



## Fast Algorithms for Joint Power Control and Scheduling in Wireless Networks

Journal:	<i>IEEE Transactions on Wireless Communications</i>
Manuscript ID:	Paper-TW-Jun-09-0824
Manuscript Type:	Original Transactions Paper
Date Submitted by the Author:	03-Jun-2009
Complete List of Authors:	FU, LIQUN; The Chinese University of Hong Kong, Dept. of Information Engineering Liew, Soung; The Chinese University of Hong Kong, Dept. of Information Engineering Huang, Jianwei; The Chinese University of Hong Kong, Dept. of Information Engineering
Keyword:	Cross-layer design and optimization < Multimedia, Networks and Systems, Resource allocation and interference management < Multimedia, Networks and Systems

# Fast Algorithms for Joint Power Control and Scheduling in Wireless Networks

Liquan Fu, *Student Member, IEEE*, Soung Chang Liew, *Senior Member, IEEE*, and Jianwei Huang, *Member, IEEE*

## Abstract

This paper studies the problem of finding a minimum-length schedule of a power-controlled wireless network subject to traffic demands and SINR (signal-to-interference-plus-noise ratio) constraints. We propose a column generation based algorithm that finds the optimal schedules and transmit powers. The column generation method decomposes a complex linear optimization problem into a restricted master problem and a pricing problem. We develop a new formulation of the pricing problem using the Perron-Frobenius eigenvalue condition, which enables us to integrate link scheduling with power control in a single framework. This new formulation reduces the complexity of the pricing problem, and thus improves the overall efficiency of the column generation method significantly – for example, the average runtime is reduced by 99.86% in an 18-link network compared with the traditional column generation method. Furthermore, we propose a branch-and-price method that combines column generation with the branch-and-bound technique to tackle the integer constraints on time slot allocation. We develop a new branching rule in the branch-and-price method that maintains the size of the pricing problem after each branching. Our branch-and-price method can obtain optimal integer solutions efficiently – for example, the average runtime is reduced by 99.72% in 18-link networks compared with the traditional branch-and-price method. We further suggest efficient heuristic algorithms based on the structure of the optimal algorithms. Simulation results show that the heuristic algorithms can reach solutions within 16% of optimality in 10 seconds for a 32-link network.

## Index Terms

Scheduling, power control, SINR constraints, Column Generation, Branch-and-Price.

All the authors are with the Department of Information Engineering, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong. (email: {lqfu6, soung, jwhuang}@ie.cuhk.edu.hk.)

## I. INTRODUCTION

To avoid detrimental interference and boost throughputs, it is important to properly schedule the wireless links in the proximity of each other and control their transmission powers. In this paper, we investigate the joint scheduling and power control problem within the context of Spatial-reuse Time Division Multiple Access (STDMA) wireless networks. Consider a set of links with pre-allocated traffic demands (i.e., active time slots needed within each frame). STDMA scheduling assigns each link a set of time slots within the frame to meet its traffic demand. The system objective is to minimize the needed time slots such that all links' traffic demands are satisfied. We adopt the *physical interference model* [1], in which a receiver decodes its signal successfully if the signal-to-interference-plus-noise ratio (SINR) is above a certain threshold. Here the interference is the sum of the powers it receives from all concurrent transmitters other than its own. A good power control algorithm can improve the system performance. Through careful choices of transmit powers, we can mitigate the interference so that more wireless links can transmit simultaneously and thus require less total time needed in a frame.

STDMA scheduling has been extensively studied since 1980s (e.g., [2]–[13]). Only recent years have witnessed much research effort on cross-layer study of joint scheduling and power control (e.g., [10]–[16]). In [10], the authors proposed a simple heuristic of two alternating phases solution: A central controller first selects a set of valid links in a greedy way that eliminates strong interference in phase one, and then applies the power control algorithm based on [18] to find the minimal power solutions in phase two. Reference [11] formulated the joint scheduling and power control problem with fairness considerations and solved it using a serial linear programming rounding heuristic algorithm. However, it is difficult to evaluate the performance of these heuristic algorithms when optimal solutions are not known. Reference [13] examined the complexity of the joint power control and scheduling problem. The authors assumed that the allocated time to each link can be any real number and formulated the problem as a linear programming (LP) problem. They proved that this problem is at least as hard as the MAX-SIR-MATCHING problem, which they claimed is very difficult. The main difficulty in the LP problem is that the possible combinations of simultaneously active links grow exponentially with the number of links in the network. As a result, the LP problem has an exponential number of variables and is computationally intractable.

In this paper, we propose a column generation method to alleviate the above problem so that the optimal solutions could be found efficiently. The column generation method decomposes

the original problem into a master problem and a pricing problem. In each iteration, the pricing problem finds a better set of simultaneously active links only if the objective function can be further improved. Furthermore, we consider the realistic case where the number of time slots allocated to a link should be an integer instead of any real number. Building upon the column generation method, we further propose a branch-and-price method that combines the column generation method with the branch-and-bound method to provide optimal integer solutions. We note that column generation has previously been considered in [15]–[17]. In this work, we significantly improve the efficiency of both the column generation and the branch-and-price methods by exploiting the special structure of our problem. Our key contributions are as follows:

- 1) *Simplification of the Pricing Problem* : We integrate the Perron-Frobenius eigenvalue condition ( [18], [19]) into the formulation of the pricing problem. This integration eliminates the continuous variables and also reduces the number of the constraints, and thus reduces the computational complexity of solving the pricing problem.
- 2) *Efficient Algorithm to Solve the Pricing Problem*: We propose a Smart Enumerating (SE) algorithm that solves the pricing problem to optimality. Instead of relegating the pricing problem to a general optimization solver, we design smart search policies that eliminate the infeasible solutions and the non-optimal solutions in an efficient way.
- 3) *New Branching Rule that Controls the Pricing Problem Size*: We develop a new branching rule in the branch-and-price method. Our new branching rule maintains the size of the pricing problem after each branching, and thus improves the overall efficiency of the branch-and-price method.

In recent works using column generation method (e.g., [15]–[17]) , the pricing problem is formulated as a Mixed Integer Programming problem that contains both continuous and integer variables. Due to its high complexity, most of the runtime of the column generation method is spent on solving the pricing problem to optimality. Our new pricing problem formulation, together with the SE algorithm, reduce the runtime spent on the pricing problem. As a result, the efficiency of the column generation method can be significantly improved. Take an 18-link network as an example. Simulations show that using our new pricing problem formulation and the proposed SE algorithm, it takes only 2.031 seconds on average for the column generation method to find optimal solutions. However, using the column generation approach in [15]–[17], an average runtime of 1461.3 seconds is required. Our column generation method reduces the runtime by 99.86%.

For cases where reaching a close to optimal solution in a short time is more attractive than reaching the precise optimal solution, we propose a fast heuristic algorithm, Combined Sum Criterion Removal (CSCR), for the pricing problem. If we just use the CSCR algorithm to solve the pricing problem, this makes both the column generation and the branch-and-price methods fast heuristic algorithms. Simulations show that these heuristic algorithms can obtain near-optimal solutions with very short runtime.

The rest of this paper is organized as follows. In section II, we describe the wireless network model assumed. The joint power control and scheduling problem is formulated in section III. In section IV, we introduce the column generation method with emphasis on the new formulation of the pricing problem, and propose the Smart Enumerating (SE) algorithm to solve the pricing problem to optimality. In section V, we introduce the branch-and-price method and propose the new branching rule. In section VI, we discuss the column generation and branch-and-price based heuristics. The simulation results are shown in section VII. Section VIII concludes this paper.

## II. THE NETWORK MODEL

A wireless network consists of a set of directed links  $\mathcal{L} = \{l_i, 1 \leq i \leq |\mathcal{L}|\}$  with positive traffic demands on all links. Let  $T_i$  and  $R_i$  denote the transmitter and receiver of link  $l_i$ , respectively.  $T_i$  transmits with power  $p_i$ , which is upper-bounded by  $p_{\max}^i$ .

In general, the transmitter set  $\mathcal{T} = \{T_i, 1 \leq i \leq |\mathcal{L}|\}$  and the receiver set  $\mathcal{R} = \{R_i, 1 \leq i \leq |\mathcal{L}|\}$  can have nodes in common. We assume a simple receiver structure in which the following primary constraints must be satisfied:

*Definition 1 (Primary constraints):* A node can not transmit and receive simultaneously. Furthermore, a node is not allowed to transmit to or receive from more than one node simultaneously.

A set of links which satisfies the primary constraints constitutes a matching:

*Definition 2 (Matching):* A matching  $\mathcal{M} \subseteq \mathcal{L}$  is a subset of the link set  $\mathcal{L}$  such that no two links in  $\mathcal{M}$  share the same node, i.e., if links  $l_i, l_j \in \mathcal{M}$ , then the nodes  $T_i, R_i, T_j, R_j$  are distinct.

Let  $\mathcal{T}_{\mathcal{M}}$  and  $\mathcal{R}_{\mathcal{M}}$  denote the set of transmitting nodes and the set of receiving nodes of the links in matching  $\mathcal{M}$ , respectively. Thus sets  $\mathcal{T}_{\mathcal{M}}$  and  $\mathcal{R}_{\mathcal{M}}$  are disjoint node sets.

Besides the primary constraints, wireless links that are simultaneously active should satisfy the signal to interference-plus-noise ratio (SINR) constraints at the receiving nodes<sup>1</sup>. A matching that satisfies the SINR constraints is a feasible matching:

<sup>1</sup>Here we assume fixed coding and modulation schemes.

*Definition 3 (Feasible matching):* A matching  $\mathcal{M}$  is feasible if there exists a positive power vector  $\mathbf{p} = (p_i : \forall i \text{ s.t. } l_i \in \mathcal{M})^T$  such that the SINR constraints at the receiver nodes in the set  $\mathcal{R}_{\mathcal{M}}$  are satisfied, i.e.,

$$\frac{p_i G(T_i, R_i)}{\eta_i + \sum_{l_j \in \mathcal{M}, j \neq i} p_j G(T_j, R_i)} \geq \gamma_i, \quad \forall l_i \in \mathcal{M}, \quad (1)$$

where  $G(T_j, R_i)$  is the channel gain from  $T_j$  to  $R_i$ ,  $\eta_i$  is the average noise power at  $R_i$ , and  $\gamma_i$  is link-dependent threshold depending on various considerations such as the desired bit error rate and the modulation schemes.

