# Solving a Resource Allocation Problem in Wireless Mesh Networks: a Comparison Between a CP-based and a Classical Column Generation

Antonio Capone, Giuliana Carello*, Ilario Filippini,
Stefano Gualandi, Federico Malucelli
Dipartimento di Elettronica e Informazione,
Politecnico di Milano, Italy
{capone, carello, filippini, gualandi, malucell}@elet.polimi.it

**Abstract**

This paper presents a column generation approach to a resource allocation problem arising in managing Wireless Mesh Networks. The problem consists in routing the given demands over the network and to allocate time resource to pairs of nodes. Half-duplex constraints are taken into account together with the aggregate interference due to simultaneous transmissions, which affects the signal quality. Different problems are considered, according to the assumptions on the transmission power and rate. The resource allocation problem can be formulated as a Mixed Integer Linear Programming problem and dealt with a column generation based approach. The pricing problem, due to signal quality constraints, turns out to be computationally demanding. In order to tackle these difficulties, besides a classical mathematical programming approach, we have applied a hybrid column generation approach where the pricing sub-problem is solved using Constraint Programming. Numerical results show that the two methods are comparable. The results of the column generation are then used to solve heuristically the problem. The obtained results provide very small gaps (between lower bounds and heuristic solutions) for two of the three considered problems, and reasonable gaps for the third problem.

## 1 Introduction

Wireless Mesh Networks (WMNs) are two-tier wireless ad-hoc networks whose lower layer is composed by end-user terminals (PDAs, smartphones, notebooks, etc.) that can connect to higher layer access point nodes via radio transmission.

*Corresponding author.

The set of access point nodes forms a backbone network made of wireless links, that is used to carry traffic demands among end-users. Demands are routed along multi-hop paths either by using the wireless links only, or by exploiting the connection of some higher level nodes with a wired backbone.

Given such a flexible network architecture, WMNs can potentially provide low-cost connectivity to a large number of user terminals. In addition, the flexibility makes WMNs well suited to support a wide spectrum of applications from security and territory surveillance to bringing communication in devastated areas. WMNs represent a competitive alternative to wired solutions for the provision of cheap broadband to hard-to-reach areas.

The higher level nodes are assumed to have a fixed layout in the territory, while the lower level ones, by definition, are mobile. The wireless backbone network in practice provides an almost static resource assignment to traffic flows. This allows to simplify the radio resource management at the interface between the network and the end-users and to provide quality of service guarantees.

These assumptions give rise to two main classes of optimization problems. On the one hand there are network planning issues: where to place the higher level nodes, which wireless links to set up, and in some cases which higher level nodes to connect to the wired network [1]. On the other hand, once the network is planned, there are the resource allocation, routing and scheduling problems, considering the fact that the network is made of wireless links and the activity of one link may influence that of the others.

In this paper we deal with this second class of problems – resource allocation, routing and scheduling – under different hypothesis on the transmission power and rate. The main considered assumptions are: the type of transmission protocol, the fact that each node of the network either receives or transmits on a single incident wireless link, the signal qualities affected by the overall interference due to simultaneous transmissions. Moreover, the proposed models also give the optimal routing of a given traffic matrix of point-to-point transmissions, which exploits (and actually affects) the resource allocation.

The WMN's wireless backbone – WMN in the remaining – is composed of nodes communicating through omni-directional antennas which share the same radio channel. Each node can establish a communication link at a time and simultaneously active wireless transmissions are mutually dependent, as they may interfere each other causing errors at receivers. A typical scheme used to guarantee a requested bandwidth is the Time Division Multiple Access (TDMA) scheme, this paper focuses on it.

In this protocol, the time horizon is subdivided into frames. Each frame is discretized into slots of fixed duration. In order to exploit efficiently the protocol features, the links should be assigned to the slots of the frame so as to minimize the frame length. The same assignment is then replicated cyclically in the following frames, thus providing a resource allocation among links.

Given a set of demands specified by the origin and destination nodes and the amount of information (*i.e.*, number of packets) to be sent during a frame, the resource allocation problem consists in determining a routing of the demands along the links and simultaneously assigning the links to the frame slots. Us-

ing high level of power makes the communication reliable, but increases both interference and energy consumption. The objective of the optimization is to minimize the number of time slots needed in a frame to route all the demands. We deal with three versions of the problem. In the basic problem each node transmits one packet in each assigned time slot at a fixed power level. In the second version, we allow a node to select the transmission power. Finally, the more involved version considers also variable transmission rates.

The problems are formulated as Mixed Integer Programs where we introduce a variable for each transmission pattern, that is a set of simultaneous transmissions over the links. The exponential number of variables suggests to approach the problems with a column generation based techniques. This problem must take into account the mutual exclusive transmission constraints and the interference constraints which seem to be extremely challenging for MILP solvers. Along with MILP formulations, we present Constraint Programming (CP) formulations and use the CP solver Gecode [12] to solve the problems.

The reminder of the paper is organized as follows: Section 2 introduces the problem and review the state of the art; Section 3 details the mathematical programming models; Section 4 presents the Constraint Programing formulations; Section 5 reports on the computational results of the column generation for solving the continuous relaxation of the problem; Section 6 shows how to use the results of the column generation to obtain an integer upper bound; finally, in Section 7, we conclude.

## 2 Problem description

A WMN topology with $n$ clients is formalized with a directed graph $(\mathcal{N}, \mathcal{L})$. Each node of the WMN is represented by a node of the set $\mathcal{N}$, while the set $\mathcal{L}$ represents the set of links, *i.e.* the pairs of nodes $(i, j)$ that can communicate. In the link $(i, j)$, $i$ is assumed to transmit and $j$ to receive. Each node is equipped with a device transmitting in uni-casting and half-duplex, thus it can be involved in at most one communication at a time, being either *transmitter* or *receiver*. In the TDMA protocol the time horizon is divided into equal length, cyclically repeated frames. Each frame is divided into time slots. To share time resource, communication links are assigned to time slots. In each slot, transmitters can send a certain number of packets, provided that interference requirements at receivers are satisfied. A link can be active (*i.e.* it can send packets) only in those time slots to which is assigned. To guarantee the required transmission quality, a transmission over a link $(i, j)$ can be activated only if the Signal-to-Interference and Noise Ratio (SINR) [14] at the receiver $j$ is sufficiently high to correctly decode the signal, *i.e.* it is above a given threshold. Besides the thermal noise contribution, the SINR takes into account the effect of the cumulative interference generated at the receiver $j$ by all the transmitting nodes different from $i$:

$$SINR_{ij} = \frac{p_i G_{ij}}{\eta + \sum_{l \in \mathcal{N} \setminus \{i,j\}} p_l G_{lj}} \tag{1}$$

where $p_i$ is the transmission power of $i$, and ranges in $[0..P]$, $\eta$ is the thermal noise and the matrix $G_{ij}$ provides the transmission gain between node $i$ and $j$. The transmission gain decreases with the increasing distance. With this interference model, a set of parallel transmissions on a set of links is admissible if the SINR at all receivers is above the given threshold.

Along with the directed network topology a set of traffic demands $\mathcal{D}$ between a subset of pairs of nodes is given, representing the number of packets to be sent. Traffic demands have to be routed over the available links. To send all the packets routed on it, each link must be assigned to a subset of time slots, while providing that uni-casting and half-duplex constraints are satisfied as well as SINR requirements. The problem is to decide the routing of all the traffic demands, together with the assignment of each link to the time slots. The goal is to minimize the number of needed time slots.

According to the assumption on the emitting power and on the transmission rate, three different versions of the problem can be described. In the first one (Fixed Power Resource Allocation problem – FPRA), emitting power $p_i$ is assumed to be equal to the maximum value $P$ for each node $i \in \mathcal{N}$. In each time slot a link sends one packet if it is active.

In the second problem, power can vary and it must be optimized, for each time slot and for each link (Power Control Resource Allocation problem – PCRA). Varying the transmission power allows to assign more links to the same time slot, as the SINR is strictly related to the emitted power of each node, and therefore to better exploit the time resource. A version of PCRA in which the routing is not considered is proved to be NP-hard in [4]. Also in PCRA the transmission rate is fixed and one packet is sent for each time slot.

In the third case (Power and Rate Control Resource Allocation problem – PRCRA), transmission rate can be optimized as well: the number of packets sent in a time slot can increase. Obviously, to guarantee the quality of signals the SINR threshold must grow according to the rate.

# 3 Related work

Many papers have been published on different versions of the joint routing and scheduling – *i.e.* assigning links to time slots – problem. In the following we present a brief survey of those closest to the problems addressed in this paper.

**Fixed Power Resource Allocation problem.** In [6] several Integer Linear Programming (ILP) models for fixed-power scheduling-only optimization are presented, in particular some of these are solved by column generation methods. In the models proposed in [6], it is assigned at least one time slot to each pair of nodes that can transmit, without considering the traffic load. Some other works, even though they do not apply any column generation technique, are based on transmission patterns. In [22] the authors consider a reduced set of all possible fixed-power transmission patterns. This set is populated with a greedy algorithm that maximizes the number of simultaneously active links for

each pattern. Once the set is created, a LP model considers patterns activated for a time interval to be optimized in order to minimize the total activation time. Given the exponential number of possible transmission patterns, this work presents results with small size networks or highly constrained patterns.

**Power Control Resource Allocation problem.** In [10] joint scheduling and power control are considered, with the goal of maximizing network throughput. Routing is not taken into account. The general objective is to maximize network throughput. The solution approach proposed is based on the decomposition of the problem into two sub-problems. The first sub-problem heuristically finds the maximum subset of simultaneous transmissions, while the second one sets power in order to minimize the total transmitted power. The power controlled scheduling algorithm proposed in [3] is also based on a two steps approach, however the set of links obtained in the first phase is usually unfeasible and it is pruned in the second phase through a greedy algorithm that directly considers SINR constraints. The authors in [4] propose a MILP model with quadratic constraints, which does not consider the routing. The optimal solution is only used to benchmark the quality of their heuristic algorithm for small instances. In [21] a throughput maximization problem is presented within a scheduling and power control framework. The solution approach consists in solving iteratively the relaxed LP formulation of the MILP problem. At each iteration some links are fixed and the next iteration tries to add other feasible simultaneous transmissions. In [17] end-to-end routing is included in the optimization problem. The routing problem is separately solved through a shortest path approach defining a link metric based on the number of interference conflicts caused by each link. The scheduling and power control problems are then solved with a greedy approach. In general, previous works reported in this paragraph are characterized by a significant optimality gap or by a solution quality comparison with respect to other heuristic approaches.

