaxations. That is, the solution of the linear relaxation of the Z-model gives a better (i.e., smaller) upper bound than the solution of the linear relaxation of the M-model.

Reformulation based on the matching polytope

Recall that the network graph $\mathcal{H} = (\mathcal{V}, \mathcal{A})$ is bi-directional, i.e., if $(v, w) \in \mathcal{A}$, then also $(w, v) \in \mathcal{A}$. Now consider the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the undirected edges $e \in \mathcal{E}$ corresponding to the respective pair of bi-directed arcs of $\mathcal{A}$ in the original graph $\mathcal{H}$. Let the two oppositely directed arcs corresponding to edge $e \in \mathcal{E}$ be denoted by $a'(e)$ and $a''(e)$. We observe that constraints (3b) and (3e) imply that a CS in fact induces a matching in graph $\mathcal{G}$ (for the notion of matching see Korte and Vygen 2006; Schrijver 2003). This observation leads to enhanced MWCSP formulations. Below we shall extend the Z-model and the M-model by adding matching inequalities. This, as will be illustrated in Sect. 4, improves the upper bound.

Reformulations

Let $x = (x_e, e \in \mathcal{E})$ be a vector of binary variables defined for the edges of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. For any $F \subseteq \mathcal{E}$, let $x(F)$ denote the sum $\sum_{e \in F} x_e$. A set of edges $\mathcal{W}(x)$ specified by the characteristic vector x, $\mathcal{W}(x) = \{e \in \mathcal{E} : \; x_e = 1\}$, is called a *matching* if $|x(\delta(v))| \leq 1$ for all $v \in \mathcal{V}$. A basic result of the matching theory states that the convex hull of all x corresponding to matchings in $\mathcal{G}$, known as Edmond's matching polytope, is defined as follows (Korte and Vygen 2006; Schrijver 2003):

$$\sum_{e \in \delta(v)} x_e \leq 1 \qquad v \in \mathcal{V} \tag{7a}$$

$$x(E[U]) \leq \frac{|U| - 1}{2} \qquad U \in \mathcal{O}(\mathcal{V}) \tag{7b}$$

$$x_e \geq 0 \text{ and continuous} \qquad e \in \mathcal{E} \tag{7c}$$

where $\mathcal{O}(\mathcal{V})$ denotes the family of all subsets of $\mathcal{V}$ with an odd cardinality greater than 1 (odd-sets), and $E[U]$ denotes the set of all edges in $\mathcal{E}$ with both ends in U.

Inserting inequalities (7), among which inequalities (7b) are called matching inequalities, to the Z-model, we obtain the ZC-model (8) that is formulated below. Note that additional constraints (8d) relate edge selection x to arc activation Y: edge $e \in \mathcal{E}$ must be selected ($x_e = 1$), if one of its arcs $a'(e)$ or $a''(e)$ is active.

$$\max \sum_{a \in A} c_a Y_a \tag{8a}$$

$$\sum_{e \in \delta(v)} x_e \le 1 \qquad v \in \mathcal{V} \tag{8b}$$

$$\sum_{e \in E[U]} x_e \le \frac{|U|-1}{2} \qquad U \in \mathcal{O}(\mathcal{V}) \tag{8c}$$

$$Y_{a'(e)} + Y_{a''(e)} \le x_e \qquad e \in \mathcal{E} \tag{8d}$$

$$\sum_{a \in \delta^+(v)} Y_a = X_v \qquad v \in \mathcal{V} \tag{8e}$$

$$Z_{av} \ge Y_a + X_v - 1 \qquad v \in \mathcal{V}, a \in \mathcal{A} \tag{8f}$$

$$Z_{av} \le Y_a, \ Z_{av} \le X_v \qquad v \in \mathcal{V}, a \in \mathcal{A} \tag{8g}$$

$$NY_a + \sum\nolimits_{v \in \mathcal{V} \setminus \{s(a)\}} P_{vt(a)} Z_{av} \le \frac{1}{\gamma} P_{s(a)t(a)} \qquad a \in \mathcal{A} \tag{8h}$$

$$Y \text{ binary}, x, X, Z \text{ continuous}, \quad x, Z \ge 0. \tag{8i}$$

Similarly, the matching inequalities can be added to the M-model by replacing (8f), (8g), (8h) in the ZC-model with (6); doing so results in the MC-model.

