EURO Journal on Computational Optimization manuscript No.
(online first)

Speeding Up Column Generation for Robust Wireless Network Planning

Grit Claßen · Arie M.C.A. Koster · Anke Schmeink

This is a preprint version. The final publication is available at http://www.springerlink.com, DOI: 10.1007/s13675-013-0013-0.

Abstract The wireless network planning problem consists of base station placement and traffic node assignment to base stations. To incorporate traffic demand uncertainties, we follow the Γ-robustness approach by Bertsimas and Sim. In this paper, we develop a branch-and-price algorithm, with the aim to enhance the solution process and improve the dual bounds. Instead of assigning individual traffic nodes to base stations, subsets of traffic nodes are assigned to a base station, implying the pricing problem essentially being a robust knapsack. Since a straightforward implementation does not give satisfactory results, we present techniques, which we apply to the master problem as well as to the pricing problems, to improve the performance. We investigate the effectiveness of these techniques in an extensive computational study.

Keywords branch-and-price · column generation · wireless network planning · robust optimisation

Grit Claßen
Lehrstuhl II für Mathematik and UMIC Research Centre, RWTH Aachen University, 52056 Aachen, Germany
Tel.: +49-241-8020753
Fax: +49-241-8022730
E-mail: classen@umic.rwth-aachen.de
Corresponding author

Arie M.C.A. Koster
Lehrstuhl II für Mathematik, RWTH Aachen University, 52056 Aachen, Germany
E-mail: koster@math2.rwth-aachen.de

Anke Schmeink
UMIC Research Centre, RWTH Aachen University, 52056 Aachen, Germany
E-mail: schmeink@umic.rwth-aachen.de

1 Introduction

The optimal planning of wireless networks of the third generation (3G) has attracted a great deal of attention during the last decade, see for instance, Amaldi et al (2003); Siomina et al (2006); Amaldi et al (2008), and remains a crucial and complex problem in the future not least because corresponding optimisation problems belong to the class of NP-hard problems, e.g. (Amaldi et al 2003; Glaßer et al 2005). Wireless networks of the fourth generation (4G) such as the Long Term Evolution (LTE) or LTE Advanced (3rd Generation Partnership Project 2012) utilise a couple of sophisticated techniques such as Orthogonal Frequency Division Multiple Access (OFDMA) for downlink (DL) or Single-Carrier Frequency Division Multiple Access (SC-FDMA) for uplink to overcome the resource restrictions of 3G networks (Dahlman et al 2008). Hence, an optimal planning which respects the modified requirements of future wireless networks, see e.g., Gordejuela-Sanchez and Zhang (2009); Engels et al (2010); Siomina and Yuan (2012), is inevitable to fully utilise the gains of these techniques. Besides the consideration of OFDMA etc., also the energy efficient planning (Boiardi et al 2012; El-Beaino et al 2012) comes into focus. Furthermore, significantly increasing user demands (Cisco Systems 2012) impair the planning problem. The bit rate requirements increase since the user behaviour of mobile customers shifts from ordinary telephony or short message services towards data transfer such as web browsing, data download, broadcasting or Voice-over-IP (VoIP).

Another aspect which should be considered already in the planning of a wireless network are non-deterministic factors, e.g., user mobility, fluctuating bit rate requirements and channel conditions. To handle such uncertainties, robust optimisation is a recently proposed technique. For uncertain factors with an unknown probability distribution, Bertsimas and Sim (2003, 2004) introduced the Γ-robustness approach which limits the number of uncertain entries by a robustness parameter Γ. The application of this concept does not significantly increase the complexity of the problem.

A straightforward (compact) formulation of the (robust) wireless network planning problem consists of a huge number of variables since, e.g., one variable for each base station-traffic node (BS-TN) pair is needed, and can have a weak linear program (LP) solution. A prominent procedure to tackle these problems is to reformulate the model via a column generation approach, see Lübbecke and Desrosiers (2005) for a survey. For the column generation formulation, the solution process starts with a subset of variables (columns) and only variables having the potential to improve the objective are generated on the fly. This method can significantly improve the LP solution compared to the compact model.

In this paper, we develop a branch-and-price (B&P) algorithm for the robust planning of wireless networks which is based on the integer linear program (ILP) presented in Claßen et al (2011, 2013). The model considers DL data transmission and guarantees a certain link quality while the inter-cell interference is limited. Since variables of this problem are integer, we have to develop problem specific branching rules to be able to price further variables after leaving the root node of the branch-and-bound (B&B) tree.

The main contribution of this paper besides the presentation of a novel B&P algorithm are the performance improvements presented in Section 3 which are necessary since a straightforward implementation does not give satisfactory results. First, some general applied settings such as a good initial solution and cutting planes are stated. Furthermore, we introduce a lower bound, which is used as a stop criterion for the solving of a single B&B node, and techniques to speed up the pricing problems (PPs), which compute further columns. Finally, we adjust the number of added columns per pricing round and present a primal heuristic to compute better primal bounds. The performance of these techniques is analysed in a computational study using six test instances of different dimensions. By means of this study, we are able to reveal the best setting for the B&P approach for the robust wireless network planning problem among the evaluated strategies.

The remainder of this paper is organised as follows. In Section 2, we present some preliminaries and the complete B&P algorithm for the planning of a Γ-robust wireless network, which includes the master problem (MP), the restricted master problem (RMP), the PPs, and problem specific branching rules. The techniques to speed up the column generation are presented in Section 3 and their performance is evaluated in a computational study in Section 4. Section 5 concludes the paper with some final remarks.

2 Problem Formulation

In this section, we briefly describe the problem at hand as well as the Γ-robust approach by Bertsimas and Sim (2004) and give a motivation for the column generation approach. Furthermore, we introduce the MP, the RMP, and the PPs based on the robust model of the wireless network planning problem presented in Claßen et al (2011, 2013). To obtain a full B&P algorithm, we develop problem specific branching rules in the last subsection.

2.1 Problem Statement

Given a geographical area and the task to design a new wireless network infrastructure, both locations for antennae and traffic estimations have to be defined. A possible site location and a configuration are consolidated in a BS candidate site $s \in S$ which entails costs c_s and provides a total DL bandwidth b_s. Traffic demands, i.e., bit rate requirements, of users in a small area are accumulated in a traffic node $t \in T$ which has a demand w_t. To include the demand uncertainties in the problem formulation, following the approach by Bertsimas and Sim (2004), we model the demand values as symmetric and bounded random variables w_t taking values in the interval $[\bar{w}_t - \hat{w}_t, \bar{w}_t + \hat{w}_t]$. Here, $\bar{w}_t$ denotes the nominal demand and $\hat{w}_t$ the highest deviation. Hence, the peak demand of TN t is $\bar{w}_t + \hat{w}_t$.

Table 1 Overview of the system parameters

Description of Parameters	**Notation**
set of BS candidates	S
costs of BS s	c_s
total DL bandwidth of BS s	b_s
conflict graph of conflicting BSs	$G = (S, E)$
set of TNs	T
nominal demand of TN t	$\bar{w}_t$
highest demand deviation of TN t	$\hat{w}_t$
DL spectral efficiency from s to t	e_{st}
min. required spectral efficiency	$e_{\min}$
max. allocated bandwidth	$\frac{\bar{w}_t+\hat{w}_t}{e_{st}}$

The intra-cell interference for DL in 4G networks is negligible due to OFDMA whereas the inter-cell interference is limited by means of a conflict graph $G = (S, E)$. The concept of conflict or interference graphs has been applied, e.g., in the planning of GSM (Global System for Mobile Communications) (Mathar and Niessen 2000) networks, WLANs (Riihijarvi et al 2005), and LTE networks (Engels et al 2011) and in a modified way via complement sets for the deployment of cooperation clusters in wireless cellular networks (Niu et al 2012).

Two BSs are adjacent in G if they cannot be deployed at the same time. Hence, all installed BSs are obliged to constitute an independent set in G. We would like to point out that the conflict graph is a quite general concept to model different levels of interference. Hence, it is possible that even for BSs in a maximum independent set of the conflict graph inter-cell interference occurs to some extent depending on the definition of the conflict graph.

We guarantee a certain link quality by requiring a minimum spectral efficiency $e_{\min}$ for each BS-TN link. The spectral efficiency, denoted by e_{st} ($s \in S$, $t \in T$), gives the ratio between required data rate and allocated bandwidth. It incorporates, among other things, the modulation and coding scheme that is supported by the associated signal-to-noise ratio (SNR). We include the constraint on the spectral efficiency in the following auxiliary sets.

$$\begin{aligned} S * T &:= \{(s,t) \in S \times T \ : e_{st} \geq e_{\min}\}, \\ S_t &:= \{s \in S \ : (s,t) \in S * T\} && \forall t \in T, \\ T_s &:= \{t \in T \ : (s,t) \in S * T\} && \forall s \in S. \end{aligned}$$

A summary of the system parameters is given in Table 1.

From a financial point of view, the costs caused by the installed BSs are an important factor. Hence, the objective is to minimise the total costs of the network while minimising the number of not covered TNs. These two objectives are combined via a scaling parameter λ. However, if we regard the costs c_s of a BS as the consumed power, the minimisation of the total costs is equivalent to the minimisation of the total power consumption of the network which is also an important factor from an ecological prospect. Thus, “costs” should be regarded as a generalised term and can be adapted to the specific aims of the network planning, possibly adjusting the scaling parameter λ.

Moreover, we introduce the robustness parameter $\Gamma \in \{0, \dots, |T|\}$ which limits the number of TNs deviating from their nominal value simultaneously (in the worst case towards the peak demand $\hat{w}_t + \bar{w}_t$).

For each BS $s \in S$, we introduce a deployment indicator $x_s \in \{0, 1\}$ equal to 1 if the BS is installed. Moreover, we introduce an assignment variable $z_{st} \in \{0, 1\}$ denoting whether TN t is assigned to BS s. Finally, the binary slack variable u_t is equal to 1 if TN t is not served by any BS. Hence, the *original* or *compact* model of the robust wireless network planning problem can be stated as follows, see Claßen et al (2011, 2013) for details.

$$\min \quad \sum_{s \in S} c_s x_s + \lambda \sum_{t \in T} u_t \tag{1a}$$

$$\text{s.t.} \quad \sum_{s \in S_t} z_{st} + u_t = 1 \qquad \forall\, t \in T \tag{1b}$$

$$\sum_{s \in U} x_s \leq 1 \qquad \forall\, U \subset S,\ U \text{ max. clique} \tag{1c}$$

$$\sum_{t \in T_s} \frac{\bar{w}_t}{e_{st}} z_{st} + \max_{T' \subseteq T_s, |T'| \leq \Gamma} \sum_{t \in T'} \frac{\hat{w}_t}{e_{st}} z_{st} \leq b_s x_s \qquad \forall\, s \in S \tag{1d}$$

$$z_{st} \leq x_s \qquad \forall\, (s,t) \in S * T \tag{1e}$$

$$x_s,\ z_{st},\ u_t \in \{0,1\} \qquad \forall s,\ \forall\ (s,t),\ \forall\ t \tag{1f}$$

As stated before, the objective (1a) minimises the total costs and the number of not covered TNs combined by the scaling parameter λ. Constraints (1b) ensure that each TN is covered by at most one BS (hard handover). The independent set constraints $x_i + x_j \leq 1\ \forall ij \in E$ are strengthened by *maximal clique inequalities* (1c), i.e., at most one BS in a maximal clique of the conflict graph can be installed. The maximal cliques of the conflict graph are precomputed by the Bron-Kerbosch algorithm (Bron and Kerbosch 1973). Constraints (1d) are the non-linear robust capacity constraints guaranteeing that all nominal demands and the Γ-worst deviations of all TNs assigned to one BS do not exceed the capacity. Finally, constraints (1e) are the so-called *variable upper bound* constraints which guarantee that a TN can only be assigned to a BS if the BS is installed.