*Proposition 1* ([25]): Any subset of a feasible matching is also feasible. Any superset of an infeasible matching is also infeasible.

*Definition 4 (Maximal feasible matching):* A feasible matching is maximal if it is not a strict subset of any other feasible matching.

The key notations of this paper are listed in Table I. We use lowercase boldface symbols, e.g.,  $\mathbf{p}$ , to denote vectors, with  $p_i$  denoting the  $i$ th component. We use uppercase boldface symbols, e.g.,  $\mathbf{Q}$ , to denote matrices, with  $q_{ij}$  denoting the  $(i, j)$ th component and  $Q_j$  denoting the  $j$ th column. We use calligraphic symbols, e.g.,  $\mathcal{L}$ , to denote sets. The vector inequalities denoted by  $\succeq$  and  $\preceq$  are component-wise inequalities.

### III. PROBLEM FORMULATION

We consider a time-slotted system with a fixed frame length  $U$ . Each link in set  $\mathcal{L}$  has a fixed traffic demand in a frame, representing a fixed average rate requirement from the corresponding upper-layer applications. Each link  $l_i \in \mathcal{L}$  requires a throughput  $th_i$ . Let  $f_i$  denote the required active airtime of link  $l_i$  in a frame. The achievable data rate  $r_i$  per unit time of link  $l_i$  depends on the SINR threshold  $\gamma_i$  at its receiver  $R_i$ . The relation between  $th_i$  and  $f_i$  is  $th_i = r_i \frac{f_i}{U}$ . Therefore, the throughput demand  $th_i$  is equal to the active airtime demand  $f_i = \frac{th_i \cdot U}{r_i}$  in each frame.

Our focus is to minimize the length of airtime  $W$  needed to satisfy all the traffic demands in each frame  $U$  by using maximum feasible matchings, so that the maximum airtime  $U - W$  can be left for other traffic (e.g., best-effort traffic). A minimum value of  $W$  greater than  $U$  indicates that the total traffic demands (and the corresponding average rate requirements) exceed the system capacity and can not be satisfied.

Now let us formulate the optimization problem formally. Let  $\mathcal{E} = \{\mathcal{E}_k : 1 \leq k \leq |\mathcal{E}|\}$  denote the set of all the feasible matchings of link set  $\mathcal{L}$ . The transmit power vector of the feasible

matching  $\mathcal{E}_k$  is denoted by  $\mathbf{p}_k$ . We can also use an  $|\mathcal{L}| \times |\mathcal{E}|$  incidence matrix  $\mathbf{Q}$  to represent  $\mathcal{E}$ , where the element  $q_{ik}$  of  $\mathbf{Q}$  is

$$q_{ik} = \begin{cases} 1, & \text{if link } l_i \text{ is in the feasible matching } \mathcal{E}_k, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

We denote the  $k$ th column in  $\mathbf{Q}$  by  $Q_k, 1 \leq k \leq |\mathcal{E}|$ , which represents a feasible matching  $\mathcal{E}_k$ .

Our objective is to minimize the airtime length  $W$  required. If the incidence matrix  $\mathbf{Q}$  and all the associated transmit power vectors can be obtained a priori, the joint power control and minimum-airtime scheduling problem can be formulated as follows:

$$\begin{aligned} & \text{minimize} && W = \mathbf{e}^T \mathbf{u} \\ & \text{subject to} && \mathbf{Q}\mathbf{u} \succeq \mathbf{f}, \\ & \text{variables} && \mathbf{u} \succeq 0, \end{aligned} \quad (\text{P1})$$

where  $\mathbf{e}$  is the  $|\mathcal{E}| \times 1$  all-one vector and  $\mathbf{f} = (f_1, f_2, \dots, f_{|\mathcal{L}|})^T$ . The variables  $\mathbf{u} = (u_k : 1 \leq k \leq |\mathcal{E}|)^T$  indicate the chosen feasible matchings to be scheduled in the frame. In particular, each variable  $u_k$  denotes the airtime allocated to the feasible matching  $\mathcal{E}_k$  in the frame.

Problem (P1) is a Linear Program (LP). In a real wireless system, a slot is the smallest time unit in the time allocation process. Thus, the joint power control and scheduling problem should be formulated with additional integer constraints on the variables  $\mathbf{u} = (u_k : 1 \leq k \leq |\mathcal{E}|)^T$ . We denote (P1) with integer constraints on  $\mathbf{u}$  by  $(\text{P1})_{\text{INT}}$ , which is an Integer Linear Program (ILP).

There are two difficulties in (P1) and  $(\text{P1})_{\text{INT}}$ . First, there is no known polynomial-time algorithm for finding all the feasible matchings and the corresponding transmit power vectors. Thus, completely characterizing the coefficient matrix  $\mathbf{Q}$  in advance is difficult. Second, even if all the feasible matchings could be found, the size of set  $\mathcal{E}$  can be huge and in general increases exponentially with the number of links [13]. This means that (P1) and  $(\text{P1})_{\text{INT}}$  can be too large to be tackled directly. We circumvent these difficulties using iterative methods that generate feasible matchings on-the-fly rather than a priori. Specifically, we propose a column generation method to solve (P1). Building upon it, we use a branch-and-price method to solve  $(\text{P1})_{\text{INT}}$ .

The column generation method and the branch-and-price method are efficient techniques for solving large LP and ILP that have (exponentially) many variables. They have been applied to a wide variety of problems [23]. However, there are two fundamental difficulties when applying the column generation and the branch-and-price methods to solve (P1) and  $(\text{P1})_{\text{INT}}$ :

- 1) In the column generation method, the pricing problem contains both binary variables and continuous variables. In particular, the pricing problem is a Mixed-Integer Programming

(MIP) which is difficult to solve.

- 2) In the branch-and-price method, applying the conventional branching rule that branches on a single variable causes a new constraint to be added at each branch. As a result, a dual variable is added to the corresponding pricing problem. The pricing problem becomes progressively more complex as we go down the branching tree.

The following sections go into the details of the column generation and branch-and-price methods. In particular, we will present details on how the above difficulties arise and how we can overcome them.

#### IV. COLUMN GENERATION METHOD

In this section, we propose a Column Generation (CG) method to solve (P1). The column generation method decomposes (P1) into two sub-problems – a Restricted Master Problem (RMP) and a Pricing Problem (PP) – and solves them iteratively. The flowchart of the column generation method is shown in Fig. 1.

##### A. Restricted Master Problem

The Restricted Master Problem (RMP) is similar to the original problem (P1) except that only a subset feasible matchings  $\mathcal{E}' \subseteq \mathcal{E}$  is considered. Let  $\mathbf{Q}'$  denote the incidence matrix of  $\mathcal{E}'$ . The restricted master problem is as follows:

$$\begin{aligned}
 & \text{minimize} && W = \mathbf{e}^T \mathbf{u} \\
 & \text{subject to} && \mathbf{Q}' \mathbf{u} \succeq \mathbf{f}, \\
 & \text{variables} && \mathbf{u} \succeq 0.
 \end{aligned} \tag{RMP}$$

An initial subset of feasible matchings  $\mathcal{E}'$  can be easily formed by letting the  $i$ th matching consist of link  $l_i$  only ( $1 \leq i \leq |\mathcal{L}|$ ) and none of the other links. Since there is only one link in each matching, all these  $|\mathcal{L}|$  matchings are feasible matchings. The corresponding initial incidence matrix  $\mathbf{Q}'$  is an identity matrix.

First we solve (RMP) to optimality. We can achieve this with the Simplex Method [22] and obtain a primal optimal solution  $\mathbf{u}^*$  and a dual optimal solution  $\boldsymbol{\omega}^*$ . Since we only consider a subset of feasible matchings  $\mathcal{E}'$ ,  $\mathbf{u}^*$  may not be the optimal solution to the original problem (P1).

Consider the columns that are in the matrix  $\mathbf{Q}$  but not in matrix  $\mathbf{Q}'$ . Notice that in (P1), the cost coefficient of every variable in the objective function is 1. The reduced cost of a column  $Q_k$  in the matrix  $\mathbf{Q}$  but not in  $\mathbf{Q}'$  is defined as [22]:

$$\sigma_k = 1 - (\boldsymbol{\omega}^*)^T Q_k. \tag{3}$$

The reduced cost  $\sigma_k$  of column  $Q_k$  is the amount by which the objective function will improve if the corresponding variable  $u_k$  is assumed to be a positive value. If the reduced cost  $\sigma_k$  is less than 0, the objective function of (P1) can be further reduced. On the other hand, if the reduced costs of all the columns that are in the matrix  $Q$  but not in  $Q'$  are all non-negative, this means the current solution  $\mathbf{u}^*$  is the optimal solution to (P1).

In the column generation method, instead of computing the reduced costs for all the columns that are in the matrix  $Q$  but not in  $Q'$ , we consider the problem of minimizing the reduced cost  $\sigma = 1 - (\boldsymbol{\omega}^*)^T Q$ , which is equivalent to maximizing  $(\boldsymbol{\omega}^*)^T Q$ , subject to the constraints that ensure  $Q$  is a feasible matching. This optimization problem is called the Pricing Problem, which is discussed in the next subsection.