**Power and Rate Control Resource Allocation problem.** In [9] the joint routing, scheduling, and power control problem is considered taking into account long term minimum rate requirements of traffic flows. A two steps approach is considered that optimizes separately scheduling with power control and routing with rate control. The optimization objective is the total power minimization. In [5] this problem version is studied considering a non-linear optimization problem and the solution is found applying an approximation multi-step algorithm. In [8] power minimization routing and scheduling are decomposed in a two-phase problem. The routing is determined with an energy-aware routing protocol that discovers shortest paths with an energy metric. Once the routing is fixed, a power control link scheduling is adopted to minimizes the total power, given a minimum guaranteed rate. In spite of heuristic solution approaches, these works consider small size networks with 5-6 nodes. Other works show results on bigger instances, however none of them careful assesses the quality of obtained solutions in terms of optimality. A MILP formulation for the joint scheduling,

power control, and rate control problem is proposed in [16] but the solution approach is based on a greedy algorithm. In [15] generic non linear formulations are proposed for routing and transmission scheduling problems. The aims is to evaluate the performance of a network in terms of throughput, rate, or fairness, rather than providing a tool to solve real life size resource allocation problems. Finally, in [7] all the three versions of the problem, considering only the scheduling, are introduced. Extensive computational results obtained via a classical column generation approach are reported along with their quality.

To the best of our knowledge, none of the previous papers present an approach providing both lower and upper bounds for all the different versions of the problems and tackling real life size instances.

# 4 Models

In this section the mathematical programming models of the different versions of the problem are presented: in the first one the power is fixed for each link (FPRA), in the second one emitting power is to be optimized (PCRA), while in the third one both power and transmission rate are variables (PRCRA).

We formalize WMN topology with the above described graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$. The traffic demands are represented by a set of triplets $\mathcal{D} = \{\langle o, t, d\rangle \mid o, t \in \mathcal{N}, d \in \mathbb{Z}^+\}$, where each triplet $\langle o, t, d\rangle$ indicates the number of packets $d$ to be routed from node $o$ to node $t$ in a frame. The transmission gain between two devices $i$ and $j$ is denoted by $G_{ij}$, and the SINR threshold by $\gamma$. The problem can be formulated using variables associated to *configurations*, *i.e.*, sets of links that can be active simultaneously without violating the SINR constraint (1) and the half-duplex matching constraint. The set of all possible configurations (columns) is denoted by $\mathcal{S}$, while the set of configurations with link $(i, j)$ active by $\mathcal{S}_{ij}$.

## 4.1 Basic version: fixed transmission power, and no rate adaptation

We introduce an integer variable $\lambda_s$ for each configuration in $\mathcal{S}$; variable $\lambda_s$ represents the number of slots to which configuration $s$ is assigned. The total number of time slots is equal to the sum of variables $\lambda_s$, since only one configuration is active in a time slot. Besides $\lambda_s$ variables, a set of flow variables $f_{ij}^k$ is introduced, indicating the number of packets of demand $\langle o^k, t^k, d^k\rangle$ routed over link $(i, j)$. The problem of routing and scheduling the traffic demand set $\mathcal{D}$ is

formalized as follow in case of fixed power and rate (Master Problem MP):

$$\min \sum_{s \in \mathcal{S}} \lambda_s \tag{2}$$

$$\text{s.t.} \sum_{(i,j) \in \mathcal{L}} f_{ij}^k - \sum_{(j,i) \in \mathcal{L}} f_{ji}^k = \begin{cases} d^k & i = o^k \\ -d^k & j = t^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in \mathcal{N}, \forall k \in \mathcal{D} \tag{3}$$

$$\sum_{s \in \mathcal{S}_{ij}} \lambda_s \geq \sum_{k \in \mathcal{D}} f_{ij}^k \qquad \forall (i,j) \in \mathcal{L} \tag{4}$$

$$f_{ij}^k \in \mathbb{Z}^+ \qquad \forall (i,j) \in \mathcal{L}, \forall k \in \mathcal{D} \tag{5}$$

$$\lambda_s \in \mathbb{Z}^+ \qquad \forall s \in \mathcal{S} \tag{6}$$

The objective function (2) minimizes the number of time slots. Constraints (3) are the routing constraints imposing flow balance at every node $i$, for each demand $k$. Constraints (4) impose that every link $(i, j)$ is assigned to a number of time slots at least equal to the number of times the link is used for routing any demand, while (5) and (6) are integrality constraints.

Let us remark that, differently from [6], routing constraints (3) are considered, and in constraints (4) traffic load is taken into account. As traffic load is to be satisfied, one configuration can be used more than once, thus integer variables are used instead of binary ones.

As the number of configurations is huge, the continuous relaxation of MP (2)–(6) is to be solved applying column generation. We start solving the continuous relaxation of the master problem over a subset of configurations $\mathcal{S}_0 \subseteq \mathcal{S}$ (Restricted Master Problem), and then, using the dual variables $\overline{\sigma}_{ij}$ of constraints (4), we iteratively generate new configurations to be added to $\mathcal{S}_0$ through a pricing procedure. Since the routing is considered in the master problem, and the bottleneck of the approach is the pricing solution, these constraints do not increase the difficulty of the problem.

To generate a feasible improving configuration, we consider that all nodes transmit in uni-casting and are half-duplex, and that the active links must satisfy the SINR constraint. We introduce a binary variable $z_{ij}$ for each arc $(i, j)$ in $\mathcal{L}$; variable $z_{ij} = 1$ if and only if the corresponding link is active in the configuration to be built. In case of fixed power, the emitting power of each node is equal to the maximum value, denoted with $P$. The pricing problem solved to verify whether the solution of the Restricted Master Problem is optimal for the

MP or to generate an improving configuration is the following:

$$\max \quad \sum_{(i,j)\in\mathcal{L}} \overline{\sigma}_{ij} z_{ij} \tag{7}$$

$$\text{s.t.} \quad \sum_{(i,j)\in\mathcal{L}} z_{ij} + \sum_{(j,i)\in\mathcal{L}} z_{ji} \leq 1 \qquad \forall i \in \mathcal{N} \tag{8}$$

$$PG_{ij} \geq \gamma \left( \eta + \sum_{\substack{h\in\mathcal{N}\setminus\{i,j\} \\ l\in\mathcal{N}\setminus\{i,j\}}} PG_{hj} z_{hl} \right) z_{ij} \qquad \forall (i,j) \in \mathcal{L} \tag{9}$$

$$z_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{L} \tag{10}$$

Objective function (7) maximizes over the optimal dual variables $\overline{\sigma}_{ij}$ of constraint (4). By Duality Theory, an improving column exists if and only if the optimal value is greater than 1, *i.e.*, the reduced cost of the corresponding configuration is negative. Constraints (8) assign each device $i \in \mathcal{N}$ to at most one of its incident links; this realizes the uni-casting and half-duplex constraints. Constraints (9) are the SINR non-linear constraints: if link $(i,j)$ is active, then the ratio between the received signal and the sum of the thermal noise and the amount of signal of other active nodes received by $j$ must be above the given threshold $\gamma$.

## 4.2 Extensions to power control and rate adaptation

**Extension to power control.** Let now extend the models to the power control case (PCRA). The master problem (2)–(6) does not involve power control, so it does not change, while we introduce a continuous non-negative variable $p_i$ for each transmitting node in the pricing problem. The objective function (7) and constraints (8) are kept, while constraint (9) is to be replaced by

$$p_i G_{ij} \geq \gamma \left( \eta + \sum_{h\in\mathcal{N}\setminus\{i,j\}} p_h G_{hj} \right) z_{ij} \qquad \forall (i,j) \in \mathcal{L}. \tag{11}$$

Further, a new constraint is added connecting $z_{ij}$ and $p_i$ variables:

$$p_i \leq P \sum_{(i,j)\in\mathcal{L}} z_{ij} \qquad \forall i \in \mathcal{N} \tag{12}$$

**Extension to rate adaptation.** Finally, we extend the formulations to PRCRA. By changing the transmission rate it is possible to increase the number of packets sent by a link in a time slot. Each configuration is now described by a set of links and the rate – and therefore the number of sent packets – for each link. Let constant $v_{ijs}$ represent the number of packets sent over link $(i,j)$ in configuration $s$, which depends on the chosen rate. Constraints (4) of the master

problem becomes:

$$\sum_{s \in \mathcal{S}_{ij}} v_{ijs} \lambda_s \geq \sum_{k \in \mathcal{D}} f_{ij}^k \qquad \forall (i,j) \in \mathcal{L}. \tag{13}$$

To keep the transmission quality good, SINR threshold is to be increased with the increasing of the rate. Therefore the threshold $\gamma$ depends on the chosen rate and it is replaced by $\gamma_w$, $w$ denoting the different admissible rate (set $\mathcal{W}$). Since higher transmission rates require higher SINR thresholds, we introduce a binary variable $z_{ijw}$ for each link $(i,j)$ and for each rate $w$. By replacing in problem (7)–(10) variables $z_{ij}$ with $z_{ijw}$, threshold $\gamma$ with $\gamma_w$, and representing with $T_w$ the number of sent packets associated to rate $w$ we get the pricing problem for PRCRA:

$$\max \quad \sum_{w \in \mathcal{W}} \sum_{(i,j) \in \mathcal{L}} T_w \bar{\sigma}_{ij} z_{ijw} \tag{14}$$

$$\text{s.t.} \quad \sum_{w \in \mathcal{W}} \left( \sum_{(i,j) \in \mathcal{L}} z_{ijw} + \sum_{(j,i) \in \mathcal{L}} z_{jiw} \right) \leq 1 \qquad \forall i \in \mathcal{N} \tag{15}$$

$$p_i G_{ij} \geq \gamma_w \left( \eta + \sum_{h \in \mathcal{N} \setminus \{i,j\}} p_h G_{hj} \right) z_{ijw} \qquad \forall (i,j) \in \mathcal{L}, \forall w \in \mathcal{W} \tag{16}$$

$$p_i \leq P \sum_{\substack{(i,j) \in \mathcal{L} \\ w \in \mathcal{W}}} z_{ijw} \qquad \forall i \in \mathcal{N} \tag{17}$$

$$z_{ijw} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{L}, \forall w \in \mathcal{W} \tag{18}$$

$$p_i \geq 0 \qquad \forall i \in \mathcal{N} \tag{19}$$

The number of packets $v_{ijs}$ sent from $i$ to $j$ in the built configuration $s$ depends on the rate chosen for $(i,j)$: $v_{ijs} = \sum_{w \in \mathcal{W}} T_w \bar{z}_{ijw}$.