Generating matching inequalities

The number of matching inequalities (8c) is exponential in $|\mathcal{V}|$. Therefore, in the process of solving the linear relaxation of the ZC-model and the MC-model, we need to generate them on-the-fly. Below, following Pióro (2011), we describe how to generate the most violated matching inequality (8c) by means of a minimum odd-cut procedure. In the sequel, for a given undirected graph $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}))$ with the set of nodes $V(\mathcal{G})$ and the set of edges $E(\mathcal{G})$, $\delta_{\mathcal{G}}(v)$ will denote the set of all edges in $E(\mathcal{G})$ incident to node $v \in V(\mathcal{G})$. (We will skip subscript $\mathcal{G}$ only when it is clear what graph is considered). Besides, instead of a vector $x = (x_e, \ e \in E(\mathcal{G}))$ where $0 \le x_e \le 1, \ e \in E(\mathcal{G})$, we will make use of the notion of edge "capacity" function $x : E(\mathcal{G}) \to [0, 1]$. In fact, we will use the function x and the vector $x = (x_e, \ e \in E(\mathcal{G}))$ interchangeably. Let U be a subset of $V(\mathcal{G})$. Recall that a cut induced by a set U is defined as the set of all edges in $E(\mathcal{G})$ such that each edge in the cut has one end in U and the other in the complementary set $V(\mathcal{G}) \setminus U$. We denote such a set of edges, i.e., the cut, by $\delta_{\mathcal{G}}(U)$. For a given set of nodes T ($T \subseteq V(\mathcal{G})$) with an even number of elements, a cut $\delta_{\mathcal{G}}(U)$ is called a T-cut if $|T \cap U|$ is an odd number. As before, $E_{\mathcal{G}}[U]$ will denote the set of all edges of $E(\mathcal{G})$ with both ends in U. Finally, for any $F \subseteq E(\mathcal{G})$, $x(F) = \sum_{e \in F} x(e)$.

Given a graph $\mathcal{G}$ and a capacity function x, the most violated matching inequality is defined by any odd-set U^* that maximizes the quantity

$$C(U) = x(E[U]) - \frac{|U|-1}{2} \tag{9}$$

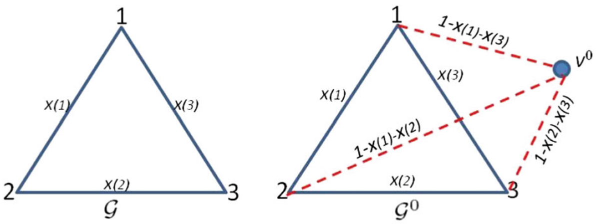


Fig. 1 Graphs $\mathcal{G}$ and $\mathcal{G}^0$

over all odd-sets $U \in \mathcal{O}(V(\mathcal{G}))$. If $C(U^*) > 0$, then the inequality

$$x(E[U^*]) \leq \frac{|U^*| - 1}{2} \tag{10}$$

defines the most violated matching inequality for the current solution x of the linear relaxation of (8). Otherwise, if $C(U^*) \leq 0$, then x is an element of the matching polytope (7).

Let us call the odd-set U^* maximizing the value of (9) the *maximum odd-set*. To give the basic result characterizing the maximum odd-set in an undirected graph $\mathcal{G}$ for a given edge capacity function x, we need to consider an extended graph $\mathcal{G}^0 = (V(\mathcal{G}^0), E(\mathcal{G}^0))$, which is obtained from graph $\mathcal{G}$ by adding one extra node v^0 and by linking the new node v^0 to each node in $V(\mathcal{G})$ by an extra edge. That is, $V(\mathcal{G}^0) = V(\mathcal{G}) \cup \{v^0\}$ and $E(\mathcal{G}^0) = E(\mathcal{G}) \cup \{\{v, v^0\} : v \in V(\mathcal{G})\}$. The edge capacity function x^0 in the extended graph is defined as

$$x^0(e) = \begin{cases} x(e) & \text{if } e \in E(\mathcal{G}) \\ 1 - x(\delta_{\mathcal{G}}(v)) & \text{if } e = \{v, v^0\} \text{ and } v \in V(\mathcal{G}) \end{cases} \tag{11}$$

Figure 1 shows an example of graph $\mathcal{G}$ and its extended graph $\mathcal{G}^0$.

In the following, we formalize the equivalence between maximum odd-sets and minimum T-cuts.