To linearise constraints (1d), we reformulate the maximisation term for a fixed $s \in S$ and a solution (x, z) as the following ILP.

$$\max \quad \sum_{t \in T_s} \frac{\hat{w}_t}{e_{st}} z_{st} \varphi_t \tag{2a}$$

$$\text{s.t.} \quad \sum_{t \in T_s} \varphi_t \leq \Gamma \tag{2b}$$

$$\varphi_t \in \{0,1\} \qquad \forall t \in T_s \tag{2c}$$

Introducing dual variables μ_s and ν_{st} and exploiting LP duality, constraints (1d) can be replaced by the (linear) robust counterpart

$$\sum_{t\in T_s} \frac{\bar{w}_t}{e_{st}} z_{st} + \Gamma \mu_s + \sum_{t\in T_s} \nu_{st} \le b_s x_s \quad \forall\, s \in S \tag{3}$$

and adding further constraints

$$\mu_s + \nu_{st} \ge \frac{\hat{w}_t}{e_{st}} z_{st} \qquad \forall t \in T_s \tag{4a}$$

$$\mu_s \ge 0,\, \nu_{st} \ge 0 \qquad \forall (s,t) \in S * T. \tag{4b}$$

For more details, see Claßen et al (2013).

There exist usually several arguments to apply a column generation approach rather than to solve the compact model. For the robust wireless network planning problem, one reason is the potential weakness of the LP relaxation of the compact model (1), i.e., we can compute a better LP solution by the column generation method than by the compact model which we present in the next subsections. To confirm this, we give a tiny non-robust example after the statement of the master problem at the end of Section 2.2. However, even the B&P algorithm with the best setting cannot compete with the compact model regarding the time consumption.

Another reason for a B&P algorithm is the decomposition of the compact model into master and pricing problems. In the compact formulation, there is a robust knapsack problem embedded (via the capacity constraints). Due to the decomposition described in Section 2.3, this robust knapsack problem is completely sourced out to the pricing problems. Hence, the MP is identical for all Γ values. Moreover, we can exploit all known approaches such as extended robust cover inequalities (see Section 3.1) to enhance the performance of solving the robust knapsack problem.

2.2 The Master Problem

We reformulate the compact model (1) via a Dantzig-Wolfe decomposition to obtain a column generation approach. For this purpose, we consider a set-wise assignment of TNs to BSs. For each BS $s \in S$ we introduce a set $\mathcal{T}_s$ which consists of all possible subsets of TNs that can be assigned to s:

$$\mathcal{T}_s = \{\tau \subseteq T_s : \text{all } t \in \tau \text{ can be assigned to } s \text{ simultaneously}\},$$

i.e., in particular the BS capacity is not exceeded. Obviously, $\mathcal{T}_s \subseteq 2^{T_s}$. The assignment variables are denoted by $\zeta_{s\tau}$ for $s \in S$ and $\tau \in \mathcal{T}_s$ with

$$\zeta_{s\tau} = \begin{cases} 1, & \text{set } \tau \subseteq T_s \text{ is assigned to BS } s \text{ (and } T_s \setminus \tau \text{ not)} \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, as in the compact model (1) we introduce a deployment indicator $x_s \in \{0,1\}$ for each BS $s \in S$, which is equal to 1 if the BS is installed, and a binary slack variable u_t, which is equal to 1 if TN t is not served by any BS. The following ILP represents the MP.

$$\min \sum_{s\in S} c_s x_s + \lambda \sum_{t\in T} u_t \tag{5a}$$

$$\text{s.t.} \sum_{s\in S_t} \sum_{\tau\in\mathcal{T}_s : t\in\tau} \zeta_{s\tau} + u_t = 1 \qquad \forall t\in T \tag{5b}$$

$$\sum_{s\in U} -x_s \geq -1 \qquad \forall U\subset S, U \text{ max. clique in } G \tag{5c}$$

$$x_s - \sum_{\tau\in\mathcal{T}_s} \zeta_{s\tau} \geq 0 \qquad \forall s\in S \tag{5d}$$

$$x_s,\ \zeta_{s\tau},\ u_t \in \{0,1\} \qquad \forall s\in S,\ \tau\in\mathcal{T}_s,\ t\in T \tag{5e}$$

The objective (5a) and the maximal clique inequalities (5c) have not changed compared to the compact model. Furthermore, constraints (5b) are the reformulated constraints (1b) and (5d) the reformulated variable upper bound constraints (1e).

Note, any restriction on the assignment of TNs to a BS (capacity constraints) is incorporated in the definition of the sets $\mathcal{T}_s$ and hence, does not occur in the MP.

In the ILP, it is sufficient to consider only "$\geq$", i.e.,

$$\sum_{s\in S_t} \sum_{\tau\in\mathcal{T}_s : t\in\tau} \zeta_{s\tau} + u_t \geq 1 \quad \forall t\in T \tag{6}$$

instead of the equality condition (5b). By this decision, it is sufficient to consider only maximal sets τ, i.e., no further TNs can be added without violating the capacity constraint. Furthermore, the definition of the variables as binary variables is not necessary. Instead it is sufficient to have

$$x_s,\ \zeta_{s\tau},\ u_t \in \mathbb{Z}_{\geq 0} \quad \forall s\in S,\ \tau\in\mathcal{T}_s,\ t\in T. \tag{7}$$

The upper bound of 1 for all variables is guaranteed by constraints (5c) and (5d) as well as the minimisation in the objective (5a).

As pointed out before, the LP solution computed by the column generation can be significantly better than the LP relaxation of the compact model (1). To demonstrate this, we give a tiny non-robust example with two BSs and three TNs, see Figure 1. Every BS has an available DL bandwidth of 40 and entails costs of 4000. We assume that the BSs are not interfering with each other so that no conflict graph exists. Every TN has a nominal demand of 30 and no deviation. We choose the spectral efficiencies such that TNs 0 and 1 can be assigned to BS 0 and TNs 1 and 2 to BS 1 represented by arrows in the figure. The scaling parameter λ is set to 10000.

The LP solution of the compact model is 8000 with assignment variables $z_{00} = z_{12} = 1$ and $z_{01} = z_{11} = 0.5$ installing both BSs. The column generation algorithm gives an LP solution of 18000 with exactly two assignment variables equal to one ($\zeta_{0\{0\}} = \zeta_{1\{2\}} = 1$), installing both BSs and TN 1

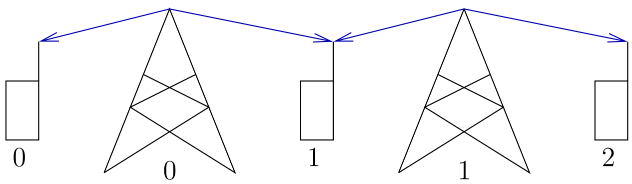

Fig. 1 An example consisting of two BSs and three TNs to demonstrate the improvement of the the LP solution by means of column generation.

remains uncovered. Since we have a minimisation problem, this is a much better lower bound. In fact, for this tiny example the LP solution of the column generation is the optimal integer solution.

The MP (5) describes the Γ-robust wireless network planning problem completely. However, for each BS there exists (potentially) an exponential number of sets $\tau \in \mathcal{T}_s$ resulting in a huge model. Hence in the following section, we restrict the MP to subsets $\mathcal{T}_s' \subseteq \mathcal{T}_s$ for each $s \in S$ obtaining the RMP and compute further necessary columns by PPs.

2.3 The Restricted Master Problem and the Pricing Problems

As stated before, the RMP does not consider the total amount of assignment variables at the outset. To decide which variable has to be added to the RMP, we can compute the reduced cost of $\zeta_{s\tau}$ for all $\tau \in \mathcal{T}_s \setminus \mathcal{T}_s'$ by means of dual variables.

We introduce dual variables α_s for each constraint (5d) and dual variables β_t for each constraint (6). The reduced cost of $\zeta_{s\tau}$ are computed via

$$0 - (-\alpha_s) - \sum_{t \in \tau} \beta_t.$$

The variable $\zeta_{s\tau}$ has to be added to the RMP if the reduced cost are negative, hence if

$$\sum_{t \in \tau} \beta_t > \alpha_s.$$

To detect variables with negative reduced cost, we introduce a PP for each BS. For $s \in S$ fixed, we introduce variables

$$a_t = \begin{cases} 1, & \text{TN } t \text{ is in the newly constructed set } \tau \in \mathcal{T}_s \setminus \mathcal{T}_s' \\ 0, & \text{otherwise} \end{cases}$$

for all TNs $t \in T_s$. The corresponding PP now reads as follows.

$$\max \sum_{t \in T_s} \tilde{\beta}_t a_t \tag{8a}$$

$$\text{s.t.} \sum_{t \in T_s} \frac{\bar{w}_t}{e_{st}} a_t + \Gamma \mu + \sum_{t \in T_s} \nu_t \leq b_s \tag{8b}$$

$$\mu + \nu_t \geq \frac{\hat{w}_t}{e_{st}} a_t \qquad \forall t \in T_s \tag{8c}$$

$$a_t \in \{0,1\},\ \mu, \nu_t \geq 0 \qquad \forall t \in T_s, \tag{8d}$$

where $\tilde{\beta}_t$ is the optimal value of the dual β_t of the current RMP and μ, ν_t are the dual variables introduced by reformulating the non-linear robust capacity constraints (1d) to constraints (3) and (4) leading to (8b) and (8c) in the PP. The index s is dropped as the BS is fixed in every PP. Note, these constraints form a robust knapsack problem.

If the objective value (8a) is greater than the optimal value $\tilde{\alpha}_s$, we have found a variable $\zeta_{s\tau}$ with negative reduced cost. Let $(\tilde{a}, \tilde{\mu}, \tilde{\nu})$ be an optimal solution of (8) with an objective value greater than $\tilde{\alpha}_s$. Then the new variable $\zeta_{s\tau}$ with $\tau = \{t \in T_s : \tilde{a}_t = 1\}$ is added to the RMP. The process, i.e., solving the RMP with new dual values and then solving the PPs to decide if further variables are needed, is repeated until no more variables with negative reduced cost exist.