### B. Pricing Problem

The Pricing Problem (PP) is to find a feasible matching  $Q$  that maximizes  $(\boldsymbol{\omega}^*)^T Q$ , which is equivalent to minimizing the reduced cost  $1 - (\boldsymbol{\omega}^*)^T Q$ . It can be formulated as follows:

$$\begin{aligned}
 & \text{maximize} && \sum_{i=1}^{|\mathcal{L}|} \omega_i^* q_i \\
 & \text{subject to} && t_{num} + r_{num} = 2 \cdot \sum_{i=1}^{|\mathcal{L}|} q_i, && 1 \leq i \leq |\mathcal{L}|, \\
 & && \frac{p_i \cdot G(T_i, R_i)}{\eta_i + \sum_{l_j \in \mathcal{L} \setminus l_i} p_j \cdot G(T_j, R_i)} \geq q_i \cdot \gamma_i, && 1 \leq i \leq |\mathcal{L}|, && \text{(PP)} \\
 & && p_i \leq q_i \cdot P_{\max}^i, && 1 \leq i \leq |\mathcal{L}|, \\
 & \text{variables} && p_i \geq 0, && 1 \leq i \leq |\mathcal{L}|, \\
 & && q_i \in \{0, 1\}, && 1 \leq i \leq |\mathcal{L}|.
 \end{aligned}$$

The coefficient  $\boldsymbol{\omega}^* = \{\omega_i^* : 1 \leq i \leq |\mathcal{L}|\}^T$  in the objective function of (PP) is the optimal dual solution of the (RMP) problem in the current iteration. The variables in (PP) are the binary variables  $Q = \{q_i : 1 \leq i \leq |\mathcal{L}|\}$  and the continuous variables  $\{p_i : 1 \leq i \leq |\mathcal{L}|\}$ . Binary variable  $q_i$  is 1 if link  $l_i$  is active, and 0 otherwise. Continuous variable  $p_i$  is the transmission power of link  $l_i$ . In the first constraint of (PP),  $t_{num} = \left| \{T_i : \forall i \text{ s.t. } q_i = 1\} \right|$  and  $r_{num} = \left| \{R_i : \forall i \text{ s.t. } q_i = 1\} \right|$  are the numbers of transmitters and receivers of the active links, respectively. The first constraint in (PP) is the primary constraint (Definition 1) that ensures the active links form a matching. The second constraint guarantees that the SINR requirement at each active receiver is satisfied. The third constraint states that the transmission powers are limited.

If the optimal objective value of (PP) is greater than 1, the solution  $Q = \{q_i : 1 \leq i \leq |\mathcal{L}|\}$  is passed to (RMP) and the next iteration starts. If the optimal objective value of (PP) is less than or equal to 1, this means the optimal solution  $\mathbf{u}^*$  to the (RMP) problem in the current iteration is already the optimal solution to the whole problem (P1), and the column generation terminates.

(RMP) is a small-scale standard linear program that is easy to solve using the Simplex Method. However, (PP) is a Mixed Integer Program with both binary variables and continuous variables that is quite difficult to solve in general. Therefore, the overall runtime performance of the column generation method depends on how well we can solve the pricing problem (see simulation results in section VII).

### C. A New Formulation of the Pricing Problem

One of the key contributions of this paper is that we reformulate the pricing problem (PP) to reduce its complexity. The new formulation enables us to remove the continuous power variables  $\{p_i : 1 \leq i \leq |\mathcal{L}|\}$  and merge the primary constraints and the SINR constraints.

Consider the SINR constraints and the maximum transmit power constraints. Given a matching  $\mathcal{M} \subseteq \mathcal{L}$ , we define an  $|\mathcal{M}| \times |\mathcal{M}|$  nonnegative relative-channel-gain matrix  $\mathbf{B}_{\mathcal{M}}$  with entries as follows:

$$b_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \frac{G(T_j, R_i)}{G(T_i, R_i)}, & \text{if } i \neq j. \end{cases} \quad (4)$$

Let the vector  $\boldsymbol{\gamma}_{\mathcal{M}} = (\gamma_i : \forall i \text{ s.t. } l_i \in \mathcal{M})$  be the SINR thresholds in the receiving-node set  $\mathcal{R}_{\mathcal{M}}$  and the matrix  $\mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}})$  be the diagonal matrix whose diagonal entries are  $(\gamma_i : \forall i \text{ s.t. } l_i \in \mathcal{M})$ , respectively. The SINR requirements (1) can be written in matrix form as

$$(\mathbf{I} - \mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}}) \mathbf{p}_{\mathcal{M}} \succeq \mathbf{v}_{\mathcal{M}}, \quad (5)$$

where  $\mathbf{I}$  is an  $|\mathcal{M}| \times |\mathcal{M}|$  identity matrix and vector  $\mathbf{v}_{\mathcal{M}} = \left( \frac{\gamma_i \eta_i}{G(T_i, R_i)} : \forall i \text{ s.t. } l_i \in \mathcal{M} \right)^T$  is the normalized noise power vector.

Let  $\rho(\mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}})$  denote the largest real eigenvalue (also called Perron-Frobenius eigenvalue or spectral radius) of matrix  $\mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}}$ . By Perron-Frobenius Theorem [24],  $\rho(\mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}})$  is positive and the corresponding eigenvector is positive componentwise. Moreover, the problem of deciding whether a matching is feasible or not becomes the Perron-Frobenius eigenvalue condition of the nonnegative matrix  $\mathbf{D}(\boldsymbol{\gamma}_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}}$ . The following proposition is a compilation of the related propositions in [19]–[21], [24].

*Proposition 2:* Consider a matching  $\mathcal{M}$ . Assume that the transmit power vector  $\mathbf{p}_{\mathcal{M}}$  is upper bounded by vector  $\mathbf{p}_{\max} = (p_{\max}^i : \forall i \text{ s.t. } l_i \in \mathcal{M})^T$ . The necessary and sufficient conditions for the existence of a positive power vector  $\mathbf{p}_{\mathcal{M}} \preceq \mathbf{p}_{\max}$  that satisfies the SINR requirements in (5) are

$$\rho(\mathbf{D}(\gamma_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}}) < 1 \quad \text{and} \quad \mathbf{p}_{\mathcal{M}}^* \preceq \mathbf{p}_{\max}, \quad (6)$$

where

$$\mathbf{p}_{\mathcal{M}}^* = (\mathbf{I} - \mathbf{D}(\gamma_{\mathcal{M}}) \mathbf{B}_{\mathcal{M}})^{-1} \mathbf{v}_{\mathcal{M}} \quad (7)$$

is a componentwise minimum solution to (5).

Proposition 2 provides easy conditions to check the feasibility of a matching and a minimum solution to the transmit powers of transmitting nodes. In particular, given a matching, if conditions (6) are satisfied, the matching is guaranteed to be feasible, and the minimum transmit power vector can be set according to (7); otherwise, no matter how we tune the transmit powers, the links in the matching can not be active simultaneously. Incorporating (6) and (7) into the formulation of the pricing problem enables us to remove the power variables in the pricing problem. Notice that conditions (6) can only be applied to a set of links which already form a matching. On the other hand, we can define a proper version of the channel gain matrix such that conditions (6) take care of both the primary and SINR constraints in the pricing problem.

Consider a general wireless network with link set  $\mathcal{L} = \{l_i, 1 \leq i \leq |\mathcal{L}|\}$ . For any two different links  $l_i, l_j \in \mathcal{L}$ , where nodes  $T_i, R_i, T_j, R_j$  have at least one node in common, we re-define the channel gains  $G(T_i, R_j)$  and  $G(T_j, R_i)$  as:

$$G(T_i, R_j) = \infty \quad \text{and} \quad G(T_j, R_i) = \infty. \quad (8)$$

The other channel gains remain the same. The definitions in (8) allow us to extend the definition of the relative-channel-gain matrix  $\mathbf{B}_{\mathcal{M}}$  defined on a matching  $\mathcal{M}$  in (4) to  $\mathbf{B}_{\mathcal{S}}$ , which is defined on an arbitrary subset of links  $\mathcal{S} \subseteq \mathcal{L}$ . The matrix  $\mathbf{B}_{\mathcal{S}}$  is called virtual relative-channel-gain matrix with elements

$$b_{ij} = \begin{cases} 0, & \text{if } i = j, \\ \frac{G(T_j, R_i)}{G(T_i, R_i)}, & \text{if } i \neq j \text{ and nodes } T_i, R_i, T_j, R_j \text{ are distinct,} \\ \infty, & \text{others.} \end{cases} \quad (9)$$

Based on the definition of the virtual relative-channel-gain matrix  $\mathbf{B}_{\mathcal{S}}$ , condition  $\rho(\mathbf{D}(\gamma_{\mathcal{S}}) \mathbf{B}_{\mathcal{S}}) < 1$  will never hold if  $\mathcal{S}$  is not a matching. This means that the necessary and sufficient conditions

for a subset of links  $\mathcal{S} \subseteq \mathcal{L}$  to be a feasible matching are

$$\rho(\mathbf{D}(\gamma_{\mathcal{S}})\mathbf{B}_{\mathcal{S}}) < 1 \quad \text{and} \quad \mathbf{p}_{\mathcal{S}}^* \preceq \mathbf{p}_{\max}, \quad (10)$$

where

$$\mathbf{p}_{\mathcal{S}}^* = (\mathbf{I} - \mathbf{D}(\gamma_{\mathcal{S}})\mathbf{B}_{\mathcal{S}})^{-1} \mathbf{v}_{\mathcal{S}}. \quad (11)$$

We can now simplify the pricing problem as follows:

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^{|\mathcal{L}|} \omega_i^* q_i \\ & \text{subject to} && \rho(\mathbf{D}(\gamma_Q)\mathbf{B}_Q) < 1, \\ & && (\mathbf{I} - \mathbf{D}(\gamma_Q)\mathbf{B}_Q)^{-1} \mathbf{v}_Q \preceq \mathbf{p}_{\max}, \\ & \text{variables} && q_i \in \{0, 1\}, \quad 1 \leq i \leq |\mathcal{L}|. \end{aligned} \quad (\text{SPP})$$

In the Simplified Pricing Problem (SPP), the only variables are the binary variables  $Q = \{q_i : 1 \leq i \leq |\mathcal{L}|\}$ . The continuous power variables  $\{p_i : 1 \leq i \leq |\mathcal{L}|\}$  have been removed. Formulation (SPP) is a binary integer programming problem which is easier to solve than the original pricing problem (PP).

#### D. Finding the optimal solution to the pricing problem

We propose a Smart Enumerating (SE) algorithm that finds an optimal solution to (SPP). With the help of conditions (10), the SE algorithm can reduce the search space by eliminating the infeasible solutions and the non-optimal solutions in an efficient way. The SE algorithm takes much less time than the naive exhaustive search among all the subsets of link set  $\mathcal{L}$ .