Proposition 1 *Define set T (composed of an even number of elements):*

$$T = \begin{cases} V(\mathcal{G}) & if\ |V(\mathcal{G})| \text{ is even} \\ V(\mathcal{G}) \cup \{v^0\} & if\ |V(\mathcal{G})| \text{ is odd} \end{cases} \tag{12}$$

A set U^* is a maximum odd-set with respect to x in graph $\mathcal{G}$ if, and only if, $\delta_{\mathcal{G}^0}(U^*)$ is a minimum T-cut with respect to x^0 in graph $\mathcal{G}^0$.

Proof By the construction of the extended graph $\mathcal{G}^0$, $x^0(\delta_{\mathcal{G}^0}(v)) = 1$ for all $v \in V(\mathcal{G})$. Hence, for any set of nodes U in $V(\mathcal{G})$ we have $|U| = \sum_{v \in U} x^0(\delta_{\mathcal{G}^0}(v))$, and therefore the equation

$$|U| = x^0(\delta_{\mathcal{G}^0}(U)) + 2x^0(E_{\mathcal{G}^0}[U]) \tag{13}$$

holds for any $U \subseteq V(\mathcal{G})$. Since $x^0(E_{\mathcal{G}^0}[U]) = x(E_{\mathcal{G}}[U])$, equation (13) directly implies that

$$C(U) = \frac{1}{2} - \frac{x^0(\delta_{\mathcal{G}^0}(U))}{2} \tag{14}$$

where, as specified in (9), $C(U) = x(E_{\mathcal{G}}[U]) - \frac{|U|-1}{2}$. Hence, for any subset $U \subseteq V(\mathcal{G})$, $C(U)$ is maximized when $x^0(\delta_{\mathcal{G}^0}(U))$ is minimized.

Now assume that $U \subseteq V(\mathcal{G})$ and $|U|$ is odd, i.e., $U \in \mathcal{O}(V(\mathcal{G}))$. It can be easily checked that a minimum cut in $(\mathcal{G}^0, x^0)$ of the form $\delta_{\mathcal{G}^0}(U)$ over all odd-sets $U \in \mathcal{O}(V(\mathcal{G}))$ is a minimum T-cut in $\mathcal{G}^0$. By (14), any such minimum T-cut $\delta_{\mathcal{G}^0}(U^*)$ defines a maximum odd-set U^* in the original graph $\mathcal{G}$. □

A minimum T-cut in a graph can be obtained from the Gomory–Hu tree (GH-Tree in short). Recall that a GH-tree for $(\mathcal{G}, x)$ is a tree $\mathcal{T} = (V(\mathcal{T}), E(\mathcal{T}))$ spanning the set of all nodes of graph $\mathcal{G}$ (i.e., $V(\mathcal{T}) = V(\mathcal{G})$) with an edge capacity function $u : E(\mathcal{T}) \rightarrow R_+$, such that for any two nodes $s, t \in V(\mathcal{G})$, the value of the minimum (with respect to x) cut separating s and t is equal to $u(f^0)$ where $f^0 = \text{argmin}\{u(f) : f \in \mathcal{P}_{st}\}$ where $\mathcal{P}_{st}$ is the unique path between nodes s and t in $\mathcal{T}$, see Korte and Vygen (2006) and Schrijver (2003). Moreover, the minimum cut in question is defined by the two connected components, obtained from $\mathcal{T}$ by deleting edge f^0. Such a cut is called a *fundamental cut*. By the Padberg–Rao theorem (see Korte and Vygen 2006, Theorem 12.17) the set of all fundamental cuts of the GH-tree contains a minimum T-cut for any even set of nodes $T \subseteq V(\mathcal{G})$. Hence, a maximum odd-set for graph $\mathcal{G}$ can be easily found from a GH-tree of graph $\mathcal{G}^0$.

For a graph with n nodes and m edges, algorithms for constructing a GH-tree run in time $\mathcal{O}(n \cdot t)$ where t is the time to perform one max-flow computation between two nodes in the graph. Depending on the algorithm, t is $\mathcal{O}(n \cdot m^2)$ (Edmonds and Karp 1972), $\mathcal{O}(n^2\sqrt{m})$ (Goldberg and Tarjan 1988), or even better (see Chapter 8 in Korte and Vygen 2006). Hence, the running time to construct a GH-tree is not worse than $\mathcal{O}(n^3\sqrt{m})$.