2.4 Branching Rules

So far, the previous section just explains how to solve the LP relaxation of the MP by column generation. This LP solution is not necessarily integer. Hence, a branching process should be started.

If the branching is performed in a straightforward way without taking special care of the variables computed in a pricing problem, this can lead to problems like loops as explained in the following. Assume that we branch on variable $\zeta_{s\tau}$, i.e., we construct two child nodes with $\zeta_{s\tau} \leq 0$ and $\zeta_{s\tau} \geq 1$, respectively. Enforcing the ζ-decision in the first child node, corresponds to adding the additional constraint

$$\sum_{t \in \tau} a_t + \sum_{t \notin \tau} (1 - a_t) \leq |T_s| - 1$$

to the PP (8) corresponding to BS s. Hence, the PP consist of a robust knapsack problem with an additional constraint. Future branching decisions lead to more additional constraints which further destroy the structure of the PPs. On the other hand, if we do not enforce the ζ-decision, it is possible that we compute exactly the same set τ again when solving the PP corresponding to s. Hence, we would add $\zeta_{s\tau}$ afresh in this subtree ending in a loop. This is the reason why we have to develop problem specific branching rules as presented in this subsection.

The first branching rule we apply is to branch on the BS deployment indicator variables as long as there is a non-integer value $\tilde{x}_s$ in the current LP solution left. If there is no BS $s \in S$ left for which x_s is not integer, we start to branch on non-integer non-coverage indicator variables creating two child nodes with $u_t \leq 0$ and $u_t \geq 1$, respectively.

The third branching rule comes into operation when all values $\tilde{x}$ and $\tilde{u}$ are integer in the current LP solution. We then have to consider non-integer values of pricing variables ζ. As described above, we cannot create two child nodes based on the integrality criterion. Instead, we apply the common technique to branch on the variables of the original problem, see e.g. Barnhart et al (1998). The compact formulation (1) includes the assignment variables z_{st} which are equal to 1 if TN t is assigned to BS s. Branching on the original variables in the B&P approach therefore describes the generation of two child nodes with $z_{st} \leq 0$ and $z_{st} \geq 1$, respectively, if $\tilde{z}_{st} := \sum_{\tau \in \mathcal{T}'_s : t \in \tau} \tilde{\zeta}_{s\tau}$ is not integer in the current LP solution. In each branching step, we branch on the most fractional variable that is the variable closest to 0.5.

Hence, for a BS-TN pair (s, t) with a non-integer value $\tilde{z}_{st}$, we generate two child nodes containing constraints

$$\sum_{\tau \in \mathcal{T}'_s : t \in \tau} \zeta_{s\tau} \leq 0 \tag{9}$$

and

$$\sum_{\tau \in \mathcal{T}'_s : t \in \tau} \zeta_{s\tau} \geq 1, \tag{10}$$

respectively. Constraint (9) implicitly fixes every $\zeta_{s\tau}$ to 0 for all $\tau \in \mathcal{T}'_s$ with $t \in \tau$. This implies that only new variables with $a_t = 0$ can be beneficial in this subproblem. Constraint (10) guarantees that TN t is served by BS s. Enforcing this constraint, we have to consider the dual of (10) for the computation of the reduced cost of a new pricing variable for BS s. Hence, we have to introduce a dual variable for (10), multiply it by a_t and add this product to the objective function of the PP. To avoid the consideration of further dual variables, we instead reformulate constraint (10) as follows. By constraints (5b), $u_t = 0$ as TN t is assigned to BS s and further, $\zeta_{\bar{s}\tau} = 0$ for all BSs $\bar{s} \neq s$ and τ containing t in this child node. Hence, (10) is equivalent to

$$\sum_{\bar{s} \in S \setminus \{s\}} \sum_{\tau \in \mathcal{T}'_{\bar{s}} : t \in \tau} \zeta_{\bar{s}\tau} + u_t \leq 0. \tag{11}$$

Replacing (10) by (11), we do not have to consider the dual in the PP corresponding to s anymore and we can further reduce the number of variables by setting $a_t = 1$ in the PP corresponding to s and by setting $a_t = 0$ in the PPs corresponding to $\bar{s} \in S \setminus \{s\}$.

Note, in each node of the B&B tree, it is necessary to know the path to the root node, i.e., to know all constraints of type (9) and (11) that have been added on this path, to adjust the PPs. Additionally, in a B&B node containing constraints of type (9) or (11) subsequent computed variables have to be added to the corresponding constraints (respecting s and t).

Proposition 1 *The presented branching scheme is complete, i.e., all variables at every leaf of the complete B&B tree are integer.*

Proof It is directly evident that the variables x and u are integer at the leaves of the B&B tree as we branch if they are fractional. On the other hand, the assignment variables ζ are fixed to 0 if they occur in any of the constraints of type (9) or (11). However, constraints of type (9) or (11) do not explicitly forbid fractionality of the remaining $\zeta_{s\tau}$ variables. Thus, assume the original assignment variables z_{st} are integer but there exist a BS s and a set τ_1 with $\zeta_{s\tau_1}$ fractional. By integrality of z_{st}, it holds that $z_{st} = 1$ for all $t \in \tau_1$. Since

$$z_{st} = \sum_{\tau \in \mathcal{T}_s : t \in \tau} \zeta_{s\tau} = 1,$$

there must exist at least one set $\tau_2 \neq \tau_1$ containing t with $\zeta_{s\tau_2}$ fractional. W.l.o.g., $\tau_1 \setminus \tau_2 \neq \emptyset$ and $\zeta_{s\tau_1} + \zeta_{s\tau_2} = 1$ (for $\zeta_{s\tau_1} + \zeta_{s\tau_2} < 1$, we replace $\zeta_{s\tau_2}$ by the sum over all assignment variables $\zeta_{s\tau}$ with $\tau \in \mathcal{T}_s, t \in \tau$ and $\tau \neq \tau_1$). For every $t' \in \tau_1 \setminus \tau_2$ there must exist (at least) one set τ_3 containing t' with $\zeta_{s\tau_3}$ fractional but $t \notin \tau_3$. But then

$$\sum_{\tau \in \mathcal{T}_s : t \in \tau} \zeta_{s\tau} + \zeta_{s\tau_3} > 1,$$

which violates constraint (5d) for BS s as $x_s \leq 1$, a contradiction. □

Note that based on the values of the original assignment variables z_{st}, we can define one set of TNs $\tau_s := \{t \in T_s \mid z_{st} = 1\}$ per BS $s \in S$ such that $\zeta_{s\tau_s} = 1$ and $\zeta_{s\tau} = 0$ for all $\tau \in \mathcal{T}_s \setminus \{\tau_s\}$ should hold in the integer solution.

3 Performance Improvements

A straightforward implementation of the B&P algorithm presented in the previous section does not give satisfying results, i.e., many small test instances cannot be solved to optimality. Hence, we investigate several techniques to improve and to speed up the column generation for the robust wireless network planning problem in this section.

3.1 General Settings

In this subsection, we present general settings which we use for all computations.

Initial solution. For the initialisation of the column generation approach, a (dual) feasible initial solution of the MP is required (or infeasibility of the MP must be proved). The quality of the initial solution impacts the dual solution of the initial RMP and thus, also the quality of the lower and upper bounds for the optimal solution of the MP.

A promising initial solution can be computed by means of the LP relaxation of the original/compact formulation (1). Denote by $(\tilde{x}, \tilde{u}, \tilde{z})$ the optimal LP solution of the compact formulation, where z_{st} is an assignment variable as mentioned in Subsection 2.1. For every BS $s \in S$ with $\tilde{x}_s \neq 0$ sort the set of TNs with $\tilde{z}_{st} \neq 0$ such that $\tilde{z}_{s0} \geq \tilde{z}_{s1} \geq \ldots$. For some $n \geq 1$ it holds $\tilde{z}_{st} = \tilde{x}_s$ for the first n TNs due to the variable upper bound constraints $z_{st} \leq x_s$ in the compact formulation. Hence, let τ denote the set of these TNs:

$$\tau := \{t \in T_s : \ \tilde{z}_{st} = \tilde{x}_s\} = \{0, \ldots, n-1\}.$$

By scaling, we have $\tau \in \mathcal{T}_s$. Then, we consider the next TN n with $\tilde{z}_{sn} < \tilde{x}_s$. If the Γ-robust capacity constraint

$$\sum_{t \in \tau \cup \{n\}} \frac{\bar{w}_t}{e_{st}} + \max_{I \subseteq \tau \cup \{n\}, |I| \leq \Gamma} \sum_{t \in I} \frac{\hat{w}_t}{e_{st}} \leq b_s \tag{12}$$

is still valid, we add this TN to the set τ, i.e., $\tau = \tau \cup \{n\}$. We add the subsequent TNs one by one as long as the BS capacity is not exceeded. In this way, we create one appropriate (and as large as possible) initial column per $s \in S$.

Absolute gap limit. In a B&P algorithm, it is not obvious when the current primal bound is an optimal solution since the solving process cannot stop until no more pricing variables with negative reduced cost are found. Let PB and DB be the primal and dual bound of the MP, respectively. If $|PB - DB| <$`abs_gap`, with `abs_gap`$:= \gcd(\min_{s \in S} c_s, \lambda)$ for integer values of c_s and λ and gcd denotes the greatest common divisor, then there cannot be another integer solution between PB and DB. Therefore, we stop the solving process of the B&P formulation. Note, this absolute gap limit is automatically known by the solver for the compact formulation as all variables are present.

Aging. Since we compute many columns, it is possible that not all pricing variables are needed during the complete solving process. Hence, we mark the pricing variables as “removable” so that the corresponding column can be removed from the LP due to aging or cleanup which is automatically performed by the branch-and-price-and-cut framework SCIP (Achterberg 2009).

Cutting planes. Since the PPs (8) are robust knapsack problems, we can apply the so-called *extended robust cover inequalities* as presented in Claßen et al (2013) which represent cutting planes. We explain the main idea briefly.