In SE, we first solve (SPP) without considering the constraint set and obtain the corresponding optimal solution  $Q_{wc} = \{q_i : 1 \leq i \leq |\mathcal{L}|\}$ .  $Q_{wc}$  is a set of links that maximizes  $\sum_{i=1}^{|\mathcal{L}|} \omega_i^* q_i$  only and can be easily found: we set  $q_i = 1$  if the corresponding coefficient  $\omega_i^* > 0$  and set  $q_i = 0$  if  $\omega_i^* \leq 0$ .

Next we check if conditions in (10) are satisfied by  $Q_{wc}$ . If yes, then the active links in  $Q_{wc}$  form a feasible matching and the pricing problem is solved. Otherwise, the optimal solution  $Q^*$  to (SPP) must be a subset of  $Q_{wc}$ . The SE algorithm performs a search among a small number of the subsets of  $Q_{wc}$  using the following two criteria:

Criterion 1: If a subset  $Q_s$  of  $Q_{wc}$  is infeasible, all the supersets of  $Q_s$  need not be considered.

Criterion 2: If a subset  $Q_s$  of  $Q_{wc}$  is feasible, all the subsets of  $Q_s$  need not be considered.

Criterion 1 is based on Proposition 1. In Criterion 2, any subset of  $Q_s$  will only have an objective value less than  $Q_s$ 's objective value. With these two criteria, a large number of subsets of  $Q_{wc}$  can be eliminated.

SE finds an optimal solution to the binary integer programming (SPP). In general, it still has exponential runtime. However, because of the reduction in the search space, the SE algorithm is reasonably efficient for modest size networks.

## V. BRANCH-AND-PRICE METHOD

In general, the optimal solution to (P1) may be fractional (i.e., non-integer) and hence not feasible to  $(P1)_{INT}$ . In this section, we propose a branch-and-price method to address  $(P1)_{INT}$ .

### A. Branch-and-Price

The branch-and-price method adds a branch-and-bound algorithm to the column generation algorithm discussed thus far. The original column generation algorithm solves the linear relaxation of the ILP problem at each node in the branch-and-bound tree. The flowchart of the branch-and-price method is shown in Fig. 2. The branch-and-price method starts with the original ILP problem  $(P1)_{INT}$  as the root node. We first apply the column generation method to solve the linear relaxation (P1) to optimality and obtain an optimal solution  $\mathbf{u}^*$ . As discussed before, the optimal solution  $\mathbf{u}^*$  may not be integral, so we need to perform branching by adding constraints. A typical branching rule is to add constraints that cut off the fractional value in the current optimal solution  $\mathbf{u}^*$ . Assume a variable  $u_k^*$  in  $\mathbf{u}^*$  takes on a fractional value  $\alpha$ . Let  $\lfloor \alpha \rfloor$  and  $\lceil \alpha \rceil$  denote the largest integer not greater than  $\alpha$  and the smallest integer not less than  $\alpha$ , respectively. The original problem  $(P1)_{INT}$  (parent node) then branches to two sub-problems (child nodes) by adding the constraint

$$u_k \leq \lfloor \alpha \rfloor, \quad (12)$$

and the constraint

$$u_k \geq \lceil \alpha \rceil, \quad (13)$$

respectively. After branching, each child node is a new ILP problem. We then apply column generation to solve the LP relaxation to each child node. When the optimal solution to the new LP relaxation at a child node is again fractional, branching continues from that node.

In the branch-and-price method, the lower and upper bounds on the objective value at each node are obtained. The lower bound of a node is obtained by solving the relaxation of that node

with column generation. An integer feasible solution of a node can be obtained by rounding up the optimal linear relaxation solution to the nearest integer. This rounding establishes an upper bound on the objective value. We maintain the value  $C$ , which is the objective value of the best integer solution across all nodes so far. If the lower bound for some node is greater than or equal to  $C$ , this node need not be considered further (i.e., this node is pruned from the branch-and-bound tree). Nodes can also be pruned when the problem in a node is infeasible. The branch-and-price terminates when all the nodes in the branch-and-bound tree have been evaluated and the optimal integer solution to  $(P1)_{\text{INT}}$  is found.

### B. Improving Pricing Rules

A component that is critical to the performance of the branch-and-price method is the branching rule. The conventional branching rule (i.e., (12) and (13)) adds a new constraint to each of the sub-problems. This causes a new dual variable to be added to the pricing problem. At depth  $H$  in the branch-and-bound tree, there will be  $H$  additional dual variables. The size of the pricing problem will increase progressively as we go down the branch-and-bound tree. Another key contribution of this paper is that we develop a more efficient new branching rule to overcome this problem.

We first modify the formulation  $(P1)_{\text{INT}}$  to incorporate an upper bound,  $g_i$ , on the number of time slots allocated to each link  $l_i$  in set  $\mathcal{L}$ . The upper bounds for all links are represented by the vector  $\mathbf{g} = (g_i : 1 \leq i \leq |\mathcal{L}|)^T$ . Initially,  $g_i$  simply takes on the value of the total frame length  $U$ . The modified formulation is

$$\begin{aligned} & \text{minimize} && W = \mathbf{e}^T \mathbf{u} \\ & \text{subject to} && \mathbf{f} \preceq \mathbf{Q}\mathbf{u} \preceq \mathbf{g}, \\ & \text{variables} && \mathbf{u} \succeq 0 \text{ and integer.} \end{aligned} \tag{14}$$

Instead of adding a constraint, the new branching rule changes the upper bound or the lower bound. If the optimal solution  $\mathbf{u}^*$  has fractional elements, we calculate the vector  $\mathbf{h} = \mathbf{Q}\mathbf{u}^*$ . Suppose that an element  $h_i$  of  $\mathbf{h}$  is fractional. We create two branches, one with

$$\sum_k q_{ik} u_k \leq \lfloor h_i \rfloor, \tag{15}$$

and the other with

$$\sum_k q_{ik} u_k \geq \lceil h_i \rceil. \tag{16}$$

These two branching inequalities can be carried out by changing the corresponding upper bound  $g_i$  to  $\lfloor h_i \rfloor$  on one branch and changing the corresponding lower bound  $f_i$  to  $\lceil h_i \rceil$  on the other branch, respectively. The new branching rule, (15) and (16), successfully avoids adding an additional constraint into the sub-problem, and thus avoids adding a new dual variable to the corresponding pricing problem.

Branching rule, (15) and (16), is not applicable to the particular situation in which the optimal solution  $\mathbf{u}^*$  is fractional, but all the elements in the vector  $\mathbf{h} = \mathbf{Q}\mathbf{u}^*$  are integers. In this case, we revert to the conventional branching rule (12) and (13).

## VI. COLUMN GENERATION AND BRANCH-AND-PRICE BASED HEURISTIC

The column generation method and the branch-and-price method can find optimal solutions to the joint power control and link scheduling problem (P1) and (P1)<sub>INT</sub>, respectively. However, for large networks, both methods require a long computation time. This is due to the high complexity of the pricing problem. In this section, we focus on efficient heuristic algorithms that yield near-optimal solutions with much faster speeds. Our heuristics are built on the foundations of the column generation and the branch-and-price methods.

### A. Heuristic Pricing Problem Solution

Designing an efficient heuristic algorithm for the pricing problem can significantly reduce the complexities of the column generation and the branch-and-price methods. Even if we aim for an optimal solution, such a heuristic algorithm will still help, as explained in the following. Specifically, to obtain an optimal solution for the overall problem, it is not necessary to solve the pricing problem optimally in each iteration – it is sufficient to obtain any feasible matching with an objective value over 1. That is, as long as the new column generated can reduce the cost, it will do, and there is no need to find the best new column to be included. Thus, we can use a heuristic to find a cost-reducing column. Only when the heuristic fails to identify a cost-reducing column is it necessary for us to resort to the SE algorithm to see if a cost-reducing column exists.

If short runtime is more of a concern than solution optimality, the heuristic is also useful for finding a near-optimal solution. In this case, we do not revert to the SE algorithm even if the heuristic fails to find a cost-reducing column. If we adopt this strategy, the column generation method and the branch-and-price method become fast heuristic algorithms for identifying near-optimal solutions for (P1) and (P1)<sub>INT</sub>, respectively.

We next propose a Combined Sum Criterion Removal (CSCR) heuristic algorithm for the pricing problem. The key idea behind CSCR is to remove one link at a time until the remaining active links form a feasible matching. The flowchart of CSCR is shown in Fig. 3.

We first solve (SPP) without considering the constraint set. This step has been discussed in section IV-D. If the resulting  $Q$  is feasible as per the conditions in (10), then the pricing problem is solved. On the other hand, if  $Q$  is infeasible, we deactivate one link at a time until the remaining links form a feasible matching. There are two possible causes of the infeasibility: (i)  $\rho(\mathbf{D}(\gamma_Q)\mathbf{B}_Q) \geq 1$ ; or (ii)  $\rho(\mathbf{D}(\gamma_Q)\mathbf{B}_Q) < 1$ , but some elements in the component-wise minimum power vector  $\mathbf{p}_Q^*$  are greater than the maximum transmit powers allowed. For (i), we use CSCR to remove one link at a time. In CSCR, we set  $q_{\tilde{i}} = 0$  for a link  $l_{\tilde{i}}$  that achieves the maximum of the row sums and the column sums of matrix  $\mathbf{D}(\gamma_Q)\mathbf{B}_Q$  :

$$\tilde{i} = \arg \max_i \left\{ \sum_j (\mathbf{D}(\gamma_Q)\mathbf{B}_Q)_{ij}, \sum_k (\mathbf{D}(\gamma_Q)\mathbf{B}_Q)_{ki} \right\}. \quad (17)$$

The combined sum criterion comes from [18] which investigated the power control problem in cellular systems. The combined sum criterion seeks to minimize the upper bound of the Perron-Frobenius eigenvalue  $\rho(\mathbf{D}(\gamma_Q)\mathbf{B}_Q)$  assuming that only one link can be removed, so that the condition  $\rho(\mathbf{D}(\gamma_Q)\mathbf{B}_Q) < 1$  can be satisfied with a high probability. The physical meaning of the combined sum criterion is that, we remove the link that generates or experiences the maximum interference in the network. For (ii), we also remove one link at a time, but each time we remove the link whose transmit power  $p_i^*$  exceeds  $p_{\max}^i$  by the maximum amount.