Numerical efficiency of the Gomory–Hu algorithm and its modification due to (Gusfield 1999) were investigated by Goldberg and Tsioutsiouliklis (1999). They devised efficient implementations of both algorithms, providing them with heuristics to speed them up (Tsioutsiouliklis 1999). Their general conclusion is that the original Gomory–Hu algorithm outperforms the Gusfield version, although the former is more complicated to implement.

Let LR-ZC denote the linear relaxation of the ZC-model, i.e., formulation (8) with variables Y relaxed. Algorithm 1 gives the procedure for solving LR-ZC through minimum odd-cut generation. Below, T is defined by (12), and LR-ZC($\mathcal{U}$) denotes a restricted version of LR-ZC with constraint (8c) substituted by

$$\sum_{e \in E[U]} x_e \leq \frac{|U|-1}{2} \quad U \in \mathcal{U} \tag{15}$$

for a given family of odd-sets $\mathcal{U} \subseteq \mathcal{O}(V(\mathcal{G}))$.

Algorithm 1 solving linear relaxation LR-ZC

Step 1: $\mathcal{U} := \emptyset$.
Step 2: Solve LR-$ZC(\mathcal{U})$. Denote the obtained optimal solution by x.
Step 3: Construct $(\mathcal{G}^0, x^0)$ from $(\mathcal{G}, x)$.
Step 4: Construct a GH-tree for $(\mathcal{G}^0, x^0)$ and find a minimum T-cut $\delta_{\mathcal{G}^0}(U)$ $(U \subseteq V(\mathcal{G}))$ among its fundamental cuts.
Step 5: If $x^0(\delta_{\mathcal{G}^0}(U)) \geq 1$ then STOP; LR-ZC has been solved. If $x^0(\delta_{\mathcal{G}^0}(U)) < 1$, $\mathcal{U} := \mathcal{U} \cup \{U\}$ and go to **Step 2**.

Algorithm 1 can also be applied to solve the linear relaxation of the MC-model (LR-MC) by replacing LR-ZC($\mathcal{U}$) with LR-MC($\mathcal{U}$), a restricted version of LR-MC.

It is important to note that, while generating maximum odd-sets, we can usually substantially decrease the problem size by deleting from the initial graph $\mathcal{G}$ all edges with $x_e = 0$ and with $x_e = 1$ (in the latter case we also delete the nodes incident to the edge). This is justified by the following result.

Proposition 2 *Assume x satisfies (7a) and (7c). Suppose $U \subseteq V(\mathcal{G})$ is an odd subset of nodes in graph $\mathcal{G}$. Consider graph $\mathcal{G}'$ obtained from $\mathcal{G}$ by deleting all edges with $x_e = 0$, and with $x_e = 1$ together with their end nodes. Then the set $U' \subseteq V(\mathcal{G}')$ corresponding to the set U in $V(\mathcal{G})$ maximizes $C(U') = x(E_{\mathcal{G}'}[U']) - \frac{|U'|-1}{2}$ in $\mathcal{G}'$ if, and only if, U maximizes $C(U) = x(E_{\mathcal{G}}[U]) - \frac{|U|-1}{2}$ in $\mathcal{G}$.*

Proof Clearly, deleting an edge with $x_e = 0$ does not change the value of $C(U)$. Deleting an edge $e \in E_{\mathcal{G}}[U]$ with $x_e = 1$ together with its end points does not change the value of $C(U)$ either, as then $x(E_{\mathcal{G}}[U])$ decreases by 1, and $|U|$ decreases by 2. □

Numerical study

In this section we compare the effectiveness of the four described MWCSP models, i.e., *M*, *Z*, MC, ZC, through computational experiments. The network instances were generated assuming the basic radio channel model for wireless networks discussed in Goldsmith (2005). The nodes are located randomly in a square area of 800 m by 800 m. The transmitting power for each node is set to $P_T = 1$ mW, and the noise power to $N = 10^{-13}$ W. The SINR threshold is $\gamma = 2.24$. The received power is calculated as $P_R = P_T \times d^{-4}$ where d is the distance from the potential transmitter to the potential receiver. For each pair of nodes, if $P_R \geq \gamma \times N$, i.e., the signal-to-noise ratio condition is satisfied, then the two arcs connecting the nodes can be used (if needed). The weights c for objective (3a) are generated randomly between 0 and 1.