A robust cover $(C \cup J) \subseteq T_s$ for BS s is a set of TNs for which holds

$$|J| \leq \Gamma,\ |C| \geq 0, C \cap J = \emptyset \text{ and } \sum_{t \in C} \frac{\bar{w}_t}{e_{st}} + \sum_{t \in J} \frac{\bar{w}_t + \hat{w}_t}{e_{st}} > b_s,$$

i.e., the sum of the demands of the TNs, including the deviation of up to Γ many demands from the nominal demand, in the robust cover exceeds the BS capacity. The following robust cover inequality for BS s is therefore a valid inequality of the robust knapsack problem in the corresponding PP.

$$\sum_{t \in C \cup J} a_t \leq |C \cup J| - 1$$

A robust cover can be extended by adding TNs which have a higher nominal demand as well as a higher peak demand than the highest values of the TNs already contained in the robust cover (Büsing 2011; Büsing et al 2011). The extended robust cover is denoted by $E(C, J) := (C \cup J) \cup E$ with

$$E := \left\{ t \in T_s \setminus (C \cup J) : \frac{\bar{w}_t}{e_{st}} \geq \max_{t' \in C} \frac{\bar{w}_{t'}}{e_{st'}}, \frac{\bar{w}_t + \hat{w}_t}{e_{st}} \geq \max_{t' \in J} \frac{\bar{w}_{t'} + \hat{w}_{t'}}{e_{st'}} \right\}.$$

The extended robust cover inequality

$$\sum_{t \in E(C,J)} a_t \leq |C \cup J| - 1$$

is also a valid inequality.

The separation problem of robust cover inequalities is NP-hard (Klopfenstein and Nace 2009) while the complexity of the separation of extended robust cover inequalities is unknown. Hence, we use a separation heuristic based on Klopfenstein and Nace (2009). Details on the separation algorithm can be found in Claßen et al (2013).

Set extension. In the first pricing rounds, many dual variables β_t have a value equal to 0 since the RMP contains only very few columns providing poor dual information. Vanderbeck (2005) calls this the *heading-in effect.* A TN t with $\tilde{\beta}_t = 0$ is not considered in the PPs. Hence, the first sets computed in the PPs have a low cardinality. This is why we extend the computed sets of TNs as follows. Assume, a PP for BS s has found a set of TNs τ. For all $t \in T_s \setminus \tau$ with $\beta_t = 0$, we include t in τ if the set $\tau \cup \{t\}$ does not violate the robust capacity constraint (compare (12)). This extension is performed for every computed set of TNs in every B&B node. As a consequence, it is possible that a TN is assigned to more than one BS in an optimal solution. However, we can just drop the redundant assignments.

All these enhancements are implemented by default and we refer to the B&P algorithm as `simple` henceforth.

3.2 The Lagrangian Bound

To evaluate the quality of the current LP solution found by the column generation algorithm, we apply the so-called *Lagrangian bound* (Desaulniers et al 2005). Let $Z^\star_{\mathrm{MP}}$ be an optimal objective value of the LP relaxation of the MP (5), $Z^\star_{\mathrm{RMP}}$ of the current LP relaxation of the RMP and $Z^\star_s$ of the PP (8) corresponding to BS $s \in S$. Obviously, every optimal solution of the current RMP yields an upper bound for the MP. Thus,

$$Z^\star_{\mathrm{MP}} \leq Z^\star_{\mathrm{RMP}}$$

holds. Further, denote by

$$\kappa^\star = \min_{s \in S} \left(\tilde{\alpha}_s - Z^\star_s\right)$$

the minimum reduced cost in the current pricing round regarding all BSs. If $\kappa^\star \geq 0$, there does not exist a variable $\zeta_{s\tau}$ with negative reduced cost and the optimal solution of the current RMP is also an optimal solution of the MP. Furthermore, we know that at most $|S|$ many variables $\zeta_{s\tau}$ are set to one in an optimal solution of the MP. Hence, we have an upper bound on the number of pricing variables to be set to one:

$$|S| \geq \sum_{s \in S} \sum_{\tau \in \mathcal{T}_s} \zeta_{s\tau}.$$

Based on this, we can derive a lower bound on $Z^\star_{\mathrm{MP}}$ for $\kappa^\star < 0$:

$$Z^\star_{\mathrm{RMP}} + |S|\kappa^\star \leq Z^\star_{\mathrm{MP}}, \tag{13}$$

i.e., we cannot reduce the optimal objective value of the current RMP by more than $|S|$ times the minimum reduced cost. This lower bound is called the *Lagrangian bound*.

Now, let ξ denote the cardinality of the maximum independent set in the conflict graph G. At most ξ many BSs can be deployed at the same time. Hence, we can enhance the lower bound (13) by replacing $|S|$ with ξ.

The Lagrangian bound is used to speed up computations at B&B nodes, in particular at the root node. Given a value `gap`, we leave the current node if $-\xi\kappa^\star <$ `gap`. In fact, in case $Z^\star_{\mathrm{RMP}}$ is a multiple of $\gcd(\min_{s \in S} c_s, \lambda)$, and `gap`$\leq \gcd(\min_{s \in S} c_s, \lambda)$, we can be sure that no integer solutions have a value less than $Z^\star_{\mathrm{RMP}}$ and this value is a lower bound. In general, there might exist integer solutions with a value less than the LP value $Z^\star_{\mathrm{RMP}}$.

We denote the B&P algorithm which includes the settings presented in the previous subsection and the Lagrangian bound as a stop criterion by `LB` henceforth.

3.3 Speeding Up the Pricing Problems

The time spent on the solving process of the PPs has significant influence on the performance of the B&P algorithm since it has an impact, e.g., on the number of visited B&B nodes. Therefore, we present several techniques to speed up the PPs in this subsection.

Stabilisation. As explained, e.g., in Vanderbeck (2005); Leitner et al (2011), column generation suffers from several drawbacks such as slow convergence (*tailing-off effect*), generation of irrelevant columns mainly in the first iterations (*heading-in effect*), and *primal degeneracy* entailing multiple dual optimal solutions. These computational instabilities cause long running times with many iterations. Many stabilisation techniques have been developed to diminish these drawbacks (see Lübbecke and Desrosiers (2005) for an overview).

Here, we focus on the primal degeneracy of the RMP and apply stabilisation using alternative dual optimal solutions (Leitner et al 2011). The dual of the RMP, the Restricted Dual Problem (RDP), can be described as the following LP.

$$\max \sum_{t\in T} \beta_t - \sum_{U\subseteq S\text{: max. clique}} \gamma_U \tag{14a}$$

$$\text{s.t.} \sum_{t\in\tau} \beta_t - \alpha_s \le 0 \qquad \forall s\in S, \tau \in \mathcal{T}'_s \tag{14b}$$

$$\beta_t \le \lambda \qquad \forall t \in T \tag{14c}$$

$$\alpha_s - \sum_{U\subseteq S\text{: max. clique},\, s\in U} \gamma_U \le c_s \qquad \forall s \in S \tag{14d}$$

$$\alpha_s,\ \beta_t,\ \gamma_U \ge 0, \tag{14e}$$

where γ_U is the dual variable corresponding to constraint (5c). Further, constraints (14b) correspond to variables $\zeta_{s\tau}$, (14c) to u_t and (14d) to x_s. For a dual optimal solution $(\tilde\alpha, \tilde\beta, \tilde\gamma)$, we define the *dual slack* of variable x_s as follows.

$$\varDelta_s := c_s - \tilde\alpha_s + \sum_{U\subseteq S\text{: max. clique},\, s\in U} \tilde\gamma_U$$

For every dual optimal solution $(\tilde\alpha, \tilde\beta, \tilde\gamma)$, there exists another optimal solution $(\alpha^\star, \beta^\star, \gamma^\star)$ of the RDP computed as

$$\alpha^\star_s = \tilde\alpha_s + \varDelta_s = c_s + \sum_{U\subseteq S\text{: max. clique},\, s\in U} \tilde\gamma_U,$$
$$\beta^\star = \tilde\beta,$$
$$\gamma^\star = \tilde\gamma.$$

The solution $(\alpha^\star, \beta^\star, \gamma^\star)$ is optimal for the RDP since the objective value is not changed, constraints (14d) are fulfilled with equality and constraints (14b) are fulfilled more conspicuously. Obviously, $\alpha^\star_s \ge \tilde\alpha_s$. Hence, if we compare the objective value of the PP to $\alpha_s^\star$ instead of $\tilde{\alpha}_s$ the comparison becomes more restrictive and the generated columns are more likely to be relevant.

Even though the variables β_t are of greater importance since they occur as costs in the objective of the PPs, we cannot increase their value in a given dual optimal solution by adding the dual slack of variable u_t. This would on the one hand increase the objective value of the RDP and on the other hand we could not guarantee the compliance of constraints (14b) anymore.

Note, the stabilisation is only performed at the root node since the dual values of the branching constraints are typically unknown and cannot be computed easily in the subsequent B&B nodes.

We denote the B&P algorithm which includes the settings of algorithm `LB` and the stabilisation approach by `PP_stab` henceforth.

Suboptimal solving of PPs. The proof that a primal solution of a PP is optimal can be rather time consuming for larger test instances. Hence, we stop the solving process of a PP if the integrality gap is less than 1 %. Furthermore, we restrict the number of B&B nodes per PP to 500. If the gap limit is not reached within the first 500 B&B nodes, the gain of solving this PP any further is not sufficient to justify the additional time consumption.

In the case that we have not found any pricing variable at the current B&B node, we solve all PPs again without a gap limit and without the restriction on the number of B&B nodes. By this means, we can guarantee that we have not missed to compute a necessary pricing variable in any node.

The algorithm based on `PP_stab` with the suboptimal solving of the PPs is denoted by `PP_subopt`.

3.4 Limited Number of Added Columns

Per pricing round, we can add up to $|S|$ many variables which can lead to a high number of total variables. Therefore, it seems reasonable to investigate the restriction of the number of added variables per pricing round, e.g., to 1, 5 or 10. Even though we restrict the number of variables added per pricing round, we solve all (necessary) PPs and sort the computed variables by their reduced cost in ascending order. We then add the variable with the most negative reduced cost (or the 5, 10 variables). The setting which fixes the number of added columns per pricing round is denoted by `added_cols` and includes the setting `PP_subopt`.

3.5 A Primal Heuristic

The primal bounds computed during a B&P procedure are in general rather poor. To overcome this drawback we could solve the current RMP to optimality once in a while. However, this can take some time if the solving process has progressed due to the number of currently added columns. This is why we develop a primal heuristic which can be called at the end of each B&B node.

We intend to compute a feasible solution $(\hat{x}, \hat{u}, \hat{\zeta})$ from the current LP solution $(\tilde{x}, \tilde{u}, \tilde{\zeta})$. For this purpose we define a new set of BSs based on the current LP solution $\tilde{x}_s$:

$$S^\star := \{s \in S \,:\, \tilde{x}_s > 0.5\}.$$

For all BSs not in this set, we fix $\hat{x}_s = 0$ and $\hat{\zeta}_{s\tau} = 0 \ \forall s \in S \setminus S^\star, \tau \in \mathcal{T}'_s$. Furthermore, we ignore all already decided assignments, i.e., we set $\hat{\zeta}_{s\tau} = 1 \ \forall (s,\tau)$ if $\tilde{\zeta}_{s\tau} = 1$.