The CSCR heuristic algorithm is simple and computationally efficient. The simulation results in section VII show that it also works well in practice.

### B. Initial Feasible Matchings

For initializing our algorithms, we need a subset of feasible matchings. Section IV-A shows one possibility where the initial feasible matchings for (RMP) are those that contain only one active link. The corresponding initial incidence matrix  $\mathbf{Q}'$  is an identity matrix. This choice, although simple, is not necessarily the best choice. A better choice could reduce the number of iterations required in column generation. We propose a heuristic algorithm, referred to as Increasing Demand Greedy Scheduling (IDGS), that solves  $(P1)_{\text{INT}}$  heuristically to obtain a good initial subset of feasible matchings. In particular, the IDGS produces a subset of *maximal* feasible matchings which will be used by the optimal solution with high probability.

IDGS is described in Algorithm 1. The inputs are the virtual relative-channel-gain matrix  $\mathbf{B}_{\mathcal{L}}$  (as defined in Section IV-C) of the link set  $\mathcal{L}$ , the traffic demand vector  $\mathbf{f}$ , the SINR requirement vector  $\boldsymbol{\gamma}$ , and the normalized noise vector  $\mathbf{v}$ . The outputs of the IDGS algorithm are the set of maximal feasible matchings  $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_K\}$ , the corresponding transmit power vectors  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$ , and the corresponding airtime allocation vectors  $\{u_1, u_2, \dots, u_K\}$ .

In the first step of IDGS, the links are sorted according to their traffic demands in an increasing order. In each iteration of the main scheduling while-loop (lines 3 to 11), we determine one feasible matching  $\mathcal{E}_k$ , the corresponding transmission power vector  $\mathbf{p}_k$ , and an integral airtime allocation  $u_k$ . The main idea of IDGS is that we want to eliminate one link from the link set  $\mathcal{L}$  in each iteration so that it does not have to be considered in future iterations. At the same time, we want to allow as many other links to transmit together with the link to be removed. For these other links, we would like to address those links with heavy demands. In IDGS, we choose to eliminate the link with the lightest traffic demand. The purpose of the inner while-loop (lines 5 to 10) is to compose  $\mathcal{E}_k$ . In the  $k$ th iteration, let  $l_{ki}$  and  $f_{ki}$  denote the  $i$ th link in the remaining link set  $\mathcal{L}$  and the remaining traffic demand of link  $l_{ki}$ , respectively. The feasible matching  $\mathcal{E}_k$  is formed in a greedy way. We first put link  $l_{k1}$  that has the minimum traffic demand among the remaining link set  $\mathcal{L}$  into  $\mathcal{E}_k$ . Then we start from the last link which has the largest traffic demand within the remaining link set  $\mathcal{L}$  and add one link to the set  $\mathcal{E}_k$  at a time. The link added to the set  $\mathcal{E}_k$  must satisfy the condition that the link and all the links already in the  $\mathcal{E}_k$  form a feasible matching. This can be done by checking the conditions in (10). If adding a link will cause  $\mathcal{E}_k$  to be infeasible, we move on to the next link without adding it.

After the feasible matching  $\mathcal{E}_k$  is formed, we allocate  $u_k = f_{k1}$  time slots to  $\mathcal{E}_k$ . The transmission power vector of the links in set  $\mathcal{E}_k$  is computed by  $(\mathbf{I} - \mathbf{D}(\boldsymbol{\gamma}_{\mathcal{E}_k}) \mathbf{B}_{\mathcal{E}_k})^{-1} \mathbf{v}_{\mathcal{E}_k}$ . If link  $l_{ki}$  is selected in the feasible matching  $\mathcal{E}_k$ , the remaining traffic demand of link  $l_{ki}$  is updated by subtracting  $u_k$  from it. And if all the traffic demand of link  $l_{ki}$  has been satisfied, link  $l_{ki}$  can be removed from the link set  $\mathcal{L}$ .

The IDGS algorithm continues until all the links are removed from the link list. The maximum number of iterations required is no larger than the number of links in the network and is typically much smaller.

## VII. SIMULATION RESULTS

We carry out extensive simulations to evaluate the performances of the proposed column generation method and the branch-and-price method. We conduct our simulations on a computer

with a 1.86 GHz CPU and 1 GB of RAM.

In our simulations, random network topologies are generated. The locations of the transmitters are uniformly distributed in a square area of  $1000m \times 1000m$ . The length of each link ranges from 100 to 200 meters. More specifically, the location of each receiver is randomly chosen within a radius of 200 meters and outside a radius of 100 meters from the corresponding transmitter. A random network with 32 links is shown in Fig. 4. The red nodes represent the transmitters and the blue nodes represent the receivers. The large-scale path loss model with a typical path loss exponent of 4 is assumed. The maximum transmit power is  $100mW$ . The required SINR threshold is uniformly distributed over  $[10dB, 20dB]$ . The traffic demand of each link is a discrete random variable with 10 equally likely values chosen from the set of integers  $\{1, 3, 5, \dots, 19\}$ . The expected traffic demand is 10 time slots. For each given number of links, we investigate 100 random networks and present the averaged results.

#### A. Finding Optimal Solutions

Our first investigation focuses on finding optimal solutions to problems (P1) and  $(P1)_{INT}$ . We compare the performances of the following four algorithms:

- 1) CG-SE (our algorithm): the column generation method in which the pricing problem is simplified as (SPP). The (SPP) is solved with CSCR heuristic first. If the CSCR fails then we turn to the Smart Enumerating (SE) algorithm to solve (SPP) to optimality. The initial feasible matchings are found by the IDGS algorithm.
- 2) B&P-SE (our algorithm): the branch-and-price method in which the enhanced branching rule (i.e., (15) and (16)) is used and the linear relaxation at each node is solved with CG-SE.
- 3) CG-traditional: the column generation method in which the pricing problem (PP) is solved directly with optimization solver YALMIP [26]. The incidence matrix of the initial feasible matchings is the identity matrix.
- 4) B&P-traditional: the branch-and-price method in which the conventional branching rule (i.e., (12) and (13)) is used and the linear relaxation at each node is solved with CG-traditional.

The simulation results of random networks with 18 links are shown in Table II. The CG-traditional and CG-SE algorithms find optimal solutions to (P1); the B&P-traditional and B&P-SE algorithms find optimal solutions to  $(P1)_{INT}$ . Fig. 5 shows the average runtimes of these four

algorithms for networks of different sizes. It is clear that CG-SE outperforms CG-traditional, and B&P-SE outperforms B&P-traditional in terms of the runtime performance. The improvement becomes more significant as the number of links increases. When the network has 18 links, the average runtimes of CG-traditional and B&P-traditional are 1461.3 seconds and 1576.7 seconds, respectively. However, the average runtimes of CG-SE and B&P-SE are only 2.031 seconds and 4.336 seconds, respectively. CG-SE and B&P-SE reduce the average runtimes by 99.86% and 99.72%, respectively. In column generation and branch-and-price, the computational effort is mainly spent in solving the pricing problem to optimality. In CG-traditional and B&P-traditional, the pricing problem (PP) is solved directly with the general optimization solver. However, in CG-SE and B&P-SE, we solve the simplified pricing problem (SPP) with the Smart Enumerating (SE) algorithm instead. Thanks to the Perron-Frobenius eigenvalue condition, the complexity in the pricing problem can be reduced significantly. We can conclude that if we want to guarantee optimality, CG-SE and B&P-SE are much more efficient than CG-traditional and B&P-traditional.

### B. Performance of Column Generation and Branch-and-Price Based Heuristics

Next, we investigate the performances of the column generation and the branch-and-price based heuristics. The column generation and the branch-and-price methods become heuristic algorithms for (P1) and (P1)<sub>INT</sub> when the pricing problem is solved sub-optimally with the CSCR heuristic. We study the performances of the following three heuristic algorithms:

- 1) IDGS: Increasing Demand Greedy Scheduling algorithm, a heuristic for (P1)<sub>INT</sub>.
- 2) CG-Heu: the column generation based heuristic for (P1) in which the pricing problem is solved with CSCR only. The initial feasible matchings are found by the IDGS algorithm.
- 3) B&P-Heu: the branch-and-price based heuristic for (P1)<sub>INT</sub> in which the enhanced branching rule (i.e.,(15) and (16)) is used and the linear relaxation is solved with CG-Heu.

For comparison purposes, we also obtain optimal solutions to (P1) and (P1)<sub>INT</sub> with CG-SE and B&P-SE, respectively. We introduce the percentage cost penalty to describe the penalty of the above three heuristic algorithms compared to the optimal solution. Let  $W_{algorithm}$  and  $W_{opt}$  denote the number of time slots the algorithm needs and the optimal value, respectively. The percentage cost penalty of each algorithm is defined by

$$P_{algorithm} = \frac{W_{algorithm} - W_{opt}}{W_{opt}} \times 100\%. \quad (18)$$

The results of random networks containing 32 links are shown in Table III. Fig. 6 shows the performance of the average percentage cost penalty of the three heuristic algorithms as a function of the number of links. We see that the performance of IDGS is the worst among the three algorithms. Compared with  $P_{IDGS}$ ,  $P_{CG-Heu}$  and  $P_{B\&P-Heu}$  increase much more slowly with number of links (i.e., network density). For 32-link networks, The averaged value of  $P_{IDGS}$  is 25.15%. However, the averaged values of  $P_{CG-Heu}$  and  $P_{B\&P-Heu}$  are reduced to 15.76% and 13.24%, respectively. In both Fig. 6 and Table III, it is surprising to find that B&P-Heu has better performance than CG-Heu in terms of the average percentage cost penalty. This can be explained as follows. The solution of CG-Heu is the same as the solution of the linear relaxation at the root node in B&P-Heu. In the branch-and-bound tree of B&P-Heu, each node is solved sub-optimally. Therefore, it may happen that in addition to the feasible matchings found at the root node, some more desirable feasible matchings can be found when using CG-Heu to solve the linear relaxation at the child nodes. So it happens that the integer solutions in B&P-Heu can be better than the non-integer solutions in CG-Heu.