Constraints (9), (11), and (16) are non linear, but they can be linearized using big-M constraints.

### 4.3 Strategies for generating new columns

The first solution approach we propose is the classical MILP-based column generation. Unfortunately, the big differences of values in the propagation gain $G_{ij}$ force the use of high value of big-M, resulting in loose constraints and instability issues. This makes the pricing problem hard to solve with an integer linear solver, and therefore we have experimented a few strategies to boost the computation time.

As initial set of columns we either use a *Dummy* set consisting in a configuration $s$ for each link $(i,j)$, where only the link $(i,j)$ is active, or a *Heuristic* one, heuristically generated. Moreover, we use two strategies for adding new columns: (i) solving the pricing to optimality (*OPT*), *i.e.*, looking for the most

negative reduced cost column, and (ii) looking for a negative reduced cost column, *i.e.* that is any solution of the pricing with objective function greater than 1 (*FIRST*). The combination of these alternatives give four strategies, herein called respectively *Dummy-OPT*, *Dummy-FIRST*, *Heuristic-OPT*, and *Heuristic-FIRST*.

In the two *FIRST* strategies, the pricing models of the FPRA and the PCRA problems need an additional constraints to limit the search space to negative reduced cost column:

$$\sum_{(i,j)\in\mathcal{L}} \bar{\sigma}_{ij} z_{ij} \geq 1 + \epsilon \tag{20}$$

which becomes for the PRCRA problem:

$$\sum_{w\in\mathcal{W}} \sum_{(i,j)\in\mathcal{L}} T_w \bar{\sigma}_{ij} z_{ijw} \geq 1 + \epsilon \tag{21}$$

In this way, the column generation algorithm stops when the pricing with the additional constraint (20) (or (21)) becomes infeasible.

Let us remark that the duality theory demands for *a* negative reduced cost column and not for the *most* negative. That is the pricing is effectively a feasibility problem. Therefore, it can be tackled with techniques other than Integer Linear Programming, and the next paragraphs presents how to model and solve the pricing problems using the Constraint Programming technique.

# 5 Constraint Programming approach

Constraint Programming (CP) is a programming paradigm for solving combinatorial search problems. CP systems combine expressive modeling languages with efficient solver implementations. The two basic solving techniques of constraint programming are constraint propagation and constraint labeling. Constraint propagation is an efficient inference mechanism aiming at reducing the domains of the problem variables by exploiting the semantic of the problem constraints. When constraint propagation reaches a fixed point, *i.e.*, roughly speaking, it is no more able to reduce the domains of the variables, the constraint labeling comes into play. Constraint labeling splits a problem into complementary cases, and it performs a tree-based search on each of them by branching new constraints which will trigger again constraint propagation. At each step of the constraint labeling, the CP solver has to select a still undetermined variable, and a value for that variable to *label* with. The order used to label over the variables does dramatically affect the size of the tree search, as a consequence of constraint propagation. By iterating constraint propagation and labeling, the solver will eventually determine the solutions of a problem.

Differently from mathematical programming, CP models can use a wide range of constraints over both numeric and symbolic variables. Even if in the following we use only a few types of constraints (namely, linear expressions, Boolean formulas, and the `element` constraint), CP can admit polynomials,

trigonometric functions, max, min, abs, and many other functions. In CP, it is also possible to associate with any constraint a Boolean variable that is true if and only if the constraint holds true. In this case the constraint is called *reified*. Notice that this can be achieved with mathematical programming using a big-M formulation, but this deteriorates in most cases the efficiency and the stability of the mathematical solvers.

Constraint Optimization Problems are solved naively in CP: during search a series of feasibility problems are solved, where each problem has a bounding constraint imposing that future solutions should be better than the best one found so far. A number of techniques have been proposed to enhance the solution of optimization problem, and we refer to [19] for a recent survey. A complete introduction to CP is out of the scope of this paper, and we refer to [2] for an introductory text book to, and to [18] for advanced topics on the integration of CP with Integer Programming. The CP-based column generation framework used in this paper was introduced in [11] for a crew assignment problem.

The main motivation for investigating the use of Constraint Programming for solving the pricing is to exploit its strength in solving highly constrained feasibility problems. The following subsection present how CP is used to model the pricing FPRA, PCRA, and PRCRA problems.

## 5.1 Basic version: fixed transmission power, and no rate adaptation

Let us describe how the pricing (7)–(10) can be modeled as Constraint Optimization Problem. The CP model has the following variables: for every link $(i, j)$ there is a Boolean variable $x_{ij}$ indicating whether node $i$ is transmitting to node $j$. Every node $i$ has associated two Boolean variables $receiver_i$ and $transmitter_i$ indicating whether the node $i$ is playing the role of receiver or transmitter, and a finite domain integer variable $noise_i$ to sum up the interference at node $i$. A finite domain integer variable *cost* is used for the solution cost. More formally, the variables are:

$$x_{ij} \in \{0, 1\} \qquad \forall (i, j) \in \mathcal{L} \tag{22}$$

$$transmitter_i \in \{0, 1\} \qquad \forall i \in \mathcal{N} \tag{23}$$

$$receiver_i \in \{0, 1\} \qquad \forall i \in \mathcal{N} \tag{24}$$

$$noise_i \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{N} \tag{25}$$

$$cost \in \mathbb{Z}^+ \tag{26}$$

The constraints modeling the FPRA pricing problem are:

$$transmitter_i = \sum_{(i,j)\in\mathcal{L}} x_{ij} \qquad \forall i \in \mathcal{N} \qquad (27)$$

$$receiver_i = \sum_{(j,i)\in\mathcal{L}} x_{ji} \qquad \forall i \in \mathcal{N} \qquad (28)$$

$$\neg\left(receiver_i \wedge transmitter_i\right) \qquad \forall i \in \mathcal{N} \qquad (29)$$

$$x_{ij} \Rightarrow \left(G_{ij}\, P \geq noise_j\right) \qquad \forall (i,j) \in \mathcal{L} \quad (30)$$

$$noise_j = \left(\frac{\gamma}{1+\gamma}\right)\left(\eta + \sum_{l\in\mathcal{N}\setminus\{j\}} G_{lj}\, P\, transmitter_l\right) \qquad \forall j \in \mathcal{N} \qquad (31)$$

Constraint (27) sets the Boolean variable $transmitter_i$ to one (*i.e.*, true) if node $i$ is transmitting to another node, and impose that node $i$ can transmit at most to another node. Similarly, constraint (28) sets the Boolean variable $receiver_i$ to one if node $i$ is receiving a transmission, and impose that node $i$ can receive at most from another node. Constraints (29) impose that every node is either transmitter or receiver, or none. Note that constraints (27)–(29) are equivalent to constraints (8). Constraints (30) are the interference constraints: if node $i$ is transmitting to node $j$, it implies that the quantity $noise_j$ at the receiver must be smaller than the quantity computed by constraint (31), with $P$ being the constant transmission power for every transmitter $l$. Constraints (31) accumulate in the variables $noise_j$ the thermal noise $\eta$ and the signals received at every node $j$. Remark that constraints (30) and (31) are equivalent to constraints (9).

The objective function is handled in the following way: the variable *cost* is used in the additional constraint:

$$cost = \sum_{(i,j)\in\mathcal{L}} \bar{\sigma}_{ij} x_{ij} \qquad (32)$$

and the objective function is rewritten as:

$$\min\left(1 - cost\right) = 1 + \max cost$$

that is, the solution has negative reduced cost if and only if $cost > 1$.

## 5.2 Extensions to power control and rate adaptation

**Extension to power control.** To model the pricing for the problem with power control (PCRA), for every node $i$ is introduced a new finite domain integer variable for the transmission power:

$$p_i \in \{0, \ldots, P\} \qquad \forall i \in N \qquad (33)$$

New constraints are introduced to force variable $p_i$ being strictly positive if and only if node $i$ is transmitting. Constraints (30) and (31) are slightly modified

to replace the constant power $P$ with the corresponding variables:

$$transmitter_i \Leftrightarrow (p_i > 0) \qquad \forall i \in \mathcal{N} \tag{34}$$

$$x_{ij} \Rightarrow (G_{ij}p_i \geq noise_j) \qquad \forall (i,j) \in \mathcal{L} \tag{35}$$

$$noise_j = \left(\frac{\gamma}{1+\gamma}\right)\left(\eta + \sum_{i \in \mathcal{N}\setminus\{j\}} G_{ij}p_i\right) \qquad \forall j \in \mathcal{N} \tag{36}$$

For the power control case, the objective function does not change with respect to the basic version.

**Extension to rate adaptation.** To formulate the pricing problem with rate adaptation, three new variables for each node $i$ are introduced: a finite domain integer variable $y_i$ gives the index of transmission rate used by transmitter $i$, which uniquely determines the threshold value $\gamma_i$ and the actual transmission rate $r_i$. The number of possible levels of transmission rates is denoted by $T$. Formally, these variables are:

$$y_i \in \{1, \ldots, T\} \qquad \forall i \in \mathcal{N} \tag{37}$$

$$\gamma_i \in \{\bar{\gamma}_1, \ldots, \bar{\gamma}_T\} \qquad \forall i \in \mathcal{N} \tag{38}$$

$$r_i \in \{\bar{r}_1, \ldots, \bar{r}_T\} \qquad \forall i \in \mathcal{N} \tag{39}$$

Note that threshold $\gamma$ in constraints (36) is replaced by variables $\gamma_i$:

$$noise_j = \left(\frac{\gamma_j}{1+\gamma_j}\right)\left(\eta + \sum_{i \in \mathcal{N}\setminus\{j\}} G_{ij}p_i\right) \qquad \forall j \in \mathcal{N} \tag{40}$$

The variables $y_i$ are used as indexing variable in `element` constraints. The constraints $\texttt{element}([c_1, \cdots, c_n], y, z)$, where `y` and `z` are variables and $[c_1, \cdots, c_n]$ are constants, requires that $z = c_y$. This constraint used as follows:

$$\texttt{element}([\bar{\gamma}_1, \ldots, \bar{\gamma}_T], y_i, \gamma_i) \qquad \forall i \in \mathcal{N} \tag{41}$$

$$\texttt{element}([\bar{r}_1, \ldots, \bar{r}_T], y_i, r_i) \qquad \forall i \in \mathcal{N} \tag{42}$$

The variables $r_i$ are used in the objective function to determine the number of packets sent in the communication pattern as follows:

$$cost = \sum_{(i,j) \in \mathcal{L}} \bar{\sigma}_{ij} x_{ij} r_i \tag{43}$$

# 6 Computational results

This section presents the computational results obtained over a set of random instances, and compares the MILP-based and the CP-based column generation over the three problems. The MILP-based column generation is implemented

using Ampl-Cplex version 10.0, and the CP-based column generation is implemented with the Gecode constraint environment [12]. Gecode is a research tool developed by an academic research group, and it used to experiment new ideas on Constraint Programming [20], while CPLEX is an outstanding commercial software.