To determine the remaining values we solve the following subMIP (based on the current RMP).

$$\min \sum_{s \in S^\star} c_s x_s + \lambda \sum_{t \in T} u_t \tag{15a}$$

$$\text{s.t.} \sum_{s \in S^\star_t} \sum_{\tau \in \mathcal{T}'_s : t \in \tau} \zeta_{s\tau} + u_t = 1 \qquad \forall t \in T \tag{15b}$$

$$\sum_{s \in U} -x_s \geq -1 \qquad \forall U \subset S^\star, U \text{ max. clique} \tag{15c}$$

$$x_s - \sum_{\tau \in \mathcal{T}'_s} \zeta_{s\tau} \geq 0 \qquad \forall s \in S^\star \tag{15d}$$

$$x_s,\ \zeta_{s\tau},\ u_t \in \{0,1\} \qquad \forall s \in S^\star,\ \tau \in \mathcal{T}'_s,\ t \in T \tag{15e}$$

To speed up the primal heuristic we set a limit of one on the number of solutions, i.e., as soon as an integer solution with a value better than the known primal solution of the RMP is found, the solving process of the subMIP is stopped. Based on this solution, we set the remaining values for $(\hat{x}, \hat{u}, \hat{\zeta})$ to the values of (15) and add $(\hat{x}, \hat{u}, \hat{\zeta})$ as a new primal solution to the RMP. We denote the B&P algorithm which applies this primal heuristic and includes the setting `added_cols` with the best parameter (to be determined) by `heuristic`.

4 Computational Study

In this section, we present a comprehensive computational study to investigate the performance of the techniques illustrated in Section 3. First of all, we describe the considered scenarios and give some information on the general settings for the performed computations. Afterwards, we analyse the gains of the different settings achieved at the root node and for the complete solving process.

4.1 The Scenarios

The planning scenarios considered in this study are based on signal propagation data for Munich available at COST 231 (1996). We consider six different instances of two dimensions as displayed in Table 2. For all instances, the TNs

Table 2 Number of BSs and TNs, and the maximum independent set number of the six test instances.

identifier	# BSs	# TNs	ξ
scen_20_200a	20	200	12
scen_20_200b	20	200	11
scen_20_200c	20	200	12
scen_30_300a	30	300	14
scen_30_300b	30	300	14
scen_30_300c	30	300	15

are randomly distributed, where the BSs are randomly chosen from a larger set of 60 BS candidate sites. We consider simplified scenarios since the computational study focuses on the performance of the different settings for the B&P algorithm. Hence, the BS candidate sites are limited to the location of the BS. However, the integration of configuration options would not change the implementation. For the robustness parameter $\varGamma$, we take values in $\{0, 1, \ldots, 20\}$ for all scenarios into account. We fixed the maximum value to 20 since initial computations performed with the compact model showed a constant solution for $\varGamma \geq 20$ for all scenarios.

Signal prediction, which is needed for the computation of the spectral efficiencies, is done by a cube oriented ray launching algorithm (Mathar et al 2007). Furthermore, two BSs are adjacent in the conflict graph if and only if the distance between them is less than or equal to 500 m. As an example, the resulting graph for scenario scen_30_300b is illustrated in Figure 2.

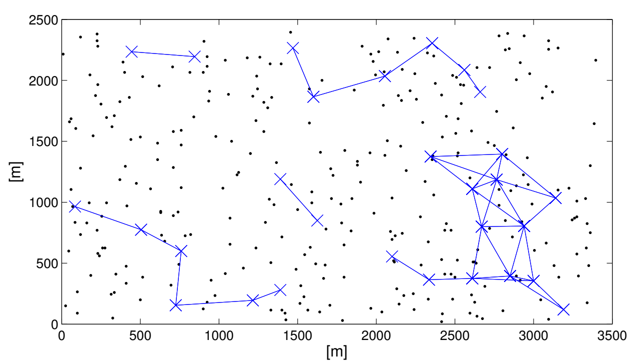

Fig. 2 The conflict graph for scenario scen_30_300b: BSs are denoted by crosses, TNs by dots and the conflict graph is displayed by the connecting lines.

Furthermore, we use the following parameters for all instances: $b_s = 10,000$, $c_s = 4000 \; \forall s \in S$ (based on Deruyck et al (2010)), $e_{\min} = 0.5$, and $\lambda = 1000$. Due to these parameters, we set `abs_gap` $= \gcd(4000, 1000) = 1000$.

We compute the demand values $\bar{w}_t$ and $\hat{w}_t$ for each $t \in T$ as explained in detail in Claßen et al (2013). Mainly, we randomly generate user profiles from Table 3. For the nominal demands $\bar{w}_t$, we consider a regular traffic demand

Table 3 Profiles for TNs

service	regular [%]	high [%]	bit rate [kbps]
data	[10,20]	[30,40]	[512,2000]
web	[20,40]	[40,50]	[128,512]

scenario whereas we consider a high traffic demand scenario for the peak demand values $\bar{w}_t + \hat{w}_t$. A percentage for both data and web services is uniformly drawn from the considered percentage column and multiplied by a bit rate uniformly drawn from the "bit rate" column. Then the remaining percentage is used for VoIP with a bit rate of 64 kbps.

All computations are performed on a Linux machine with 3.40GHz Intel Core i7-2600 processor, a memory limit of 11GB RAM and a general CPU time limit of 2h. We use SCIP 3.0.0 (Achterberg 2009) with CPLEX 12.4 (IBM – ILOG 2012) as underlying LP solver. Furthermore, the PPs are directly solved using CPLEX.

The different algorithms we investigate in the computational study are summarised in Table 4 based on the descriptions given in Section 3.

Table 4 Summary of settings considered in Section 4.

Identifier	Description
`simple`	straightforward B&P, initial solution via compact LP, absolute gap limit, aging of pricing variables, extended cover inequalities for PPs, extended sets of TNs
`LB`	uses the Lagrangian bound as stop criterion at each B&B node
`PP_stab`	as `LB` plus stabilisation at the root node
`PP_subopt`	as `PP_stab` plus solving the PPs suboptimal: gap limit of 1 %, at most 500 B&B nodes
`added_cols`	as `PP_subopt` plus the number of added variables per pricing round is limited
`heuristic`	as `added_cols` plus deploying a primal heuristic

In the following two subsections, we analyse the quality of the LP relaxation and the behaviour of the Lagrangian bound exemplarily for scen_30_300b. We chose this scenario randomly from the three largest instances since the root node of scenarios scen_20_200a–c is mostly solved too fast to reveal the information we would like to present.

4.2 LP Relaxation

In Section 2.1 we have demonstrated via a small example that the LP relaxation of the compact model can be weak and the column generation approach can compute a better LP solution. Exemplarily for scenario scen_30_300b, we present the LP relaxation of the compact model and the LP solution computed at the root node via the column generation approach for $\Gamma \in \{0, 1, \ldots, 50\}$ in Figure 3. Obviously, the LP solution of the column generation is at least

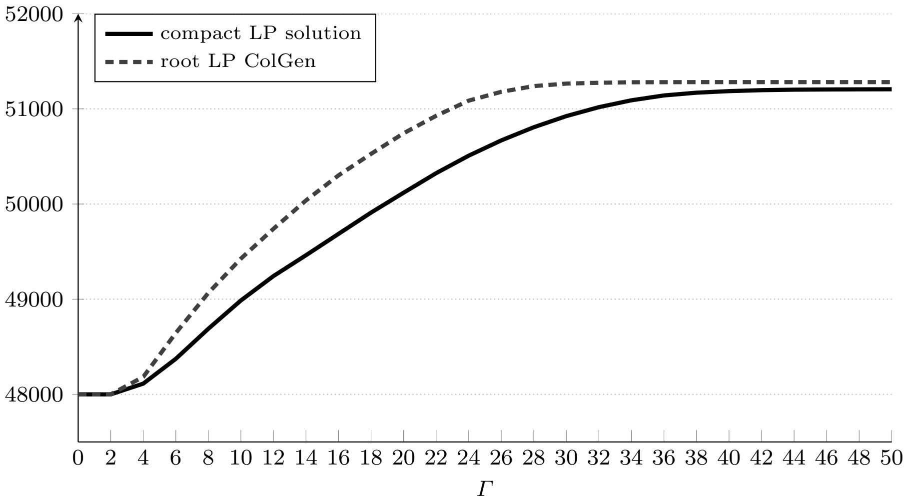

Fig. 3 Comparison of the LP solution of the compact model and the LP solution at the root node of the column generation algorithm for scenario scen_30_300b and $\Gamma \in \{0, 1, \ldots, 50\}$.

as good as the LP relaxation of the compact model. For $6 \leq \Gamma \leq 32$, the LP solution of the column generation approach is significantly better than the compact LP solution. The considerable improvement lies in the actual values of the LP solutions rather than in the percentage value. In half of the instances for $6 \leq \Gamma \leq 32$, the LP solution of the column generation is above the next higher multiple of thousand, e.g., for $\Gamma = 14$, 50038 versus 49462. Due to the fact that the parameters in the objective are multiples of thousand, the LP solution of the column generation gives a significantly better lower bound than the LP relaxation of the compact model. The instances for small and large values of Γ are easier to solve, which is why the two curves (almost) meet. We consider values for Γ of up to 50 here to illustrate the complete behaviour of the LP solutions for this specific scenario even though for all other computations we set $\Gamma \leq 20$.

4.3 Lagrangian Bound

In this subsection, we analyse the behaviour of the Lagrangian bound introduced in Section 3.2. Exemplarily for scen_30_300b and $\Gamma = 4$, Figure 4 presents the Lagrangian bound and the current LP solution per pricing round at the root node. For a better readability, we have omitted the value of the

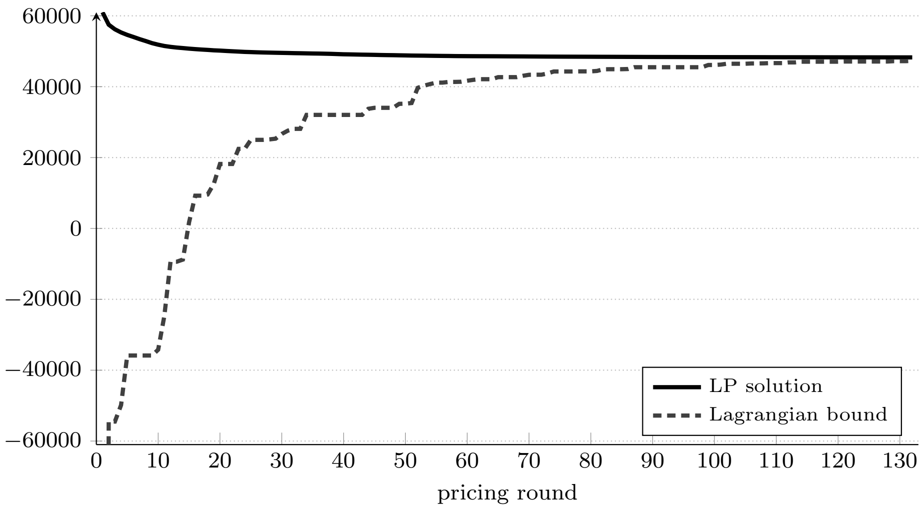

Fig. 4 LP solution and Lagrangian bound per pricing round at the root node for scenario scen_30_300b and $\Gamma = 4$.