We used the implementation of the GH-tree algorithm from Tsioutsiouliklis (1999), and embedded it into our procedure of generating matching inequalities, as described in Sect. 3. The algorithms were implemented in Python. The GH-tree algorithm was compiled to a dynamic-link library so the resulting function inside it could be called from Python. For the linear relaxations, the Gurobi solver was used. All the computations were executed on a Windows XP computer equipped with a dual core Intel 2.53 GHz CPU and 1.93 GB RAM.

A comparison of the quality of linear relaxations

Table 1 shows running times (in seconds) and optimality gaps for the linear relaxations (LR) of the four considered models. The optimality gap measures the quality of the upper bound of a LR and is defined as $\frac{\text{LRO}-\text{MIPO}}{\text{MIPO}} \times 100$, where MIPO denotes the optimal objective value of MWCSP, and LRO is the optimal objective value of its LR. In the table, the two columns named “cuts” give the number of matching inequalities generated in the solution process of the MC and ZC models. The Gurobi MIP solver was applied to compute the values for MIPO using, for computation time comparisons, both the M-model and the Z-model. It turned out that with Gurobi the M-model was on the average faster, but it did not systematically outperform the Z-model.

In Table 1 we examine five groups of networks that differ in the number of nodes. Each group contains five instances. The number of nodes varies from 20 to 60. This range is consistent with the scenarios considered in the literature (e.g., Björklund et al. 2004). We observe that the Z-model is superior (has smaller gaps) than the M-model. Adding matching inequalities improves both models. However, the MC-model is still worse not only than the ZC-model but also than the Z-model. The ZC-model gives the smallest gaps. The gap tends to grow, however, with the number of nodes $|\mathcal{V}|$, whereas within each network group of the same $|\mathcal{V}|$, there is no clear trend of the gap behaviour with respect to $|\mathcal{A}|$. The running time grows in $|\mathcal{V}|$ for all the models. For the M-model, however, the LR optimum is reached in a very short time for all the instances. Clearly, the more cuts generated, the longer the running time for the MC and ZC models—although the time for generating one cut is very short, the relaxations of the models must be solved for each new cut added.

Figure 2 depicts the average optimality gap for the six network groups with different numbers of nodes. The quality of the gap for the four models always obeys the following order: ZC (the best), Z, MC, M (the worst). Matching inequalities improve the M-model and the Z-model to a similar degree, and this improvement is not sensitive to the network size. A general observation is that the differences in the gap are not significant. Still, as we will soon see in Table 2, these differences are important for the efficiency of the considered models.

Figure 3 illustrates the effectiveness of the process of adding cuts to reach the LR optimality for the MC-model and the ZC-model. The presented curves correspond to one selected representative instance from each group. The horizontal axis shows the number of the generated cuts, and the vertical axis shows the normalized objective value, $\frac{\text{current LR objective}}{\text{optimal LR objective}}$. We note that adding cuts gives a significant effect on approaching the optimum of LR at the initial steps and this effect decreases substantially with the number of executed steps (i.e., with the number of generated cuts). This indicates that stopping adding cuts after several steps can potentially improve the efficiency of branch-and-cut.

The efficiency of solving the MIP models for MWCSP

To make comparisons of the potential of the methods fair, we have implemented our own branch-and-bound (B&B) algorithms BBM and BBZ for the M and Z models,