Fig. 7 plots the average runtime versus the number of links. We can see that the runtimes of all the three heuristic algorithms are very small compared with the runtimes of CG-SE and B&P-SE. The simplest algorithm, IDGS, consumes the smallest runtime. CG-Heu and B&P-Heu consume a little more time than IDGS. Adding a heuristic column generation or branch-and-price component only adds a small amount of runtime. The average runtimes of these three heuristic algorithms are all less than 10 seconds. Furthermore, we find that CG-SE and B&P-SE work efficiently for networks less than 30 links. The average runtimes of CG-SE and B&P-SE are 31.5 seconds and 57.5 seconds for 29-link networks. However, when the number of links further increases, the runtimes of CG-SE and B&P-SE increase dramatically. The average runtimes of CG-SE and B&P-SE increase to 172.90 seconds and 269.97 seconds for 32-link networks. The reason is that although we are able to reduce the complexity of the pricing problem by incorporating the Perron-Frobenius eigenvalue condition, the pricing problem still has an exponential complexity by nature. Therefore, CG-SE and B&P-SE also have an exponential complexity in the number of links. We can conclude that for modest-size network (i.e., less than 30 links), CG-SE and B&P-SE are computationally efficient optimal algorithms. For large-size network, finding optimal solutions is really difficult. In this situation, CG-Heu and B&P-Heu serve as fast heuristic algorithms that find close to optimal solutions with short runtimes.

## VIII. CONCLUSION

We have considered the minimum-length scheduling problem in STDMA wireless networks with power control, subject to traffic demands and SINR constraints. When power control is considered, the feasibility of a set of links under the SINR constraints can be checked by the Perron-Frobenius eigenvalue condition. This turns out to be a rather useful condition for expediting the optimization. We propose the column generation method that finds optimal time schedule and power solutions. The way to solve the pricing problem is the key to the efficiency of the column generation method. We integrate the Perron-Frobenius eigenvalue condition into both the formulation of the pricing problem and the Smart Enumerating (SE) algorithm. Such integration improves the efficiency of the column generation method. We show that our new formulation, together with the SE algorithm, reduces the average runtime of column generation method by 99.86% for wireless networks with 18 links compared with the traditional column generation method. We further propose the branch-and-price method that combines the column generation with the branch-and-bound to provide optimal integer time schedule solutions. We develop a new branching rule in the branch-and-price method that maintains the size of the pricing problem after each branching, and thus improves the overall efficiency of the branch-and-price method. For example, our branch-and-price method reduces the average runtime by 99.72% for wireless networks with 18 links compared with the traditional branch-and-price method.

Both the column generation and the branch-and-price methods can be used as heuristic algorithms if we solve the pricing problem sub-optimally. We propose a simple heuristic algorithm, Combined Sum Criterion Removal (CSCR), for the pricing problem. Simulation results show that the column generation and the branch-and-price based heuristics can obtain near-optimal solutions with very short runtimes. In particular, the average cost penalties are below 16%, with the average runtimes both less than 10 seconds for 32-link networks.

## REFERENCES

- [1] P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks," *IEEE Trans. Information Theory*, vol.46, no. 2, pp. 388-404, Mar. 2000.
- [2] B. Hajek and G. Sasaki, "Link scheduling in polynomial time", *IEEE Trans. Information Theory*, vol. 34, no. 5, pp. 910-917, Sep. 1988.
- [3] G. Sharma, R.R. Mazumdar, and N.B. Shroff, "On the Complexity of Scheduling in Wireless Networks", *In Proceedings of ACM Mobicom*, pp. 227-238, 2006.
- [4] O. Goussevskaia, Y.A. Oswald, and R. Wattenhofer, "Complexity in Geometric SINR", *Proc. ACM MobiHoc*, 2007.

- 1  
2  
3  
4 [5] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *In Proceedings of ACM Mobicom*, pp. 66-80, 2003.
- 5  
6 [6] W. Wang, Y. Wang, X. Y. Li, W. Z. Song and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks", *In Proceedings of ACM Mobicom*, pp. 262-273, 2006.
- 7  
8 [7] G. Brar, D. M. Blough and P. Santi, "Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks", *Proc. ACM Mobicom*, pp. 2-13, 2006.
- 9  
10 [8] A. Behzad, I. Rubin and P. Chakravarty, "Optimum integrated link scheduling and power control for ad hoc wireless networks", *Proc. IEEE WiMob*, pp. 275-283, Aug. 2005.
- 11  
12 [9] U. C. Kozat, I. Koutsopoulos and L. Tassiulas, "Cross-layer design for power efficiency and QoS provisioning in multi-hop wireless networks", *IEEE Trans. on Wireless Communications*, vol.5, no. 11, pp. 3306-3315, Nov. 2006.
- 13  
14 [10] T. Elbatt and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks", *IEEE Trans. on Wireless Communications*, vol.3, pp. 74-85, Jan. 2004.
- 15  
16 [11] J. Tang, G. L. Xue, C. Chandler and W. Zhang, "Link scheduling with power control for throughput enhancement in multihop wireless networks", *IEEE Trans. Veh. Technol.*, vol. 55, no. 3, pp. 733-742, May 2006.
- 17  
18 [12] A. K. Das, R. J. Marks, P. Arabshahi and A. Gray, "Power controlled minimum frame length scheduling in TDMA wireless networks with sectored antennas", *Proc. IEEE Infocom*, pp. 1782-1793, 2005.
- 19  
20 [13] S. A. Borbash and A. Ephremides, "Wireless link scheduling with power control and SINR constraints", *IEEE Trans. Information Theory*, vol. 52, no. 11, pp. 5106-5111, Nov. 2006.
- 21  
22 [14] R. L. Cruz and A. V. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks", *Proc. IEEE Infocom*, pp. 702-711, 2003.
- 23  
24 [15] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation", *IEEE Trans. on Wireless Communications*, vol.5, no.2, pp. 435-445, Feb. 2006.
- 25  
26 [16] A. Capone and G. Carello, "Scheduling optimization in wireless mesh networks with power control and rate adaptation", *Proc. IEEE SECON*, 2006.
- 27  
28 [17] S. Kompella, J.E. Wieselthier and A. Ephremides, "Revisiting the optimal scheduling Problem", *in Proceedings of Conference on Information Sciences and Systems (CISS)*, March 2008.
- 29  
30 [18] J. Zander, "Performance of optimum transmitter power control in cellular radio systems", *IEEE Trans. Veh. Technol.*, vol. 41, no. 1, pp. 57-62, Nov. 1992.
- 31  
32 [19] S. A. Grandhi, R. Vijayan, D. J. Goodman and J. Zander, "Centralized power control in cellular radio systems", *IEEE Trans. Veh. Technol.*, vol. 42, no. 6, pp. 466-468, Nov. 1993.
- 33  
34 [20] D. Mitra, "An asynchronous distributed algorithm for power control in cellular radio systems", *Proc. 4th WINLAB Workshop*, Rutgers University, New Brunswick, NJ, 1993.
- 35  
36 [21] N. Bambos, C. Chen, and G. Pottie, "Channel access algorithms with active link protection for wireless communication networks with power control", *IEEE/ACM Transactions on Networking*, vol. 8, no. 5, pp. 583-597, Oct. 2000.
- 37  
38 [22] D. Bertsimas and J.N. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1997.
- 39  
40 [23] M. E. Lubbecke and J. Desrosiers, "Selected topics in column generation", *Operations Research*, vol. 53, no. 6, Nov.-Dec. 2005.
- 41  
42 [24] R. A. Horn and C. R. Johnson, *Matrix Analysis*, New York: Cambridge Univ. Press, 1991.
- 43  
44 [25] S. A. Borbash and A. Ephremides, "The feasibility of matchings in a wireless network", *IEEE Trans. Information Theory*, vol. 52, no. 6, pp. 2749-2755, Nov. 2006.
- 45  
46 [26] J. Löfberg, "YALMIP : a toolbox for modeling and optimization in MATLAB", *Proceedings of the CACSD Conference*, 2004, <http://control.ee.ethz.ch/~joloef/yalmip.php>.
- 47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

TABLE I  
KEY NOTATIONS

Notation	Physical Meaning
$\mathcal{L}$	the set of all links
$\mathcal{M}$	matching (Definition 2)
$\mathcal{T}$	the set of transmitters
$\mathcal{R}$	the set of receivers
$\mathcal{E}$	the set of all the feasible matchings (Definition 3)
$G(T_i, R_j)$	the channel gain between nodes $T_i$ and $R_j$
$p_i$ (in vector $\mathbf{p}$ )	transmit power of link $l_i$
$\gamma_i$ (in vector $\boldsymbol{\gamma}$ )	SINR requirement of link $l_i$
$\eta_i$	the average noise power of link $l_i$
$v_i = \gamma_i \eta_i / G(T_i, R_i)$ (in vector $\mathbf{v}$ )	the normalized noise power of link $l_i$
$\mathbf{B}$	relative channel gain matrix
$\rho(\mathbf{B})$	the Perron-Frobenius eigenvalue of matrix $\mathbf{B}$
$\mathbf{D}(\cdot)$	diagonal matrix operator
$\mathbf{Q}$	incidence matrix denoting all the feasible matchings
$U$	fixed frame length
$W$	the airtime length to satisfy the on-demand traffic
$f_i$ (in vector $\mathbf{f}$ )	the traffic demand of link $l_i$
$u_k$ (in vector $\mathbf{u}$ )	the airtime allocated to feasible matching $\mathcal{E}_k$