Lagrangian bound at the first round (-611000) in the figure. The Lagrangian bound has a stepwise behaviour since we update this value only if the current bound is better than the best known bound. Otherwise, the bound would fluctuate extensively. In the first 60 rounds, the Lagrangian bound is quite far from the LP solution. However, it converges fast against the LP solution, where the LP solution decreases per pricing round. For a more detailed view of this convergence between pricing round 60 and 133 we refer to Figure A.1 in the appendix. As explained before, we stop the solving of the root node if the value of the LP solution and the Lagrangian bound are closer together than the `abs_gap`, i.e., 1000, which is the reason why there is still a gap between the two curves at the last pricing round. Without the application of the Lagrangian bound as a stop criterion (the setting `simple`), the solving of the root node takes 692 rounds whereas with setting `LB`, it takes 133 rounds (see Tables A.2 and A.4 in the appendix for all results).

4.4 Performance of Column Generation at the Root Node

In this subsection, we analyse the performance of the column generation algorithm at the root node. Since we have focused on scen_30_300b in the current and the previous subsection, we present the detailed analysis only for this scenario. The complete results for all scenarios are given in the appendix in Tables A.1 to A.8.

The settings `LB`, `PP_stab`, `PP_subopt` and `added_cols` strongly influence the solution performance at the root node. Hence, in this section we compare the consumed time, the number of pricing rounds and the number of computed variables at the root node for these settings.

First, we take the setting `simple` as a basis and compute the reductions we gain by applying the settings `LB`, `PP_stab` and `PP_subopt`. Therefore, the time reduction is computed as follows.

$$\frac{\texttt{simple}\text{ time } - \text{ advanced time}}{\texttt{simple}\text{ time}},$$

where "advanced time" has to be replaced by the time of the considered setting. Thus, a value of 20 % means that we can reduce the solving time by 20 % due to the application of the setting compared to the time needed by `simple`, while a value of −20 % says that the computation is 20 % slower than for `simple`.

We display the time reductions and the actual time of `simple` exemplarily for scen_30_300b in Figure 5. The absolute times for all scenarios considered in

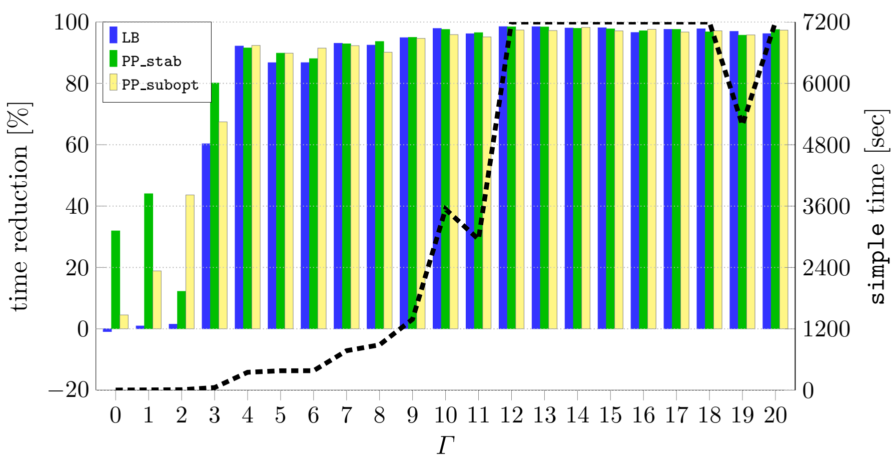

Fig. 5 Time reduction for `LB`, `PP_stab` and `PP_subopt` compared to `simple` (bars) and absolute time consumed by `simple` (dotted line) for scenario 30_300b.

this study can be found in the appendix in Table A.1 for scenarios with 20 BSs and in Table A.3 for the remaining scenarios. For small values of Γ, instance scen_300_30b is rather easy to solve with low total running time. Hence, the time reductions by `LB`, `PP_stab` and `PP_subopt` can be marginal. However, for larger values of Γ all three settings give time reductions of more than 85 % compared to the straightforward approach `simple`. In addition for $\Gamma \geq 12$, the time limit of two hours is reached by `simple` (except for $\Gamma = 19$), implying a time reduction of at least the reported values. However, tests with a time limit of 8 h led to the same results since the values are close to 100 %.

The reductions in the number of pricing rounds and in the number of variables needed at the root node are computed in a similar way as the time reduction before. Clearly, the number of pricing rounds is strongly related to the number of added columns. Hence, we just present the round reduction and the absolute rounds needed by `simple` in Figure 6 for scen_30_300b. The results regarding the number of variables are illustrated in Figure A.2 in the appendix.

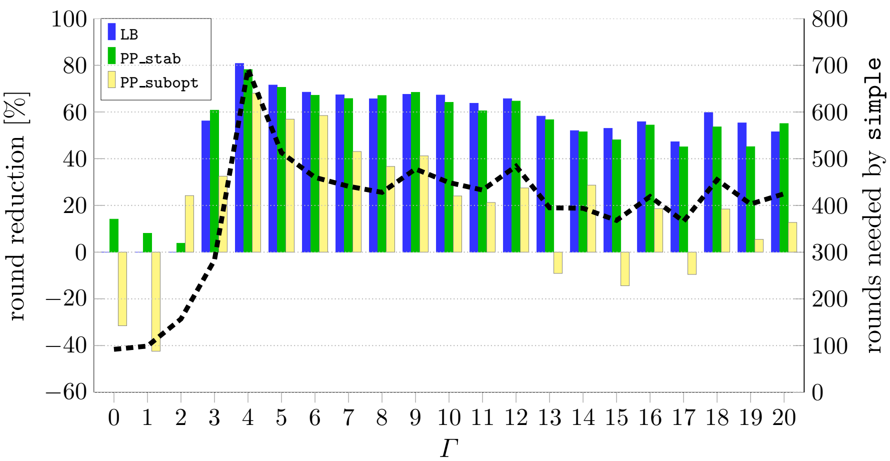

Fig. 6 Round reduction for `LB`, `PP_stab` and `PP_subopt` compared to `simple` (bars) and absolute number of rounds needed by `simple` (dotted line) for scenario 30_300b.

The highest reductions are, in general, achieved by `LB` and `PP_stab` since the utilisation of the Lagrangian bound as a stop criterion obviously reduces the number of necessary pricing rounds and hence, added pricing variables. The stabilisation approach influences the number of rounds and variables only slightly by the possibility of computing different pricing variables. However, the results demonstrate that the applied stabilisation is not strong. On the other hand, even though the setting `PP_subopt` also uses the Lagrangian bound, the number of pricing rounds decreases only slightly compared to `simple` and the number of added variables can even increase. This happens because a pricing variable found by a suboptimal solution of the PP might not be as effective as a variable of an optimal solution. Hence, the good performance of the Lagrangian bound at the root node can be weakened by the suboptimal solving of the PPs.

For a complete overview on the performance of the different settings, we count the cases in which each setting gives the best result (either lowest time, number of rounds, or number of variables) per scenario. Note, in the case that two settings give the same best result, we count it for each setting. The summarised results for scenarios scen_30_300a–c are given in Table 5 and for scen_20_200a–c in Table A.5 in the appendix. The last four lines in each table

Table 5 Number of instances for which each of the four settings `simple`, `LB`, `PP_stab` and `PP_subopt` gives the best results per scenario and in total for scen_30_300a–c.

		`simple`	`LB`	`PP_stab`	`PP_subopt`
scen_30_300a	time	0	2	4	15
	rounds	1	14	4	3
	vars	0	1	17	3
scen_30_300b	time	0	8	8	5
	rounds	0	14	6	1
	vars	0	6	14	1
scen_30_300c	time	0	5	5	11
	rounds	0	18	4	0
	vars	0	1	19	1
total	time	0	15	17	31
	rounds	1	46	14	4
	vars	0	8	50	5
	total	1	69	81	40

give the totals regardless of the different scenarios in each group. We do not sum over the scenarios with different sizes since they are too different in their solving behaviour, especially the solving times vary extremely. Considering the total for each group, i.e., time, rounds and variables, the setting `PP_subopt` gives the best results most frequently independent of the size of the considered scenarios. For the number of rounds, already the `LB` setting performs best and for the number of variables, `PP_stab` most frequently gives the best results. However, in real world applications the time is usually the most restrictive resource. This is the reason why we consider `PP_subopt` as the most appropriate setting since it most frequently gives the best results concerning the time, with a significant distance to the other settings.

The number of computed pricing variables is in general quite high. This leads to the question if it is beneficial to restrict the number of added pricing variables per pricing round. We tested the effect of adding at most 1, 5 or 10 variables per round. The results can be found in the appendix in Tables A.7 and A.8, respectively. Again, we count the times each setting gives the best results regarding running time, number of rounds and number of variables, see Table 6 for scen_30_300a–c and Table A.6 in the appendix for scen_20_200a–c. The lowest number of variables is clearly most frequently computed by `added_cols` with a limit of 1. However, the restriction on the number of variables in general deteriorates the solving process at the root node compared to `PP_subopt` (without these limitations). Hence, we will not consider the `added_cols` setting in the following computations.

Table 6 Number of instances for which each of the four settings `PP_subopt`, `added_cols` 1, `added_cols` 5 and `added_cols` 10 gives the best results per scenario and in total.

		`PP_subopt`	`added_cols` 1	`added_cols` 5	`added_cols` 10
scen_30_300a	time	16	0	0	5
	rounds	16	0	0	5
	vars	2	17	0	2
scen_30_300b	time	12	0	1	8
	rounds	16	0	1	5
	vars	0	18	3	0
scen_30_300c	time	10	0	2	9
	rounds	14	0	0	7
	vars	0	13	3	5
total	time	38	0	3	22
	rounds	46	0	1	17
	vars	2	48	6	7
	total	86	48	10	46

4.5 Performance of B&P

In this section, we present the B&P results for all scenarios given in Table 2 within the time limit of two hours. Note that since the set S is chosen arbitrarily, even scenarios of the same size can behave completely different. The times and optimality gaps for all scenarios can be found in Tables A.9 and A.10, respectively, in the appendix. Scenarios scen_20_200b and scen_20_200c are for most values of Γ solved within a few seconds regardless the chosen setting. Hence, we focus on scenario scen_20_200a and the larger scenarios in this section.