Let us remark that we do tune the Cplex parameters to solve the pricing, since its default setting are far from being satisfactory. First we noticed that the best results in terms of computational time are obtained by switch the cut generation off (`mipcuts=-1`), thus, is a sense, by forcing Cplex to apply a basic branch-and-bound. We force Cplex to use the dual simplex algorithm for solving the continuous relaxation of the pricing (*i.e.*, Cplex option `mipalgorithm=2`); we set emphasis to search for feasible solution (`mipemphasis=1`); we disable the Cplex rounding heuristic at the branching nodes (`heuristicfreq=-1`), switch the relaxation induced neighborhood search heuristic off (`rinsheur=-1`). Since the pricing may be numerically unstable, we also set the following parameters: `feasibility=1e-09`, `integrality=1e-09`, and `mipgap=1e-09`. Finally, to improve the stability of column generation we do not re-optimize the restricted master problem, but we solve it from scratch at each iteration of the column generation. In this way, we reduce the chance of getting stuck in a degenerate solution of the master (see [13] for additional numerical results). Besides Cplex setting, we have to compute tight values of the big-M parameters used in linearizing the SINR constraints.

In the CP-based column generation, the restricted master problem is still solved with Cplex 10.0, while the pricing is solved using the Gecode constraint programming system. While solving the pricing problem with CP, we have observed that the solutions found during column generation are very much alike. That is, there is a tendency to activate the communication links coming first (*i.e.*, with low indexes) in the vector implementing variables $z$. This does not help the column generation algorithm, and induces a high number of iterations (slow convergence). To overcome this tendency, we have experimented a static *shuffled* order of the vector of variables $z$ before starting the CP depth first search. The effect of the static *shuffled* order on the CP-based column generation is impressive: it reduces dramatically the number of column generation iterations required. The reduction of the number of iterations is due to the diversity of the solutions found by successive calls to the CP pricing sub-problem [13].

The instances of Wireless Mesh Networks considered here have nodes randomly located into a square area of $700\text{m} \times 700\text{m}$. A pair of nodes can transmit if their distance is below half of the area diagonal. The set of instances have 10 and 15 nodes, with 10, 15, 20 or 25 randomly generated traffic demand $\mathcal{D}$. The constant parameters are set as follows: $P = 0.002425$ mW, $\eta = 10^{-11}$ mW, and $G_{ij} = (\text{distance}_{ij})^{-3}$. The values used for $\gamma_w$ are $\{2.0, 2.8, 7.1, 15.9\}$, and respectively for $T_w$ are $\{1, 2, 4, 8\}$[1].

Tables 1, 2, and 3 gives the computational results for the FPRA, the PCRA

[1] All the instances are available for download at `http://home.dei.polimi.it/gualandi/resources/`.

| Instance | | Dummy Cols | | | | Heuristic Cols | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *OPT* | | *FIRST* | | *OPT* | | *FIRST* | |
| $\|\mathcal{N}\|$ | $\|\mathcal{D}\|$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ |
| 10 | 10 | 34.2 | 2.5 | 38.8 | 1.5 | 7.2 | 0.6 | 6.6 | **0.3** |
| | 15 | 43.4 | 4.9 | 53.6 | 2.8 | 12 | 1.5 | 9.8 | **0.6** |
| | 20 | 40.2 | 4.1 | 50.4 | 3.0 | 11.6 | 1.4 | 13.4 | **1.0** |
| | 25 | 34.4 | 2.1 | 41 | 2.0 | 6 | **0.4** | 6.6 | **0.4** |
| 15 | 10 | 100.8 | 112 | 153.2 | 40 | 32.6 | 36 | 65.2 | **27** |
| | 15 | 125.8 | 124 | 215.4 | 75 | 57.8 | 69 | 111.4 | **55** |
| | 20 | 134.4 | 134 | 205.6 | 74 | 64 | 81 | 108.8 | **54** |
| | 25 | 124.8 | 129 | 227.8 | 129 | 57.8 | **69** | 100.2 | 80 |

Table 1: Computational results for the FPRA problem using the MILP-based column generation.

| Instance | | Dummy Cols | | | | Heuristic Cols | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *OPT* | | *FIRST* | | *OPT* | | *FIRST* | |
| $\|\mathcal{N}\|$ | $\|\mathcal{D}\|$ Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | |
| 10 | 10 | 62.6 | 5.2 | 98.2 | 3.8 | 17 | 1.5 | 19.2 | **1.1** |
| | 15 | 55 | 5.9 | 87.4 | 4.2 | 8 | **0.8** | 13.4 | 1.0 |
| | 20 | 62 | 8 | 104.4 | 7.5 | 14.6 | 2.5 | 19.8 | **2.2** |
| | 25 | 38.2 | 3.5 | 104.4 | 7.5 | 14.6 | 2.5 | 19.8 | **2.2** |
| 15 | 10 | 196.4 | 547.5 | 380.6 | 78.6 | 89.6 | 320.2 | 171.2 | **60.3** |
| | 15 | 186.4 | 514.6 | 389.4 | 123.7 | 97.4 | 349.5 | 222.6 | **11.4** |
| | 20 | 185 | 467.5 | 400.4 | 147.3 | 95.4 | 326.8 | 215.4 | **121.7** |
| | 25 | 222.6 | 813.9 | 505 | 314.9 | 136.6 | 632.1 | 304.8 | **262.5** |

Table 2: Computational results for the PCRA problem using the MILP-based column generation.

and PRCRA problems comparing the different strategies for solving the MILP-based column generation. The first two columns of each table give the number of nodes $|\mathcal{N}|$ and the number of demands $|\mathcal{D}|$. For each pair of values, we have considered five instances. The following columns give, for every combination of the strategies, the average number of iterations of column generation, *Iter*, and the average computation time, $T_{LP}$, in seconds. We do not distinguish between master and pricing computation time, since the second takes at least 95% of the time.

Table 1 shows that for the FPRA problem the strategy *FIRST* obtains the fastest computational time when used both with the *Dummy* and the *Heuristic* set of initial feasible columns, being *Heuristic-FIRST* the fastest. The *OPT* strategy is slower because at each iteration spends a lot of time in solving the pricing to optimality. As an advantage, it requires less iterations, which turns out to be better when solving the integer problem, as it generates less variables (see Section 6). Note that for this problem, the instances with 10 nodes are solved within few seconds, the instances with 15 nodes within a couple of minutes.

Table 2 shows the results for the PCRA problem. The difference between the *FIRST* and the *OPT* strategy is more evident, and the *Heuristic-FIRST* is the fastest again. For this problem, the instances of 10 nodes are again solved

| Instance | | Dummy Cols | | | | Heuristic Cols | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | *OPT* | | *FIRST* | | *OPT* | | *FIRST* | |
| $\|\mathcal{N}\|$ | $\|\mathcal{D}\|$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ | Iter | $T_{LP}$ |
| 10 | 10 | 22.8 | 3.2 | 30.6 | 1.7 | 6 | 0.8 | 9 | **0.6** |
| | 15 | 45.8 | 10.9 | 57 | 3.9 | 11.8 | 3.1 | 13 | **1.5** |
| | 20 | 32.4 | 5.3 | 51.6 | 3.6 | 12.6 | 3 | 20.4 | **2.2** |
| | 25 | 22.2 | 2.7 | 33.2 | 1.9 | 4.2 | 0.6 | 5 | **0.5** |
| 15 | 10 | 130.2 | 841.9 | 185 | 63.4 | 80.2 | 441.7 | 110.2 | **51.8** |
| | 15 | 87.4 | 284.6 | 159 | **42.7** | 46.6 | 187.8 | 90.6 | 46.1 |
| | 20 | 106.2 | 738.5 | 225.2 | **89.2** | 76.8 | 665.2 | 138.8 | 98.2 |
| | 25 | 96.6 | 911.4 | 186.8 | **97.4** | 45.2 | 495.8 | 96.8 | 111 |

Table 3: Computational results for the PRCRA problem using the MILP-based column generation.

within few seconds. For the instances of 15 nodes, using the *FIRST* strategies yield an average computational time of at most 5 minutes, while for the *OPT* strategies it ranges from 7 to 10 minutes.

Table 3 gives the computational results for the PRCRA problem. Differently from the two previous problems, the best average computation time for the bigger instances of 15 nodes is achieved by the *Dummy-FIRST* strategy. The two *OPT* strategies have average computation time of 6 and 11 minutes, but the worst computation time exceed 1000 seconds, making this strategy not applicable.

The results reported in Tables 1, 2, and 3 show that the strategy of stopping the pricing procedure once the first feasible solution is found, *i.e.*, of looking at the pricing problem as a *feasibility* problem, is winning. This justifies our approach of formulating and solving the pricing with Constraint Programming. For the CP-based column generation we have experimented the same strategies as for the MILP-based. Indeed, the more interesting results are obtained when using the *FIRST* strategy, and we report here only those results. Differently from Cplex, the CP solver has neither big-M values nor parameters to be set.