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

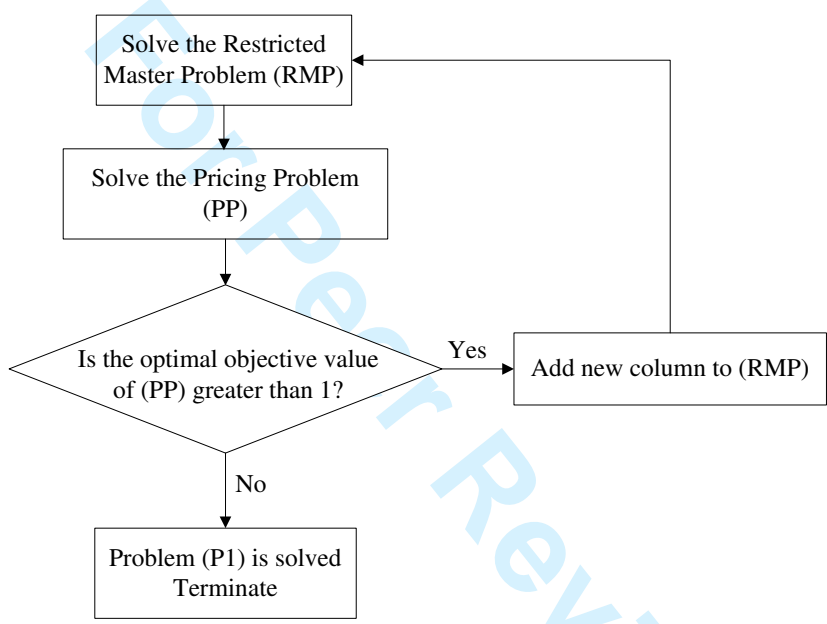


Fig. 1. Flowchart of the column generation method

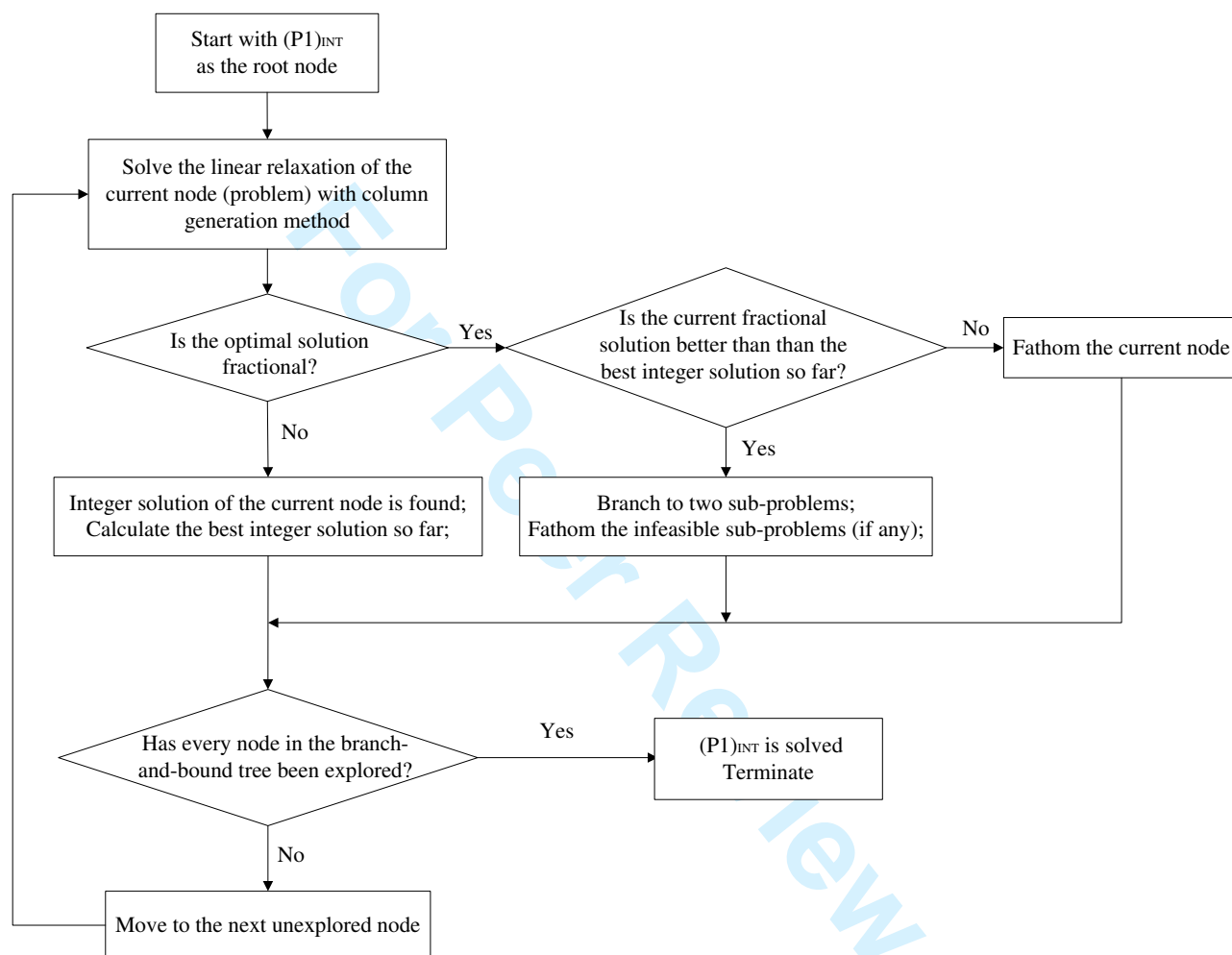


Fig. 2. Flowchart of the branch-and-price method

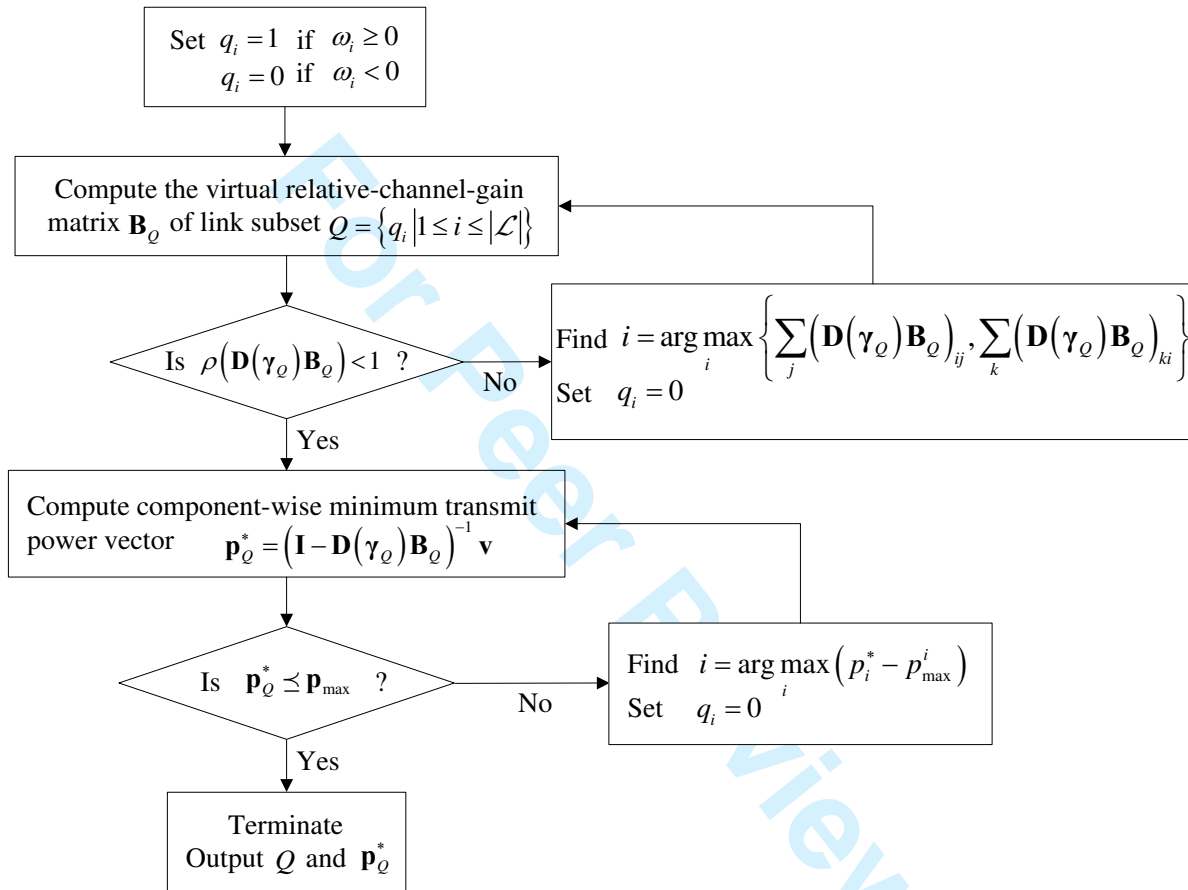


Fig. 3. Flowchart of the CSCR algorithm

**Algorithm 1:** Increasing Demand Greedy Scheduling (IDGS)

**Input:** the virtual relative-channel-gain matrix  $\mathbf{B}_{\mathcal{L}}$ , the traffic demand vector  $\mathbf{f}$ , the SINR requirement vector  $\boldsymbol{\gamma}$ , the normalized noise vector  $\mathbf{v}$

**Output:** a subset of maximal feasible matchings  $\{\mathcal{E}_1, \mathcal{E}_2, \dots, \mathcal{E}_K\}$ , the transmit power vectors  $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$ , the set of integers  $\{u_1, u_2, \dots, u_K\}$

1 Sort the links in increasing order of their traffic demands  $f_1 \leq f_2 \leq \dots \leq f_{|\mathcal{L}|}$ ;

2  $k \leftarrow 1$ ;

3 **while**  $\mathcal{L} \neq \emptyset$  **do**

4      $\mathcal{E}_k \leftarrow \{l_{k1}\}$ ;  $u_k \leftarrow f_{k1}$ ;  $i \leftarrow |\mathcal{L}|$ ;

5     **while**  $i \geq 1$  **do**

6         **if**  $\mathcal{E}_k \cup \{l_{ki}\}$  forms a feasible matching **then**

7              $\mathcal{E}_k \leftarrow \mathcal{E}_k \cup \{l_{ki}\}$ ;  $f_{ki} := f_{ki} - u_k$ ;

8             **if**  $f_{k,i} \leq 0$  **then**

9                  $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l_{ki}\}$ ;

10          $i \leftarrow i - 1$ ;

11      $\mathbf{p}_k \leftarrow (\mathbf{I} - \mathbf{D}(\boldsymbol{\gamma}_{\mathcal{E}_k}) \mathbf{B}_{\mathcal{E}_k})^{-1} \mathbf{v}_{\mathcal{E}_k}$ ;