Analogue to the previous subsection, we consider the time reduction which can be achieved by the application of settings `LB`, `PP_stab`, `PP_subopt` and also `heuristic` compared to `simple`. Note, a reasonable evaluation of the time reduction for one scenario is not possible if the algorithms run into the time limit for many values of Γ. For scen_20_200a, most of the instances are solved to optimality within the time limit. Hence, we focus on this scenario for the following evaluations and display the time reduction in Figure 7.

A time reduction of zero indicates that both `simple` and the other setting could not solve the instance within the time limit. As explained before, the instances with a low value of Γ are easier (and faster) and thus, the applied performance improvements can slow down the solving process. However, for larger values of Γ (but $\Gamma \leq 11$) `PP_subopt` and `heuristic` perform best with up to 99 % of time reduction for $\Gamma = 4$. The settings `LB` and `PP_stab` in contrast are subject to high fluctuations see, e.g., $\Gamma = 10$, which partly result from the fluctuations in the solving time. For $13 \leq \Gamma \leq 16$, `simple` computes the optimal solutions quite fast, even though the problems are not as easy as

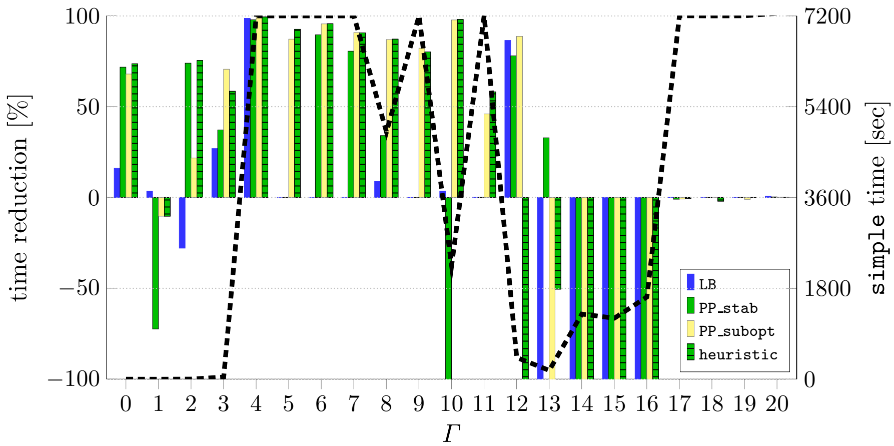

Fig. 7 Time reduction for `LB`, `PP_stab`, `PP_subopt` and `heuristic` compared to `simple` (bars) and absolute time consumed by `simple` (dotted line) for scenario 20_200a.

for $\Gamma \leq 4$. Thus, actually no performance improvements are necessary and the improved settings rather slow down the solving process.

To evaluate the instances which could not be solved within the time limit, we compute the gap reduction as follows. If the optimality gap computed by `simple` is not 0, the gap reduction is

$$\frac{\texttt{simple}\text{ gap } - \text{ advanced gap}}{\texttt{simple}\text{ gap}}.$$

Hence, the maximum possible reduction is 100 %. If `simple` gap equals 0, we set the gap reduction to 0 if the advanced gap is equal to 0 and to −100 % if the advanced gap is strictly greater than 0. Note, if no optimality gap could be computed since no dual bound was found, we assume a gap of 100 % for these evaluations. This is why we also assume the highest occurrent optimality gap is 100 % even though higher gaps are possible in theory.

The gap reduction for scen_20_200a is presented in Figure 8. For the instances with no time reduction in Figure 7, there exists a significant gap reduction, e.g., consider $\Gamma = 9, 18$. This occurs as no optimality gap could be computed quite frequently for `simple` when running into the time limit. For $\Gamma \geq 14$, the optimality gap computed by simple is rather small and thus, `LB`, `PP_stab`, `PP_subopt` and `heuristic` give quite often relatively high increases of the gap even though the computed gaps are in most cases below 5 %. Note, the instances for consecutive values of Γ are completely decoupled. Hence, fluctuations of the optimality gap such as for $7 \leq \Gamma \leq 12$ can occur.

As explained before, the gap reduction cannot reveal any information on the absolute gaps which can in general be rather low. Therefore, we give a different evaluation of the settings for scenarios scen_30_300a–c to accommodate

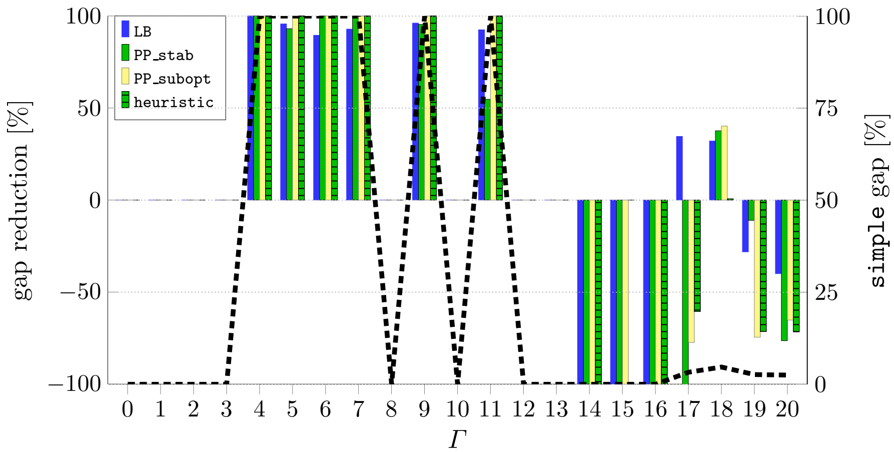

Fig. 8 Gap reduction for `LB`, `PP_stab`, `PP_subopt` and `heuristic` compared to `simple` (bars) and absolute gap computed by `simple` (dotted line) (worst case 100 %) for scenario 20_200a.

this aspect. We focus on the set of larger scenarios since most of these instances cannot be solved within the time limit. Thus, the absolute gaps are of great interest. In total, we have considered 63 different instances for scenarios scen_30_300a–c with $\Gamma \in \{0, 1, 2, \ldots, 20\}$. For an overview on the performance of the different settings `simple`, `LB`, `PP_stab`, `PP_subopt`, and `heuristic` during the complete solving process and a comparison to the compact model, we compute the percentage of these 63 instances which have at most a certain gap after the time limit of two hours is reached. The results are displayed in Figure 9.

All instances solved by the setting `heuristic` have an optimality gap of less than 12 %, which is the best we could achieve for the set of the largest scenarios. However, already by means of `PP_subopt` (which is included in `heuristic`) more than 60 % of the instances are solved with a gap less than or equal to 4 %. Concerning the straightforward approach `simple`, around 43 % of the instances are solved to optimality, whereas for 30 % an optimality gap could not be computed at all (due to missing dual bounds).

Overall, the results suggest a clear outperformance of the `PP_subopt` and `heuristic` settings for computing small gaps within two hours. Nevertheless, the compact model solves 57 % of the largest scenarios to optimality and all instances regarded with a gap less than 9 %.

4.6 Proportion of Time Spent in PPs

Finally, we evaluate the percentage of the total solving time spent in the PPs for each setting in Table 4. In Figure 10, we illustrate these percentage values and the average solving time for scen_30_300b and $\Gamma \in \{0, 2, 4, \ldots, 20\}$. We consider this scenario instead of scen_20_200a here since for scen_20_200a only

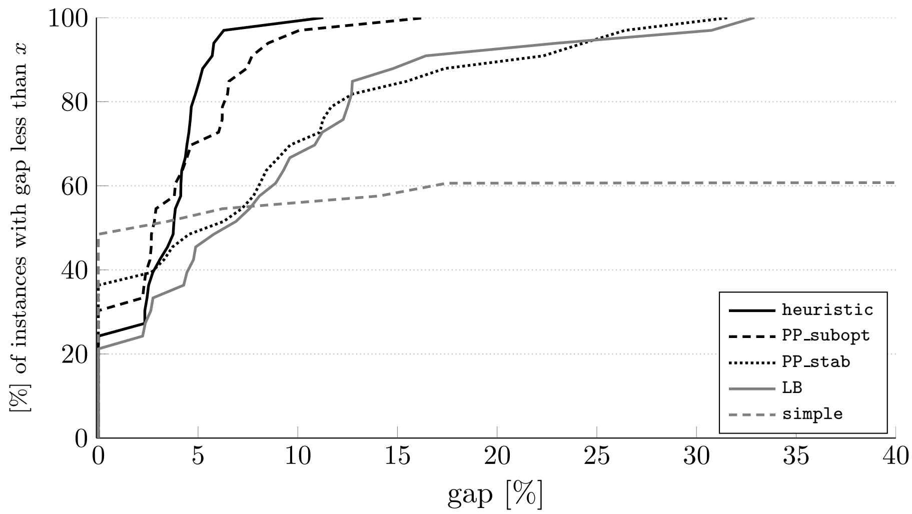

Fig. 9 Percentage of the 63 instances (scen_30_300a–c, $\varGamma \in \{0, 1, 2, \dots, 20\}$) with a gap less than optimality gap given at the x-axis for a time limit of two hours and the different settings.

very few instances spent less than 90 % of the solving time in the PPs (see Figure A.3 in the appendix). In contrast, for scen_30_300b we can in general say that the percentage of time spent in the PPs is higher, the longer the average solving time is. Certainly, `PP_subopt` spends the shortest time in the PPs but still up to 86 % ($\varGamma = 16$). These long times spent to initialise and solve the PPs are a strong indicator that even the improved B&P algorithms presented are not competitive with the compact model regarding the solving time.

5 Conclusion

In this paper, we presented a full B&P approach for the robust wireless network planning problem. Furthermore, we investigated in total seven different settings to improve the solving performance compared to a straightforward implementation. We presented an extensive computational study performed on six test instances of two dimensions and evaluated the settings at the root node and for the complete solving process.

The limitation on the number of added variables per pricing round crystallised to have a negative effect on the solving performance. However, all other enhancements have in general a positive effect, i.e., can reduce the solving time, the number of needed pricing rounds and the number of added pricing variables. The best performance at the root node gives the suboptimal solving of the PPs and for the complete solving process the primal heuristic which also includes the suboptimal solving of the PPs. However, the judgement of the best performance depends on the focus of the evaluation.

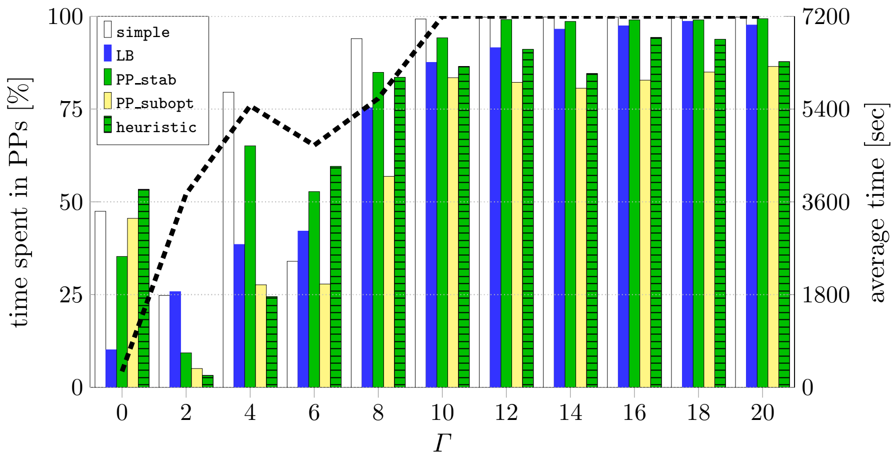

Fig. 10 Percentage of total solving time which is spent in the PPs for all five settings (bars) and the average solving time (dotted line) for scenario scen_30_300b, displayed only for $\Gamma \in \{0, 2, 4, \ldots, 20\}$.