Table 4 compares the number of iterations and the computation time of the MILP-based and CP-based column generation for the *FIRST* strategy applied to problem FPRA. For all the instances the CP-based column generation is faster. In particular for the instances with 15 nodes is one order of magnitude faster, while requiring in average the same number of iterations. Table 5 refers to the PCRA problem. In this case, the CP-based column generation is faster for the instance with 10 nodes, while the two column generation approaches have similar computation times for the instances with 15 nodes. Finally, Table 6 reports the computational results for the PRCRA problem. For this problem the MILP-based solves in less computation time almost all the instances. Note that the number of iterations is not as different as the computational time: CP is slower in solving the pricing sub-problem. This is due to poor constraint propagation of constraints (40), (41), (42) along with the cost constraint (43). Having the SINR thresholds $\gamma_j$ as variables makes the interference constraints (40) less tight, and, as a consequence, constraint propagation is less effective.

For the sake of clarity, Table 7 reports the average, the minimum, and the

maximum computation time comparing the two column generation approaches. For the instances with 15 nodes, it is clear that CP is the fastest for the FPRA problem, while it is the slowest for the PRCRA problem. For the PCRA the two approaches are comparable. Let us remark again, that the results for the CP-based approach are obtained using an accademic constraint solver as a black-box, while the MILP-based approach exploits a commercial solver with accurate tunig of the parameters.

# 7 Solving the Integer Problem

The column generation approaches presented in the previous sections provide lower bounds, as they solve the continuous relaxation of the FPRA, PCRA, and PRCRA problems. When the solution obtained with column generation is integer, then it is the optimum; otherwise, we compute an integer upper bound by solving the integer problem over the final set of generated configurations. The gap between the two bounds is used to estimate the distance to the optimum objective value.

Table 8 shows the results for the solution of the integer master problem built with all columns obtained at the end of column generation, using the *Heuristic-FIRST* strategy. The first three columns describe again the problem instances, while the following columns report: the optimum of the continuous relaxation $Z_{LP}$, the upper bound obtained by solving the integer master problem $Z_{IP}$, the computation time for solving the integer master $T_{IP}$ in seconds, and the integrality gap $\frac{Z_{IP}-Z_{LP}}{Z_{LP}}$. The solution of the restricted integer master has a time out of 3600 seconds.

For the first two problems, namely FPRA and PCRA, the lower bound is extremely tight: the gap is always of 0%, and for many instances, being integer, the lower bound is the optimum. The computational time are very small for all the instances, and it is at most of 2.6 seconds. For the PRCRA problem, the gap is still acceptable, ranging from 2% to 8%, but the computational time increases, and reaches the timeout for half of the instances.

When using the set of configurations obtained by solving the column generation with the *Heuristic-FIRST* strategy, we end in having a high number of columns. This makes the integer master problem though. For this reason, we have implemented a heuristic to eliminate the columns that during column generations seems to be the worst, and do not appear in the basis of the last solved restricted master problem. For each column, both corresponding to variables (5) and (6), we sum at each iteration the values of its reduced cost up. At the end of column generation we sort the columns by increasing value of these accumulated reduced costs, and we keep a percentage of the best columns. This percentage depends on a parameter $th$: when it is set to 1.0, all the columns are kept; if it is set to 0.3, only 30% of the overall columns are kept. Table 9 shows the computational results over the instances with 10 nodes, and values of the threshold equal to 1.0, 0.5, and 0.3. While for $th = 1.0$, 12 instances out of 20 reach the time out, by setting $th = 0.3$, only 2 instances out of 20 does

reach the time out. The results for the instances with 15 nodes (here omitted) are very similar, though obtained with smaller values of $th$. Indeed, to achieve this decrease of computation time, we have to pay an slight increase of the gap. However, the gap stays always below 10%.

Finally, let us remark that we have limited our computational experiments to instances up to 15 nodes only for comparison reasons, since with this network size all versions of the problem can be solved in reasonable time. However, in the case of fixed power and rate we can tackle bigger instances. For instance, Table 10 shows the results obtained with 25 nodes and 25 demands using the CP-FIRST strategy and the dummy set of initial columns.

# 8 Conclusions

In this paper, we presented a column generation based approach for a resource allocation problem arising in managing wireless mesh networks. It can be applied to different versions of the considered problem. The proposed approach provides both lower and upper bounds and tackles real life size instances in reasonable computational time. The approach turns out to provide good quality heuristic solutions as the gap is always under 10%.

Further, we propose a comparison between MILP-based and CP-based column generation: in fact the pricing problem is solved both via MILP programming, through the commercial CPLEX solver, or via CP models, using the free and open-source Gecode environment. The behavior of the two approaches depends on the considered problems: MILP based column generation turns out to be more efficient on the PRCRA problem, the CP based approach on the FPRA problem.

The proposed results provided by the hybrid approach confirm that CP can be usefully applied to solve problems different from those usually considered in the literature, namely crew scheduling and vehicle routing with time windows.

The proposed approach provides an efficient tool for tackling real life instances. However, it might be further improved by designing custom and efficient cost based filtering algorithm to boost the CP performance, or ad hoc exact MILP based approaches for solving the pricing subproblem.

# References


[1] E. Amaldi, A. Capone, M. Cesana, and F. Malucelli. On the design of wireless mesh networks. In *Proceedings of International Network Optimization Conference*, 2007.

[2] K. R. Apt. *Principles of Constraint Programming*. Cambridge University Press, 2003.

[3] A. Behzad and I. Rubin. Multiple access protocol for power-controlled wireless access nets. *IEEE Transaction on Mobile Computing*, 3(4):307–316, 2004.

[4] A. Behzad and I. Rubin. Optimum integrated link scheduling and power control for ad hoc wireless networks. *IEEE Transactions on Vehicular Technology*, 56(1):194–205, January 2007.

[5] R. Bhatia and M. Kodialam. On power efficient communication over multi-hop wireless networks: joint routing, scheduling, and power control. In *Proceedings of IEEE INFOCOM*, 2004.

[6] P. Bjorklund, P. Varbrand, and Di Yuan. A column generation method for spatial tdma scheduling in ad hoc networks. *Ad-Hoc Networks*, 2(4):405–418, 2004.

[7] A. Capone and G. Carello. Scheduling optimization in wireless mesh networks with power control and rate adaptation. In *IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, 2006.

[8] R. Casaquite and W. Hwang. Evaluation and optimization of joint scheduling, power control, and routing in ad hoc wireless networks. In *Proceedings of IEEE ICHIT*, 2006.

[9] R. L. Cruz and A. V. Santhanam. Optimal routing, link scheduling and power control in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, 2003.

[10] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transaction on Wireless Communications*, 3(1):74–85, 2004.

[11] Torsten Fahle, Ulrich Junker, Stefan E. Karisch, Niklas Kohl, Meinolf Sellmann, and Bo Vaaben. Constraint programming based column generation for crew assignment. *J. Heuristics*, 8(1):59–81, 2002.

[12] Gecode Team. Gecode: Generic constraint development environment, 2006. Available from `http://www.gecode.org`.

[13] S. Gualandi. *Enhancing CP-based column generation for Integer Programs.* PhD thesis, Politecnico di Milano, 2008. forthcoming.

[14] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transaction on Information Theory*, 46(2):388–404, 2000.

[15] M. Johansson and L. Xiao. Cross-layer optimization of wireless networks using nonlinear column generation. *IEEE Transaction on Wireless Communications*, 2(5):435–445, 2006.

[16] G. Kulkarni, V. Raghunathan, and M. Srivastava. Joint end-to-end scheduling, power control and rate control in multi-hop wireless networks. In *Proceedings of IEEE GLOBECOM*, 2004.

[17] Y. Li and A. Ephremides. Joint scheduling, power control and routing algorithm for ad-hoc wireless networks. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 2005.

[18] M. Milano. *Constraint and Integer Programming: Toward a Unified Methodology*. Kluwer Academic Publishers, Norwell, MA, USA, 2003.

[19] M. Milano and M. Wallace. Integrating operations research in constraint programming. *4OR*, 4(3):1–45, 2006.

[20] C. Schulte and P.J. Stuckey. Speeding up constraint propagation. In Mark Wallace, editor, *Tenth International Conference on Principles and Practice of Constraint Programming*, volume 3258 of *Lecture Notes in Computer Science*, pages 619–633, Toronto, Canada, September 2004. Springer-Verlag.

[21] J. Tang, G. Xue, C. Chandler, and W.Zhang. Link scheduling with power control for throughput enhancement in multihop wireless networks. *IEEE Transaction on Vehicular Technology*, 3(55):733–742, 2006.

[22] H. Viswanathan and S. Mukherjee. Throughput-range tradeoff of wireless mesh backhaul networks. *IEEE Journal on Selelected Area in Communications*, 3(24):593–602, 2006.