TABLE II

SIMULATION RESULTS (THE NETWORK CONTAINS 18 LINKS)

Algorithms	Average number of time slots	Average runtime (sec)	Minimum runtime (sec)	Maximum runtime (sec)
CG-traditional	70.7	1461.3	601.52	2170.3
B&P-traditional	70.9	1576.7	630.33	2486.8
CG-SE	70.7	2.031	0.33	6.129
B&P-SE	70.9	4.336	0.33	30.99

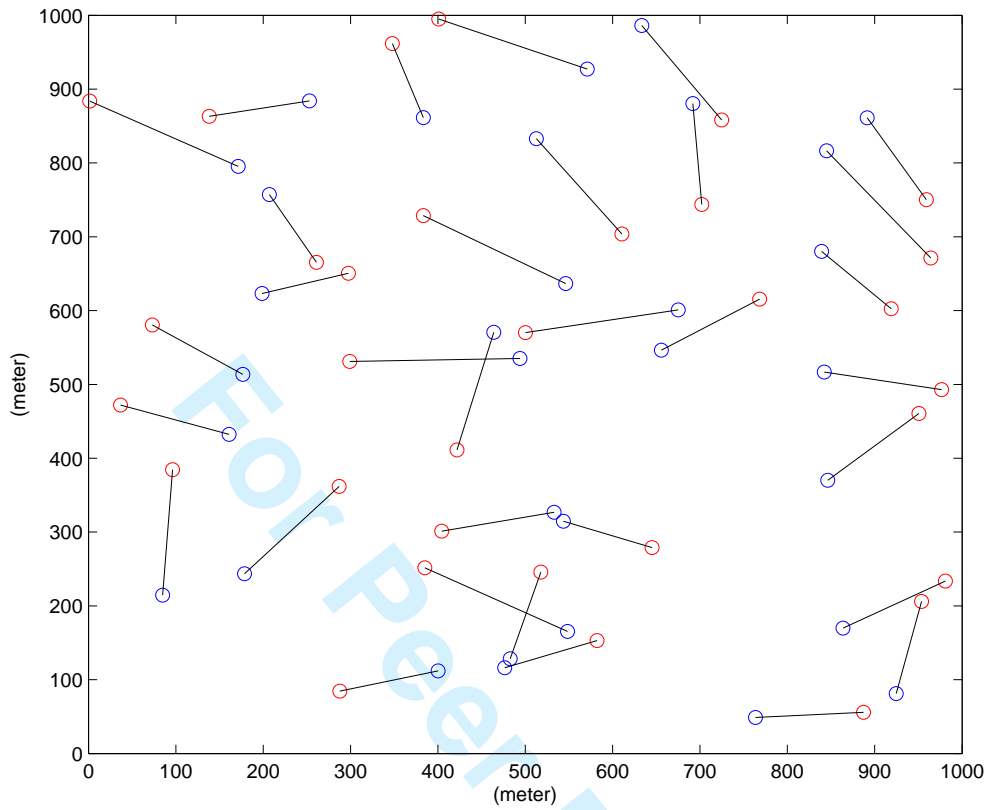


Fig. 4. Random Network Topology with 32 Links

TABLE III  
SIMULATION RESULTS (THE NETWORK CONTAINS 32 LINKS)

Algorithms	Average number of time slots	Average percentage cost penalty	Average runtime (sec)
CG-SE	98.05	0%	172.90
B&P-SE	98.20	0%	269.97
IDGS	122.9	25.15%	0.065
CG-Heu	113.5	15.76%	1.756
B&P-Heu	111.2	13.24%	5.620

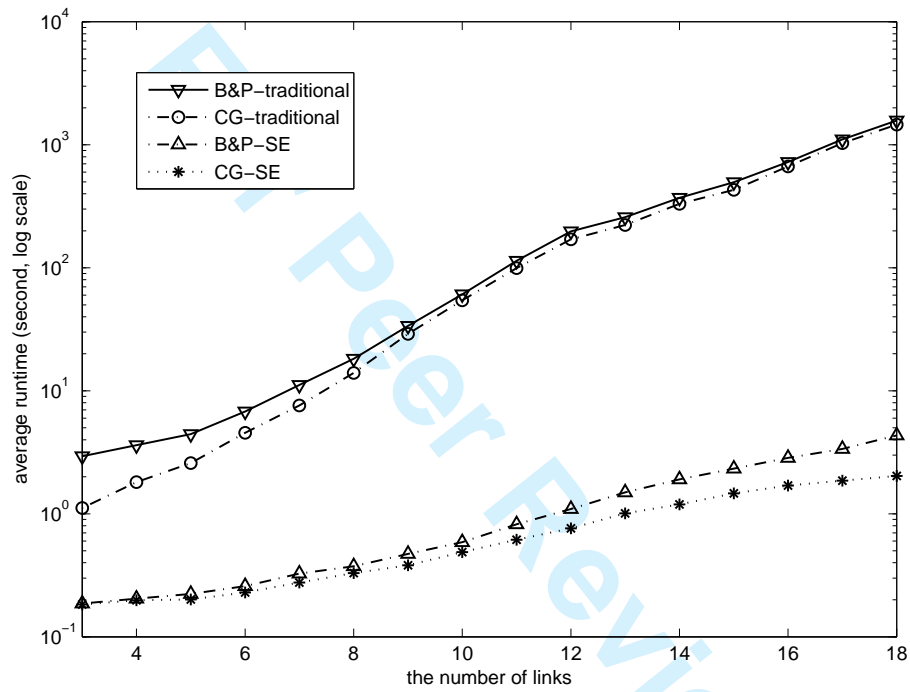


Fig. 5. Average runtime performance of our column generation method (CG-SE) and branch-and-price method (B&P-SE), compared with traditional column generation method (CG-traditional) and branch-and-price method (B&P-traditional)

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

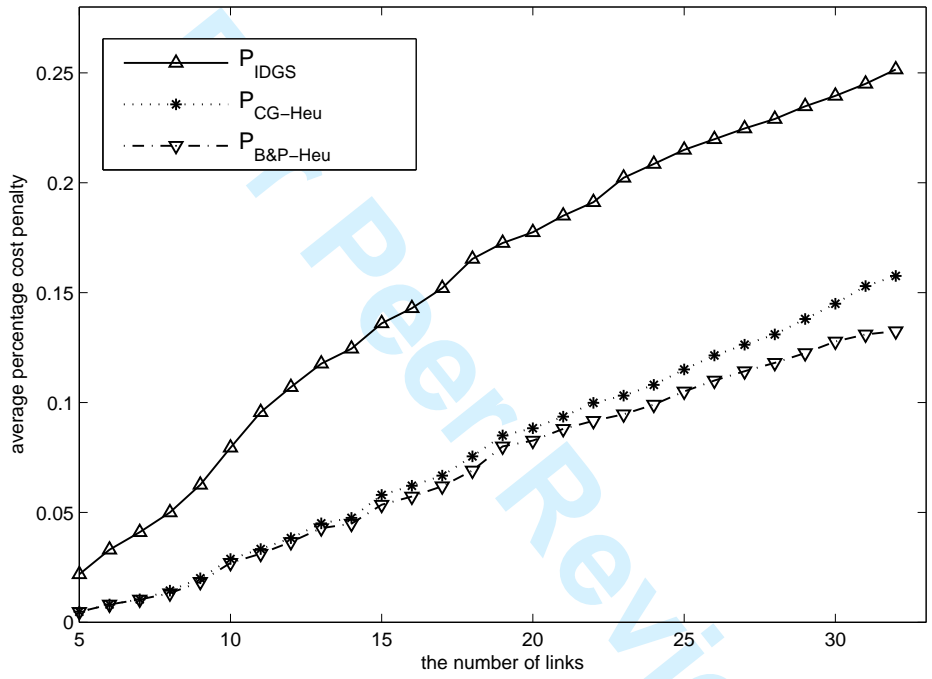


Fig. 6. Average percentage cost penalty of the column generation and branch-and-price based heuristic algorithms (CG-Heu and B&P-Heu), compared with the simple heuristic algorithm IDGS

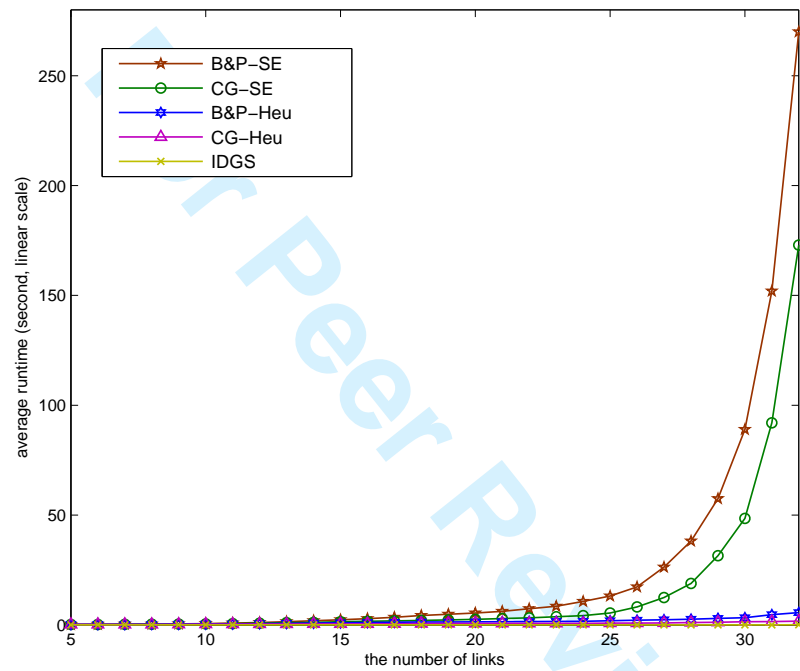


Fig. 7. Average Runtime Performance of the column generation and branch-and-price based heuristic algorithms (CG-Heu and B&P-Heu), compared with the simple heuristic algorithm IDGS, the column generation and branch-and-price methods that find optimal solutions (CG-SE and B&P-SE)