Table 1 Strength of the linear relaxations of the MWCSP models

Network		*M*-model		*Z*-model		MC-model			ZC-model		
$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Cuts	Gap (%)	Time (s)	Cuts
20	38	39.1	<0.1	33.7	0.1	38.4	<0.1	2	33.4	0.2	2
	48	57.9	<0.1	39.5	0.1	56.9	<0.1	3	38.6	0.5	3
	52	83.3	<0.1	73.1	0.1	75.2	<0.1	3	69.6	0.4	3
	56	82.2	<0.1	66.8	0.1	79.4	<0.1	6	65.6	0.6	4
	58	99.8	<0.1	77.2	0.2	92.3	<0.1	5	74.4	0.6	4
30	96	93.1	<0.1	86.3	0.5	90.3	0.2	7	83.1	3.1	6
	98	88.8	<0.1	80.4	0.4	87.1	0.1	4	79.8	1.8	4
	100	97.7	<0.1	89.0	0.4	92.5	0.3	8	85.6	2.3	6
	106	109.2	<0.1	97.2	0.5	103.1	0.3	10	91.4	3.5	6
	118	124.8	<0.1	116.9	0.5	124.8	0.2	3	116.7	1.6	3
40	162	77.0	<0.1	69.7	1.3	75.0	0.9	13	68.7	9.5	7
	168	103.7	<0.1	91.4	1.4	102.3	0.8	11	90.3	7.4	6
	180	96.5	0.1	83.2	1.4	89.2	0.8	14	78.1	16.8	10
	182	96.2	0.1	87.6	1.4	91.5	0.5	6	84.1	4.7	4
	194	82.0	0.1	74.8	1.7	75.6	1.0	19	71.7	20.8	19
50	256	90.7	0.2	81.7	3.9	88.3	1.5	14	71.1	61.0	15
	276	111.6	0.2	105.2	3.5	105.7	2.2	17	99.7	32.4	8
	286	130.8	0.2	118.2	8.8	128.9	1.4	13	116.1	41.3	9
	292	118.0	0.2	110.9	8.0	115.5	3.3	30	108.3	56.8	13
	298	104.9	0.2	94.5	4.3	104.2	1.7	18	93.8	26.1	5
60	414	123.0	0.3	114.4	13.5	120.7	4.0	20	112.8	109.2	9
	424	148.4	0.3	139.7	13.6	146.9	2.2	9	138.0	87.9	7
	428	103.3	0.3	97.1	21.9	101.9	4.7	16	96.3	235.7	13
	436	151.5	0.3	145.6	8.9	149.2	3.9	13	144.5	133.0	8
	456	122.9	0.4	114.3	14.1	121.4	2.7	9	113.4	191.2	14
Average		101.5	0.1	91.5	4.4	98.3	1.3	10.9	89.3	41.9	7.5

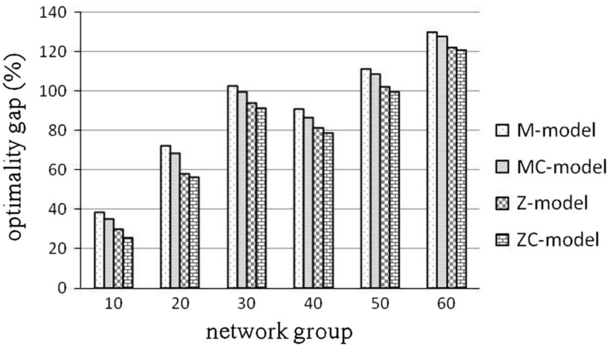


Fig. 2 Average gaps for network instances of different sizes

and branch-and-cut (B&C) algorithms BCM and BCZ for the MC and ZC models, respectively. Depth-first search is applied, and the branching rule is to use "the most fractional" (nearest to 0.5) variable. For B&C, the generated cuts are used globally. Each time we solve a B&B node, the generated cuts are added to a global list used for all the subsequent nodes.

Table 2 shows the running times and the number of processed B&B nodes of the algorithms, as well as the number of cuts for BCM and BCZ. Label "3h*" indicates that an optimum could not be reached in 3 h. As we can see, the M-model and the MC-model visit more B&B nodes than the other two models, due to their poor quality of linear relaxations. On the average, the ZC-model and the MC-model need less number of B&B nodes than the Z-model and the M-model, respectively. For some network instances, however, the B&B tree size for the ZC-model is larger than that for the Z-model. Similar cases are observed for the MC and M models. This happens typically when the matching inequalities improve the upper bound only marginally.

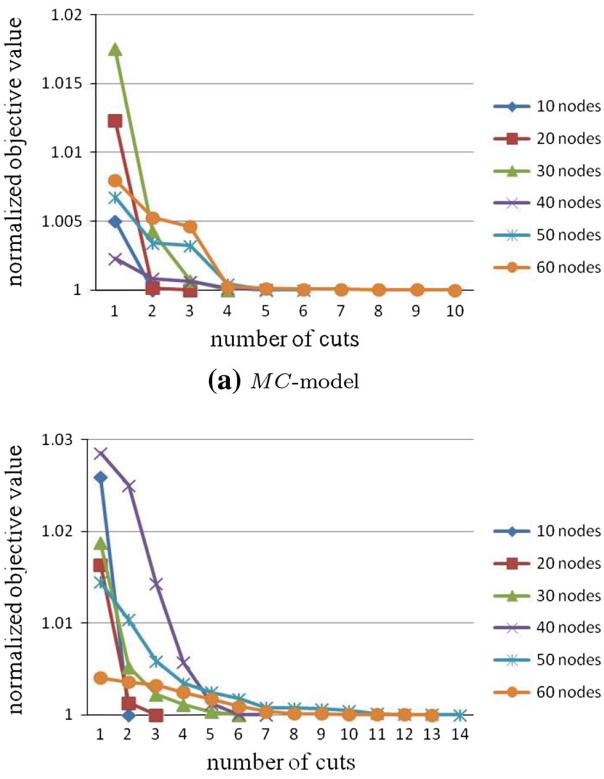
For small networks, when the time for solving one B&B node of the M-model is very short, it happens that the time for solving the M-model and the MC-model is shorter than that for the other two models. For large networks, however, solving the Z-model and the ZC-model requires shorter time in comparison with the other two models since they need much more B&B nodes to reach the optimum. It is interesting to observe that the ZC-model can find an optimum for network instances for which the other models are not able to do that in 3 h.