| Instance | | | Dummy Cols | | | | | | Heuristic Cols | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Cplex-FIRST | | | CP-FIRST | | | Cplex-FIRST | | | CP-FIRST | | |
| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ |
| 10 | 60 | 10 | 57 | 117 | 2.2 | 75 | 135 | 0.3 | 12 | 158 | 0.5 | 14 | 160 | 0.1 |
| 10 | 68 | 10 | 40 | 108 | 1.6 | 63 | 131 | 0.2 | 13 | 166 | 0.7 | 16 | 169 | 0.1 |
| 10 | 78 | 10 | 48 | 126 | 2.1 | 57 | 135 | 0.2 | 6 | 143 | 0.2 | 5 | 142 | 0.0 |
| 10 | 72 | 10 | 25 | 97 | 0.8 | 30 | 102 | 0.1 | 1 | 151 | 0.0 | 2 | 152 | 0.0 |
| 10 | 78 | 10 | 24 | 102 | 0.8 | 25 | 103 | 0.1 | 1 | 155 | 0.0 | 1 | 155 | 0.0 |
| 10 | 74 | 15 | 62 | 136 | 3.2 | 71 | 145 | 0.3 | 3 | 216 | 0.2 | 1 | 214 | 0.0 |
| 10 | 64 | 15 | 46 | 110 | 1.9 | 62 | 126 | 0.2 | 16 | 145 | 0.8 | 14 | 143 | 0.1 |
| 10 | 78 | 15 | 23 | 101 | 0.9 | 36 | 114 | 0.1 | 1 | 199 | 0.0 | 1 | 199 | 0.0 |
| 10 | 78 | 15 | 58 | 136 | 3.0 | 62 | 140 | 0.2 | 8 | 147 | 0.4 | 19 | 158 | 0.2 |
| 10 | 74 | 15 | 79 | 153 | 4.8 | 100 | 174 | 0.5 | 21 | 208 | 1.6 | 33 | 220 | 0.3 |
| 10 | 82 | 20 | 61 | 143 | 3.9 | 61 | 143 | 0.3 | 13 | 199 | 1.0 | 13 | 199 | 0.1 |
| 10 | 64 | 20 | 46 | 110 | 2.5 | 66 | 130 | 0.3 | 25 | 198 | 2.1 | 32 | 205 | 0.2 |
| 10 | 66 | 20 | 36 | 102 | 1.8 | 54 | 120 | 0.1 | 4 | 190 | 0.2 | 2 | 188 | 0.0 |
| 10 | 74 | 20 | 50 | 124 | 3.0 | 55 | 129 | 0.2 | 5 | 131 | 0.3 | 3 | 129 | 0.1 |
| 10 | 72 | 20 | 59 | 131 | 3.7 | 62 | 134 | 0.3 | 20 | 173 | 1.6 | 32 | 185 | 0.2 |
| 10 | 74 | 25 | 27 | 101 | 1.4 | 23 | 97 | 0.1 | 1 | 217 | 0.1 | 1 | 217 | 0.1 |
| 10 | 56 | 25 | 37 | 93 | 1.5 | 52 | 108 | 0.2 | 8 | 127 | 0.4 | 7 | 126 | 0.1 |
| 10 | 56 | 25 | 43 | 99 | 1.8 | 46 | 102 | 0.1 | 4 | 167 | 0.2 | 3 | 166 | 0.0 |
| 10 | 66 | 25 | 50 | 116 | 3.0 | 59 | 125 | 0.2 | 6 | 181 | 0.4 | 11 | 186 | 0.1 |
| 10 | 64 | 25 | 48 | 112 | 2.4 | 43 | 107 | 0.2 | 14 | 176 | 0.9 | 12 | 174 | 0.1 |
| 15 | 190 | 10 | 156 | 346 | 51 | 248 | 438 | 2.7 | 95 | 640 | 49 | 143 | 688 | 4.3 |
| 15 | 176 | 10 | 223 | 399 | 80 | 299 | 475 | 5.8 | 125 | 761 | 63 | 148 | 784 | 5.0 |
| 15 | 166 | 10 | 95 | 261 | 16 | 107 | 273 | 0.5 | 24 | 636 | 5 | 24 | 636 | 0.3 |
| 15 | 162 | 10 | 154 | 316 | 30 | 246 | 408 | 1.8 | 32 | 613 | 8 | 51 | 632 | 0.9 |
| 15 | 150 | 10 | 138 | 288 | 22 | 245 | 395 | 2.2 | 50 | 755 | 9 | 80 | 785 | 0.6 |
| 15 | 168 | 15 | 188 | 356 | 70 | 244 | 412 | 4.0 | 106 | 665 | 57 | 146 | 705 | 3.5 |
| 15 | 184 | 15 | 294 | 478 | 163 | 282 | 466 | 7.8 | 174 | 818 | 133 | 121 | 765 | 3.9 |
| 15 | 130 | 15 | 204 | 334 | 48 | 214 | 344 | 3.5 | 142 | 650 | 40 | 143 | 651 | 2.8 |
| 15 | 144 | 15 | 237 | 381 | 70 | 244 | 388 | 3.9 | 100 | 644 | 34 | 121 | 665 | 3.9 |
| 15 | 124 | 15 | 154 | 278 | 25 | 133 | 257 | 0.9 | 35 | 456 | 9 | 34 | 455 | 1.0 |
| 15 | 142 | 20 | 232 | 374 | 84 | 243 | 385 | 4.9 | 179 | 749 | 92 | 153 | 723 | 4.1 |
| 15 | 162 | 20 | 197 | 359 | 74 | 260 | 422 | 5.5 | 64 | 668 | 33 | 92 | 696 | 4.3 |
| 15 | 164 | 20 | 299 | 463 | 143 | 252 | 416 | 5.5 | 140 | 838 | 96 | 108 | 806 | 4.5 |
| 15 | 124 | 20 | 154 | 278 | 32 | 180 | 304 | 4.1 | 76 | 537 | 19 | 76 | 537 | 2.1 |
| 15 | 142 | 20 | 146 | 288 | 38 | 226 | 368 | 3.9 | 85 | 598 | 28 | 103 | 616 | 3.7 |
| 15 | 170 | 25 | 272 | 442 | 192 | 252 | 422 | 6.4 | 132 | 704 | 127 | 127 | 699 | 6.3 |
| 15 | 166 | 25 | 170 | 336 | 83 | 191 | 357 | 2.9 | 34 | 595 | 26 | 25 | 586 | 1.3 |
| 15 | 162 | 25 | 236 | 398 | 114 | 262 | 424 | 5.6 | 140 | 705 | 87 | 187 | 752 | 6.1 |
| 15 | 168 | 25 | 266 | 434 | 186 | 305 | 473 | 10.4 | 143 | 786 | 136 | 129 | 772 | 6.5 |
| 15 | 142 | 25 | 195 | 337 | 70 | 212 | 354 | 2.7 | 52 | 640 | 25 | 72 | 660 | 1.5 |

Table 4: Comparison between MILP-based and CP-based column generation: FPRA problem.

| Instance | | | Dummy Cols | | | | | | Heuristic Cols | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Cplex-FIRST | | | CP-FIRST | | | Cplex-FIRST | | | CP-FIRST | | |
| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ |
| 10 | 60 | 10 | 102 | 162 | 3.3 | 87 | 147 | 0.85 | 22 | 222 | 0.9 | 33 | 233 | 1.3 |
| 10 | 68 | 10 | 122 | 190 | 5.9 | 136 | 204 | 4.55 | 34 | 288 | 2.3 | 38 | 292 | 1.8 |
| 10 | 78 | 10 | 87 | 165 | 3.2 | 114 | 192 | 2.07 | 24 | 220 | 1.2 | 22 | 218 | 0.5 |
| 10 | 72 | 10 | 86 | 158 | 3.1 | 84 | 156 | 1.1 | 2 | 238 | 0.1 | 2 | 238 | 0.3 |
| 10 | 78 | 10 | 94 | 172 | 3.7 | 109 | 187 | 2.33 | 14 | 268 | 0.7 | 19 | 273 | 0.5 |
| 10 | 74 | 15 | 83 | 157 | 4.1 | 96 | 170 | 2.9 | 8 | 302 | 0.6 | 19 | 313 | 2.2 |
| 10 | 64 | 15 | 106 | 170 | 5.0 | 113 | 177 | 1.7 | 33 | 245 | 2.1 | 39 | 251 | 1.2 |
| 10 | 78 | 15 | 77 | 155 | 3.6 | 110 | 188 | 2.9 | 4 | 256 | 0.4 | 6 | 258 | 0.9 |
| 10 | 78 | 15 | 76 | 154 | 3.4 | 78 | 156 | 1.0 | 7 | 238 | 0.5 | 5 | 236 | 0.4 |
| 10 | 74 | 15 | 95 | 169 | 4.9 | 117 | 191 | 1.4 | 15 | 335 | 1.2 | 26 | 346 | 1.0 |
| 10 | 82 | 20 | 140 | 222 | 13.3 | 163 | 245 | 3.7 | 38 | 374 | 5.0 | 41 | 377 | 3.8 |
| 10 | 64 | 20 | 98 | 162 | 5.7 | 84 | 148 | 0.6 | 12 | 266 | 0.9 | 5 | 259 | 0.1 |
| 10 | 66 | 20 | 78 | 144 | 4.1 | 71 | 137 | 0.4 | 1 | 259 | 0.1 | 1 | 259 | 0.1 |
| 10 | 74 | 20 | 100 | 174 | 6.9 | 122 | 196 | 2.6 | 17 | 243 | 1.8 | 29 | 255 | 2.2 |
| 10 | 72 | 20 | 106 | 178 | 7.5 | 121 | 193 | 1.8 | 31 | 286 | 3.0 | 46 | 301 | 1.8 |
| 10 | 74 | 25 | 24 | 98 | 13.3 | 33 | 107 | 0.2 | 38 | 374 | 5.0 | 1 | 337 | 0.1 |
| 10 | 56 | 25 | 59 | 115 | 5.7 | 72 | 128 | 0.5 | 12 | 266 | 0.9 | 16 | 270 | 0.4 |
| 10 | 56 | 25 | 32 | 88 | 4.1 | 54 | 110 | 0.4 | 1 | 259 | 0.1 | 9 | 267 | 0.3 |
| 10 | 66 | 25 | 40 | 106 | 6.9 | 43 | 109 | 0.3 | 17 | 243 | 1.8 | 13 | 239 | 0.7 |
| 10 | 64 | 25 | 36 | 100 | 7.5 | 69 | 133 | 0.4 | 31 | 286 | 3.0 | 2 | 257 | 0.1 |
| 15 | 190 | 10 | 432 | 622 | 115 | 454 | 644 | 285 | 211 | 1311 | 89 | 283 | 1383 | 243 |
| 15 | 176 | 10 | 362 | 538 | 80 | 434 | 610 | 235 | 163 | 1333 | 60 | 255 | 1425 | 164 |
| 15 | 166 | 10 | 347 | 513 | 57 | 463 | 629 | 155 | 143 | 1256 | 36 | 264 | 1377 | 145 |
| 15 | 162 | 10 | 387 | 549 | 79 | 436 | 598 | 187 | 197 | 1228 | 77 | 246 | 1277 | 286 |
| 15 | 150 | 10 | 375 | 525 | 62 | 417 | 567 | 269 | 142 | 1255 | 40 | 221 | 1334 | 237 |
| 15 | 168 | 15 | 408 | 576 | 130 | 432 | 600 | 221 | 234 | 1201 | 127 | 252 | 1219 | 189 |
| 15 | 184 | 15 | 577 | 761 | 274 | 542 | 726 | 558 | 391 | 1547 | 273 | 300 | 1456 | 450 |
| 15 | 130 | 15 | 201 | 331 | 29 | 280 | 410 | 24 | 62 | 1056 | 14 | 94 | 1088 | 8 |
| 15 | 144 | 15 | 402 | 546 | 104 | 393 | 537 | 164 | 194 | 1160 | 74 | 238 | 1204 | 117 |
| 15 | 124 | 15 | 359 | 483 | 81 | 357 | 481 | 126 | 232 | 1058 | 75 | 228 | 1054 | 209 |
| 15 | 142 | 20 | 356 | 498 | 113 | 362 | 504 | 254 | 161 | 1110 | 83 | 227 | 1176 | 181 |
| 15 | 162 | 20 | 447 | 609 | 191 | 396 | 558 | 166 | 249 | 1346 | 139 | 305 | 1402 | 317 |
| 15 | 164 | 20 | 497 | 661 | 242 | 385 | 549 | 262 | 301 | 1559 | 235 | 239 | 1497 | 336 |
| 15 | 124 | 20 | 401 | 525 | 100 | 293 | 417 | 75 | 196 | 1038 | 74 | 217 | 1059 | 79 |
| 15 | 142 | 20 | 301 | 443 | 91 | 337 | 479 | 83 | 170 | 1199 | 77 | 209 | 1238 | 102 |
| 15 | 170 | 25 | 640 | 810 | 422 | 427 | 597 | 301 | 305 | 1331 | 287 | 232 | 1258 | 379 |
| 15 | 166 | 25 | 513 | 679 | 355 | 409 | 575 | 284 | 330 | 1436 | 303 | 242 | 1348 | 263 |
| 15 | 162 | 25 | 464 | 626 | 269 | 440 | 602 | 348 | 285 | 1258 | 241 | 288 | 1261 | 255 |
| 15 | 168 | 25 | 498 | 666 | 326 | 464 | 632 | 412 | 276 | 1404 | 253 | 272 | 1400 | 350 |
| 15 | 142 | 25 | 410 | 552 | 202 | 389 | 531 | 180 | 328 | 1428 | 228 | 265 | 1365 | 130 |

Table 5: Comparison between MILP-based and CP-based column generation: PCRA problem.

| Instance | | | Dummy Cols | | | | | | Heuristic Cols | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Cplex-FIRST | | | CP-FIRST | | | Cplex-FIRST | | | CP-FIRST | | |
| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ | Iter | Cols | $T_{LP}$ |
| 10 | 60 | 10 | 21 | 81 | 0.9 | 82 | 142 | 4.7 | 19 | 426 | 0.9 | 18 | 425 | 1.4 |
| 10 | 68 | 10 | 23 | 91 | 1.2 | 92 | 160 | 4.8 | 10 | 472 | 0.8 | 42 | 504 | 5.6 |
| 10 | 78 | 10 | 28 | 106 | 1.6 | 78 | 156 | 6.4 | 2 | 544 | 0.2 | 3 | 545 | 0.5 |
| 10 | 72 | 10 | 44 | 116 | 2.3 | 52 | 124 | 2.4 | 12 | 362 | 0.9 | 6 | 356 | 0.3 |
| 10 | 78 | 10 | 37 | 115 | 2.5 | 70 | 148 | 3.5 | 2 | 567 | 0.3 | 9 | 574 | 0.7 |
| 10 | 74 | 15 | 104 | 178 | 8.5 | 179 | 253 | 16.3 | 25 | 587 | 3.5 | 17 | 579 | 2.9 |
| 10 | 64 | 15 | 41 | 105 | 1.9 | 84 | 148 | 4.6 | 14 | 394 | 1.0 | 12 | 392 | 1.0 |
| 10 | 78 | 15 | 40 | 118 | 2.4 | 105 | 183 | 4.2 | 1 | 535 | 0.2 | 1 | 535 | 0.3 |
| 10 | 78 | 15 | 50 | 128 | 3.3 | 118 | 196 | 7.6 | 17 | 427 | 1.8 | 14 | 424 | 2.1 |
| 10 | 74 | 15 | 50 | 124 | 3.4 | 124 | 198 | 10.2 | 8 | 713 | 1.1 | 5 | 710 | 2.4 |
| 10 | 82 | 20 | 57 | 139 | 4.5 | 79 | 161 | 4.4 | 23 | 633 | 2.9 | 2 | 612 | 0.8 |
| 10 | 64 | 20 | 61 | 125 | 3.4 | 72 | 136 | 2.1 | 18 | 319 | 1.6 | 20 | 321 | 2.1 |
| 10 | 66 | 20 | 71 | 137 | 5.3 | 117 | 183 | 5.5 | 37 | 518 | 3.7 | 28 | 509 | 3.1 |
| 10 | 74 | 20 | 29 | 103 | 2.0 | 51 | 125 | 4.9 | 2 | 487 | 0.4 | 4 | 489 | 1.3 |
| 10 | 72 | 20 | 40 | 112 | 2.9 | 107 | 179 | 5.3 | 22 | 493 | 2.3 | 29 | 500 | 5.0 |
| 10 | 82 | 25 | 24 | 98 | 1.5 | 52 | 134 | 1.3 | 1 | 568 | 0.1 | 1 | 568 | 0.2 |
| 10 | 64 | 25 | 19 | 75 | 0.9 | 32 | 96 | 0.9 | 9 | 318 | 0.7 | 11 | 320 | 0.8 |
| 10 | 66 | 25 | 46 | 102 | 2.7 | 87 | 153 | 2.7 | 10 | 425 | 0.9 | 19 | 434 | 1.4 |
| 10 | 74 | 25 | 41 | 107 | 2.7 | 71 | 145 | 4.6 | 4 | 449 | 0.5 | 16 | 461 | 2.1 |
| 10 | 72 | 25 | 36 | 100 | 1.8 | 77 | 149 | 5.9 | 1 | 392 | 0.1 | 1 | 392 | 0.1 |
| 15 | 190 | 10 | 112 | 302 | 26 | 190 | 380 | 174 | 94 | 2028 | 32 | 108 | 2042 | 201 |
| 15 | 176 | 10 | 170 | 346 | 77 | 564 | 740 | 465 | 66 | 1681 | 46 | 128 | 1743 | 565 |
| 15 | 166 | 10 | 200 | 366 | 75 | 612 | 778 | 893 | 125 | 2290 | 63 | 169 | 2334 | 735 |
| 15 | 162 | 10 | 199 | 361 | 89 | 561 | 723 | 478 | 91 | 1549 | 68 | 474 | 1932 | 596 |
| 15 | 150 | 10 | 244 | 394 | 50 | 669 | 819 | 749 | 175 | 2281 | 50 | 300 | 2406 | 497 |
| 15 | 168 | 15 | 174 | 342 | 62 | 528 | 696 | 532 | 104 | 1787 | 69 | 116 | 1799 | 306 |
| 15 | 184 | 15 | 104 | 288 | 30 | 451 | 635 | 771 | 49 | 1829 | 51 | 118 | 1898 | 426 |
| 15 | 130 | 15 | 206 | 336 | 61 | 422 | 552 | 489 | 153 | 1182 | 54 | 161 | 1190 | 290 |
| 15 | 144 | 15 | 139 | 283 | 31 | 374 | 518 | 294 | 90 | 1485 | 40 | 147 | 1542 | 322 |
| 15 | 124 | 15 | 172 | 296 | 30 | 361 | 485 | 111 | 57 | 981 | 17 | 97 | 1021 | 58 |
| 15 | 142 | 20 | 227 | 369 | 137 | 636 | 778 | 971 | 155 | 1910 | 107 | 229 | 1984 | 789 |
| 15 | 162 | 20 | 216 | 378 | 70 | 445 | 607 | 677 | 89 | 2399 | 65 | 118 | 2428 | 435 |
| 15 | 164 | 20 | 199 | 363 | 70 | 501 | 665 | 474 | 109 | 1909 | 91 | 242 | 2042 | 711 |
| 15 | 124 | 20 | 266 | 390 | 95 | 439 | 563 | 405 | 164 | 1231 | 104 | 251 | 1318 | 489 |
| 15 | 142 | 20 | 218 | 360 | 75 | 359 | 501 | 345 | 177 | 1301 | 124 | 194 | 1318 | 441 |
| 15 | 170 | 25 | 224 | 394 | 189 | 513 | 683 | 1166 | 114 | 1608 | 154 | 186 | 1680 | 1155 |
| 15 | 166 | 25 | 211 | 377 | 74 | 458 | 624 | 641 | 137 | 1809 | 107 | 224 | 1896 | 392 |
| 15 | 162 | 25 | 148 | 310 | 67 | 386 | 548 | 874 | 54 | 2077 | 50 | 80 | 2103 | 398 |
| 15 | 168 | 25 | 177 | 345 | 86 | 478 | 646 | 1345 | 96 | 2086 | 146 | 164 | 2154 | 1230 |
| 15 | 142 | 25 | 174 | 316 | 71 | 452 | 594 | 279 | 83 | 1264 | 98 | 157 | 1338 | 213 |

Table 6: Comparison between MILP-based and CP-based column generation: PRCRA problem.

| | | | FPRA | | | PCRA | | | PRCRA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Strategy | Solver | $\mathcal{N}$ | Average | Min | Max | Average | Min | Max | Average | Min | Max |
| Dummy | Cplex | 10 | 2.3 | 0.8 | 4.8 | 5.8 | 3.1 | 13.3 | 2.8 | 0.9 | 8.5 |
| | CP | 10 | 0.2 | 0.1 | 0.5 | 1.6 | 0.2 | 4.6 | 5.1 | 0.9 | 16.3 |
| Heuristic | Cplex | 10 | 0.6 | 0.02 | 2.1 | 1.6 | 0.1 | 5.0 | 1.2 | 0.1 | 3.7 |
| | CP | 10 | 0.1 | 0.01 | 0.3 | 1.0 | 0.1 | 3.8 | 1.7 | 0.1 | 5.6 |
| Dummy | Cplex | 15 | 80 | 16 | 192 | 166 | 29 | 422 | 73 | 26 | 189 |
| | CP | 15 | 4.1 | 0.5 | 10.4 | 229 | 24 | 558 | 607 | 111 | 1345 |
| Heuristic | Cplex | 15 | 54 | 5 | 136 | 139 | 14 | 303 | 77 | 17 | 154 |
| | CP | 15 | 4.5 | 0.9 | 6.5 | 222 | 8 | 450 | 513 | 58 | 1230 |

Table 7: Average computation times.