Conclusions

We have considered the important problem of maximizing the weighted revenue from parallel link transmissions in wireless networks. We have investigated two basic approaches of modeling the SINR constraint, and showed that the model with auxiliary variables eliminating bi-linearity provides better upper bound (optimality gap) than the "big M" model.

Table 2 The effectiveness of BBM, BCM, BBZ and BCZ

Network		BBM		BCM			BBZ		BCZ		
$\|\mathcal{V}\|$	$\|\mathcal{A}\|$	Time (s)	B&B	Time (s)	B&B	Cuts	Time (s)	B&B	Time (s)	B&B	Cuts
20	38	5	218	5	230	4	5	62	7	78	6
	48	11	452	14	387	12	17	141	11	84	9
	52	20	898	26	779	18	22	193	22	184	15
	56	26	1,142	36	941	9	29	220	24	175	7
	58	25	954	27	862	14	37	288	33	247	16
Average		17	732	21	639	11	22	180	19	153	10
25	60	173	5,263	132	3,695	14	212	1,430	249	1,206	16
	62	59	1,862	45	1,293	20	43	255	36	207	13
	68	84	2,393	140	2,420	19	95	461	50	251	10
	82	124	3,074	85	1,748	39	91	395	83	356	26
	84	235	4,853	171	2,918	53	167	716	160	635	44
Average		135	3,489	114	2,415	29	121	651	115	531	22
30	96	856	15,039	810	12,379	49	447	1,386	347	1,044	34
	98	666	10,667	385	5,399	50	407	1,274	377	1,175	30
	100	1,365	25,036	1,166	19,515	39	1,178	3,677	873	2,631	40
	106	721	11,518	761	9,762	54	301	867	318	810	33
	118	1,865	29,539	1,565	21,140	120	755	2,096	905	1,865	51
Average		1,094	18,360	937	13,639	63	618	1,860	564	1,505	38
35	114	1,311	23,485	1,419	17,996	45	971	1,938	2,019	532	31
	124	1,622	21,004	2,292	24,451	86	951	1,711	992	1,773	51
	130	11,619	133,828	9,652	84,871	170	7,767	15,093	5,622	10,990	104
	144	7,549	8,759	7,235	8,044	153	4,897	5,847	2,148	3,214	147
	152	5,838	66,667	6,451	60,283	216	2,892	4,510	1,942	2,841	128
Average		5,588	50,749	5,410	39,129	134	3,496	5,820	2,545	3,870	92
40	162	3h*	105,759	3h*	93,853	121	5,357	6,878	5,014	6,321	100
	168	3h*	95,775	3h*	89,282	116	7,208	8,917	5,147	6,035	104
	180	3h*	88,965	3h*	73,002	191	9,152	9,708	9,105	12,843	139
	182	3h*	96,185	3h*	76,330	130	3h*	13,888	10,463	11,756	157
	194	3h*	96,349	3h*	72,773	308	3h*	14,506	10,825	11,606	260
Average		3h*	96,607	3h*	81,048	173	12,463	11,406	8,111	9,085	152



(a) MC-model

(b) ZC-model

Fig. 3 The effect of cuts

Then, we have introduced matching inequalities to the models and developed a minimum odd-cut generation method based on the Gomory–Hu algorithm for generating the matching cuts. This improves the optimality gaps even further. It has been observed through our numerical study that although the gaps do not differ significantly between the models, these differences become profound in the branch-and-bound algorithm, making the model with the smallest gap perform visibly better than the others.

An extension of the current work would be modeling MWCSP for variable node transmission power to examine to what extent this new dimension of freedom can improve the number of parallel transmissions. Another extension would be the use of the improved models to calculate the compatible sets for different optimization problems in wireless communications considered previously.