| Instance | | | FPRA | | | | PCRA | | | | PRCRA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | $Z_{LP}$ | $Z_{IP}$ | $T_{IP}$ | Gap | $Z_{LP}$ | $Z_{IP}$ | $T_{IP}$ | Gap | $Z_{LP}$ | $Z_{IP}$ | $T_{IP}$ | Gap |
| 10 | 60 | 10 | 180.0 | 180 | 0.02 | 0% | 158.3 | 159 | 0.04 | 0% | 46.4 | 49 | 2.2 | 4% |
| 10 | 68 | 10 | 223.0 | 223 | 0.02 | 0% | 177.3 | 178 | 0.05 | 0% | 45.1 | 48 | 115.9 | 4% |
| 10 | 78 | 10 | 201.3 | 202 | 0.02 | 0% | 132.0 | 132 | 0.03 | 0% | 37.5 | 39 | 0.2 | 3% |
| 10 | 72 | 10 | 242.0 | 242 | 0.02 | 0% | 190.4 | 191 | 0.03 | 0% | 42.8 | 45 | 9.4 | 5% |
| 10 | 78 | 10 | 236.0 | 236 | 0.01 | 0% | 170.8 | 171 | 0.04 | 0% | 42.9 | 46 | timeout | 7% |
| 10 | 74 | 15 | 440.0 | 440 | 0.03 | 0% | 406.8 | 407 | 0.05 | 0% | 62.6 | 65 | 820.9 | 3% |
| 10 | 64 | 15 | 448.0 | 448 | 0.03 | 0% | 365.1 | 366 | 0.04 | 0% | 76.4 | 79 | timeout | 3% |
| 10 | 78 | 15 | 449.0 | 449 | 0.02 | 0% | 367.2 | 368 | 0.05 | 0% | 56.3 | 59 | 24.1 | 4% |
| 10 | 78 | 15 | 377.3 | 378 | 0.04 | 0% | 338.3 | 339 | 0.05 | 0% | 53.0 | 56 | 671.1 | 6% |
| 10 | 74 | 15 | 414.5 | 415 | 0.04 | 0% | 343.3 | 344 | 0.08 | 0% | 63.6 | 66 | timeout | 3% |
| 10 | 82 | 20 | 585.0 | 585 | 0.06 | 0% | 485.2 | 486 | 0.13 | 0% | 90.2 | 94 | timeout | 3% |
| 10 | 64 | 20 | 593.3 | 594 | 0.05 | 0% | 509.0 | 509 | 0.05 | 0% | 83.9 | 87 | timeout | 4% |
| 10 | 66 | 20 | 544.0 | 544 | 0.03 | 0% | 482.0 | 482 | 0.08 | 0% | 79.4 | 82 | timeout | 3% |
| 10 | 74 | 20 | 593.0 | 593 | 0.05 | 0% | 459.3 | 460 | 0.07 | 0% | 91.0 | 94 | timeout | 3% |
| 10 | 72 | 20 | 572.3 | 573 | 0.05 | 0% | 466.5 | 467 | 0.11 | 0% | 87.5 | 91 | timeout | 3% |
| 10 | 74 | 25 | 852.0 | 852 | 0.04 | 0% | 485.2 | 486 | 0.13 | 0% | 111.0 | 116 | timeout | 5% |
| 10 | 56 | 25 | 741.5 | 742 | 0.04 | 0% | 509.0 | 509 | 0.05 | 0% | 125.6 | 128 | 45.7 | 2% |
| 10 | 56 | 25 | 796.0 | 796 | 0.04 | 0% | 482.0 | 482 | 0.08 | 0% | 111.9 | 115 | timeout | 3% |
| 10 | 66 | 25 | 714.0 | 714 | 0.05 | 0% | 459.3 | 460 | 0.07 | 0% | 103.4 | 107 | timeout | 3% |
| 10 | 64 | 25 | 701.3 | 702 | 0.05 | 0% | 466.5 | 467 | 0.11 | 0% | 110.9 | 113 | timeout | 2% |
| 15 | 190 | 10 | 218.7 | 219 | 0.09 | 0% | 183.5 | 184 | 0.56 | 0% | 48.8 | 52 | timeout | 6% |
| 15 | 176 | 10 | 186.1 | 187 | 0.15 | 0% | 155.1 | 156 | 0.33 | 0% | 45.0 | 48 | 91 | 4% |
| 15 | 166 | 10 | 206.0 | 206 | 0.06 | 0% | 145.9 | 146 | 0.18 | 0% | 41.3 | 43 | 292 | 2% |
| 15 | 162 | 10 | 195.5 | 196 | 0.16 | 0% | 142.3 | 143 | 0.51 | 0% | 36.5 | 39 | 1 | 5% |
| 15 | 150 | 10 | 181.0 | 181 | 0.07 | 0% | 159.1 | 160 | 0.29 | 0% | 35.8 | 38 | 4 | 6% |
| 15 | 168 | 15 | 389.5 | 390 | 0.22 | 0% | 319.6 | 320 | 0.51 | 0% | 66.1 | 69 | 279.0 | 3% |
| 15 | 184 | 15 | 383.7 | 384 | 0.31 | 0% | 325.4 | 326 | 0.53 | 0% | 66.5 | 70 | timeout | 4% |
| 15 | 130 | 15 | 348.7 | 349 | 0.25 | 0% | 291.0 | 291 | 0.31 | 0% | 56.6 | 59 | 53 | 4% |
| 15 | 144 | 15 | 382.4 | 383 | 0.12 | 0% | 316.3 | 317 | 0.64 | 0% | 64.6 | 69 | timeout | 6% |
| 15 | 124 | 15 | 571.5 | 572 | 0.13 | 0% | 376.7 | 377 | 0.43 | 0% | 88.9 | 92 | timeout | 3% |
| 15 | 142 | 20 | 596.0 | 596 | 0.20 | 0% | 521.1 | 522 | 0.41 | 0% | 96.7 | 101 | 581 | 4% |
| 15 | 162 | 20 | 568.2 | 569 | 0.20 | 0% | 432.5 | 433 | 0.77 | 0% | 99.1 | 103 | timeout | 3% |
| 15 | 164 | 20 | 569.2 | 570 | 0.21 | 0% | 455.9 | 456 | 0.74 | 0% | 86.9 | 92 | timeout | 6% |
| 15 | 124 | 20 | 631.3 | 632 | 0.15 | 0% | 472.7 | 473 | 0.94 | 0% | 99.0 | 103 | timeout | 3% |
| 15 | 142 | 20 | 513.5 | 514 | 0.19 | 0% | 406.5 | 407 | 0.69 | 0% | 90.8 | 96 | timeout | 5% |
| 15 | 170 | 25 | 631.1 | 632 | 1.37 | 0% | 519.6 | 520 | 2.60 | 0% | 107.1 | 112 | timeout | 4% |
| 15 | 166 | 25 | 853.7 | 854 | 0.40 | 0% | 645.3 | 646 | 1.16 | 0% | 135.6 | 140 | timeout | 3% |
| 15 | 162 | 25 | 568.7 | 569 | 0.54 | 0% | 471.5 | 472 | 1.33 | 0% | 97.8 | 101 | timeout | 3% |
| 15 | 168 | 25 | 673.3 | 674 | 0.56 | 0% | 582.5 | 583 | 0.79 | 0% | 113.8 | 117 | timeout | 3% |
| 15 | 142 | 25 | 639.6 | 640 | 0.29 | 0% | 522.2 | 523 | 1.64 | 0% | 105.8 | 109 | timeout | 3% |

Table 8: Integer upper bounds.

| Instance | | | th=1.0 | | th=0.5 | | th=0.3 | |
|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | $T_{IP}$ | Gap | $T_{IP}$ | Gap | $T_{IP}$ | Gap |
| 10 | 60 | 10 | 2.22 | 4.3% | 0.69 | 4.3% | 0.01 | 6.5% |
| 10 | 68 | 10 | 115.87 | 4.4% | 20.92 | 4.4% | 0.01 | 6.7% |
| 10 | 78 | 10 | 0.19 | 2.7% | 0.03 | 2.7% | 0.01 | 2.7% |
| 10 | 72 | 10 | 9.44 | 4.7% | 0.32 | 7.0% | 0.01 | 7.0% |
| 10 | 78 | 10 | timeout | | 8.69 | 7.0% | 0.01 | 7.0% |
| 10 | 74 | 15 | 820.86 | 3.2% | 592.85 | 3.2% | 0.36 | 4.8% |
| 10 | 64 | 15 | timeout | | 179.73 | 2.6% | 1.7 | 3.9% |
| 10 | 78 | 15 | 24.13 | 3.6% | 1.91 | 3.6% | 0.01 | 5.3% |
| 10 | 78 | 15 | 671.09 | 5.7% | 2290.48 | 7.6% | 0.01 | 9.4% |
| 10 | 74 | 15 | timeout | | timeout | | 2.81 | 4.7% |
| 10 | 82 | 20 | timeout | | 258.12 | 2.2% | 0.1 | 4.4% |
| 10 | 64 | 20 | timeout | | 1449.46 | 3.6% | 0.01 | 4.8% |
| 10 | 66 | 20 | timeout | | timeout | | 42.31 | 5.0% |
| 10 | 74 | 20 | timeout | | timeout | | 0.01 | 6.6% |
| 10 | 72 | 20 | timeout | | timeout | | 0.07 | 5.7% |
| 10 | 74 | 25 | timeout | | timeout | | timeout | |
| 10 | 56 | 25 | 45.69 | 1.6% | 22.44 | 1.6% | 2.49 | 4.8% |
| 10 | 56 | 25 | timeout | | timeout | | 2.88 | 5.4% |
| 10 | 66 | 25 | timeout | | 1875.89 | 2.9% | 0.04 | 4.8% |
| 10 | 64 | 25 | timeout | | timeout | | timeout | |

Table 9: Reducing the number of columns in the final integer problem.

| $\|\mathcal{N}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{D}\|$ | Iter | Cols | $Z_{LP}$ | $Z_{IP}$ | $T_{LP}$ | $T_{IP}$ | Gap |
|---|---|---|---|---|---|---|---|---|---|
| 25 | 434 | 25 | 1472 | 1916 | 176.9 | 177 | 1667 | 6.0 | 0% |
| 25 | 412 | 25 | 1255 | 1667 | 223.6 | 224 | 428 | 1.0 | 0% |
| 25 | 420 | 25 | 1351 | 1771 | 237.4 | 238 | 1580 | 28.6 | 0% |
| 25 | 452 | 25 | 1633 | 2085 | 217.4 | 218 | 974 | 3.6 | 0% |
| 25 | 442 | 25 | 1340 | 1782 | 242.6 | 243 | 452 | 12.7 | 0% |

Table 10: Problem FPRA. Big instances solved with the CP-FIRST strategy.