Acknowledgments We would like to thank the reviewers for their thorough and detailed reviews. This work was supported by the EC Marie Curie Actions project MESH-WISE (FP7-PEOPLE-2012-IAPP-324515) and ELLIIT—Excellence Center at Linköping-Lund in Information Technology. The research of M. Pióro was also supported by National Science Centre (Poland) under Grant 2011/01/B/ST7/02967. Yuan Li was also supported by China Scholarship Council.

References

Andrews M, Dinitz M (2009) Maximizing capacity in arbitrary wireless networks in the SINR model: complexity and game theory. In: IEEE INFOCOM. pp 1332–1340
Björklund P, Värbrand P, Yuan D (2004) A column generation method for spatial TDMA scheduling in ad hoc networks. Ad Hoc Netw 2(4):405–418
Brar G, Blough D, Santi P (2006) Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In: ACM MOBICOM. pp 2–13
Capone A, Carello G, Filippini I, Gualandi S, Malucelli F (2010a) Routing, scheduling and channel assignment in wireless mesh networks: optimization models and algorithms. Ad Hoc Netw 8(6):545–563
Capone A, Carello G, Filippini I, Gualandi S, Malucelli F (2010b) Solving a resource allocation problem in wireless mesh networks: a comparison between a CP-based and a classical column generation. Networks 55(3):221–233
Capone A, Chen L, Gualandi S, Yuan D (2011) A new computational approach for maximum link activation in wireless networks under the SINR model. IEEE Trans. Wirel Commun. 10(5):1368–1372
Edmonds J, Karp R (1972) Theoretical improvements in algorithmic efficiency for network flow problems. J. ACM 19(2):248–264
Elbatt T, Ephremides A (2004) Joint scheduling and power control for wireless ad hoc networks. IEEE Trans Wirel Commun 3(1):74–85
Goldberg A, Tarjan R (1988) A new approach to the maximum flow problem. J ACM 35(4):921–940
Goldberg A, Tsioutsiouliklis K (1999) Cut tree algorithms. In: The 10th ACM-SIAM symposium on discrete algorithms (SODA). pp 376–385
Goldsmith A (2005) Wireless communicaitons. Cambridge University Press, Cambridge
Goussevskaia O, Oswald Y, Wattenhofer R (2007) Complexity in geometric SINR. In: ACM MOBIHOC. pp 100–109
Goussevskaia O, Wattenhofer R, Halldorsson M, Welzl E (2009) Capacity of arbitrary wireless networks. In: IEEE INFOCOM. pp 1872–1880
Gusfield D (1999) Very simple methods for all pairs network flow analysis. SIAM J Comput 19(1):143–155
Kesselheim T (2011) A constant-factor approximation for wireless capacity maximization with power control in the SINR model. In: The 22nd ACM-SIAM symposium on discrete algorithms (SODA). pp 1549–1559
Korte B, Vygen J (2006) Combinatorial optimization: theory and algorithms, 3rd edn. Springer, Berlin
Li Y, Ephremides A (2007) A joint scheduling, power control, and routing algorithm for ad hoc wireless networks. Ad Hoc Netw 5(7):959–973
Li Y, Pióro M, Yuan D, Su J (2012) On joint optimization of link rate assignment and transmission scheduling in wireless mesh networks. In: The 15th international telecommunications network strategy and planning symposium (Networks). pp 1–6
Pióro M (2011) A report on minimum odd-cut generation with applications to network design. Technical report, Institute of Telecommunications, Warsaw University of Technology
Pióro M, Żotkiewicz M, Staehle B, Staehle D, Yuan D (2011) On max-min fair flow optimization in wireless mesh networks. Ad Hoc Netw. doi:10.1016/j.adhoc.2011.05.003
Schrijver A (2003) Combinatorial optimization—polyhedra and efficiency. Springer, Berlin
Tsioutsiouliklis K (1999) Cut tree algorithms: an experimental study. http://www.cs.princeton.edu/kt/cut-tree/. Accessed 10 Oct 2012
Viswanathan H, Mukherjee S (2006) Throughput-range tradeoff of wireless mesh backhaul networks. IEEE J Sel Areas Commun 24(3):593–602
Xu X, Tang S (2009) A constant approximation algorithm for link scheduling in arbitrary networks under physical interference model. In: The 2nd ACM international workshop on foundations of wireless ad hoc and sensor networking and computing (FOWANC). pp 13–20