In summary, the results presented in this paper show that it is a complex task to implement a B&P approach for the wireless network planning problem at hand. Since the column generation and the Lagrangian relaxation give the same dual bound in theory, as an alternative, Lagrangian relaxation combined with a B&B framework might be more effective. Furthermore, for more sophisticated robustness models such as the multi-band robustness (Büsing and D'Andreagiovanni 2012), the presented B&P algorithms might give better results compared to a blown-up compact model since the applied robustness approach just affects the PPs instead of the complete problem.

Acknowledgements This work was supported by the UMIC Research Centre at RWTH Aachen University and the DFG research grant KO2311/3-1, SCHM2643/5-1.

References

3rd Generation Partnership Project (2012) URL http://www.3gpp.org/

Achterberg T (2009) Scip: Solving constraint integer programs. Mathematical Programming Computation 1(1):1–41, http://mpc.zib.de/index.php/MPC/article/view/4

Amaldi E, Capone A, Malucelli F (2003) Planning UMTS base station location: optimization models with power control and algorithms. Wireless Communications, IEEE Transactions on 2(5):939–952

Amaldi E, Capone A, Malucelli F (2008) Radio planning and coverage optimization of 3G cellular networks. Wireless Networks 14(4):435–447

Barnhart C, Johnson E, Nemhauser G, Savelsbergh M, Vance P (1998) Branch-and-price: Column generation for solving huge integer programs. Operations Research 46(3):316–329

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. Mathematical Programming 98(1-3):49–71

Bertsimas D, Sim M (2004) The price of robustness. Operations Research 52(1):35–53

Boiardi S, Capone A, Sanso B (2012) Radio planning of energy-efficient cellular networks. In: Computer Communications and Networks (ICCCN), 2012 21st International Conference on, pp 1 –7, DOI 10.1109/ICCCN.2012.6289315

Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. Commun ACM 16:575–577, DOI http://doi.acm.org/10.1145/362342.362367

Büsing C (2011) Recoverable robustness in combinatorial optimization. PhD thesis, Technische Universität Berlin

Büsing C, D'Andreagiovanni F (2012) New Results about Multi-band Uncertainty in Robust Optimization. In: Klasing R (ed) Experimental Analysis — SEA 2012, LNCS, vol 7276, pp 63–74, revised version at http://arxiv.org/abs/1208.6322

Büsing C, Koster AMCA, Kutschka M (2011) Recoverable Robust Knapsacks: the Discrete Scenario Case. Optimization Letters 5(3):379–392

Cisco Systems I (2012) Cisco visual networking index: Global mobile data traffic forecast update, 2011-2016. URL www.cisco.com

Claßen G, Koster AMCA, Schmeink A (2011) Robust planning of green wireless networks. In: Network Games, Control and Optimization (NetGCooP), 2011 5th International Conference on, pp 1–5

Claßen G, Koster AMCA, Schmeink A (2013) A robust optimisation model and cutting planes for the planning of energy-efficient wireless networks. Computers and Operations Research 40(1):80–90

COST 231 (1996) Urban micro cell measurements and building data. URL http://www2.ihe.uni-karlsruhe.de/forschung/cost231/cost231.en.html

Dahlman E, Parkvall S, Skold J, Beming P (2008) 3G Evolution: HSPA and LTE for Mobile Broadband, 2nd edn. Academic Press

Deruyck M, Vereecken W, Tanghe E, Joseph W, Pickavet M, Martens L, Demeester P (2010) Comparison of power consumption of mobile WiMAX, HSPA and LTE access networks. In: 9th Conference on Telecommunications Internet and Media Techno Economics (CTTE), pp 1–7, DOI 10.1109/CTTE.2010.5557715

Desaulniers G, Desrosiers J, Solomon M (eds) (2005) Column Generation. Springer, chap. 1, 12

El-Beaino W, El-Hajj A, Dawy Z (2012) A proactive approach for lte radio network planning with green considerations. In: Telecommunications (ICT), 2012 19th International Conference on, pp 1 –5, DOI 10.1109/ICTEL.2012.6221236

Engels A, Reyer M, Mathar R (2010) Profit-oriented combination of multiple objectives for planning and configuration of 4G multi-hop relay networks. In: 7th Int. Symp. on Wireless Communication Systems (IEEE ISWCS), pp 330–334

Engels A, Neunerdt M, Mathar R, Abdullah HM (2011) Acceptance as a success factor for planning wireless network infrastructure. In: International Symposium on Wireless Communication Systems 2011 (ISWCS'11), Aachen, Germany, pp 889–893

Glaßer C, Reith S, Vollmer H (2005) The complexity of base station positioning in cellular networks. Discrete Applied Mathematics 148(1):1–12

Gordejuela-Sanchez F, Zhang J (2009) Lte access network planning and optimization: A service-oriented and technology-specific perspective. In: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, pp 1 –5, DOI 10.1109/GLOCOM.2009.5425478

IBM – ILOG (2012) CPLEX Optimization Studio 12.4. URL http://www.ilog.com/products/cplex

Klopfenstein O, Nace D (2009) Valid inequalities for a robust knapsack polyhedron – Application to the robust bandwidth packing problem. In: Proceedings of the International Network Optimization Conference (INOC)

Leitner M, Ruthmair M, Raidl GR (2011) On stabilized branch-and-price for constrained tree problems. Tech. rep., Vienna University of Technology, Austria

Lübbecke ME, Desrosiers J (2005) Selected topics in column generation. Operations Research 53(6):1007–1023

Mathar R, Niessen T (2000) Optimum positioning of base stations for cellular radio networks. Wireless Networks 6:421–428

Mathar R, Reyer M, Schmeink M (2007) A cube oriented ray launching algorithm for 3D urban field strength prediction. In: IEEE ICC

Niu Z, Zhou S, Hua Y, Zhang Q, Cao D (2012) Energy-aware network planning for wireless cellular system with inter-cell cooperation. Wireless Communications, IEEE Transactions on 11(4):1412 –1423, DOI 10.1109/TWC.2012.021412.110147

Riihijarvi J, Petrova M, Mahonen P (2005) Frequency allocation for wlans using graph colouring techniques. In: Wireless On-demand Network Systems and Services, 2005. WONS 2005. Second Annual Conference on, pp 216 – 222, DOI 10.1109/WONS.2005.19

Siomina I, Yuan D (2012) Analysis of cell load coupling for lte network planning and optimization. Wireless Communications, IEEE Transactions on 11(6):2287 –2297, DOI 10.1109/TWC.2012.051512.111532

Siomina I, Varbrand P, Yuan D (2006) An effective optimization algorithm for configuring radio base station antennas in UMTS networks. In: IEEE 64th Vehicular Technology Conference (VTC 2006-Fall), pp 1–5, DOI 10.1109/VTCF.2006.252

Vanderbeck F (2005) Column Generation, Springer, chap 12: Implementing mixed integer column generation, pp 331–358

A Appendix

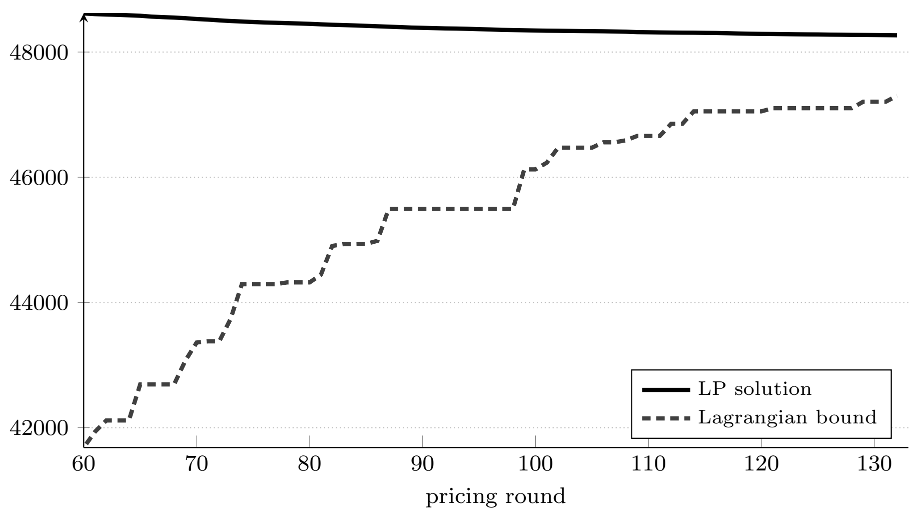

Fig. A.1 LP solution and Lagrangian bound per pricing round at the root node for scenario scen_30_300b and $\Gamma = 4$ zoomed in to rounds 60 to 133.

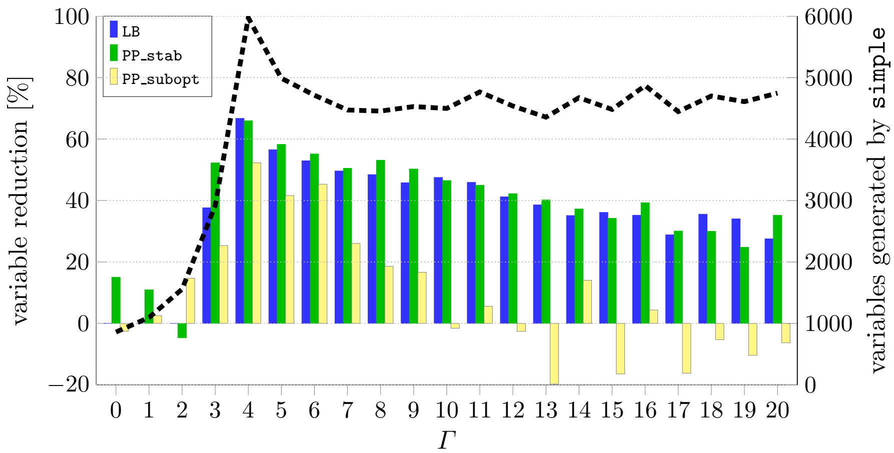

Fig. A.2 Variable reduction for LB, PP_stab and PP_subopt compared to simple (bars) and absolute number of variables generated by simple (dotted line) for scenario 30_300b.

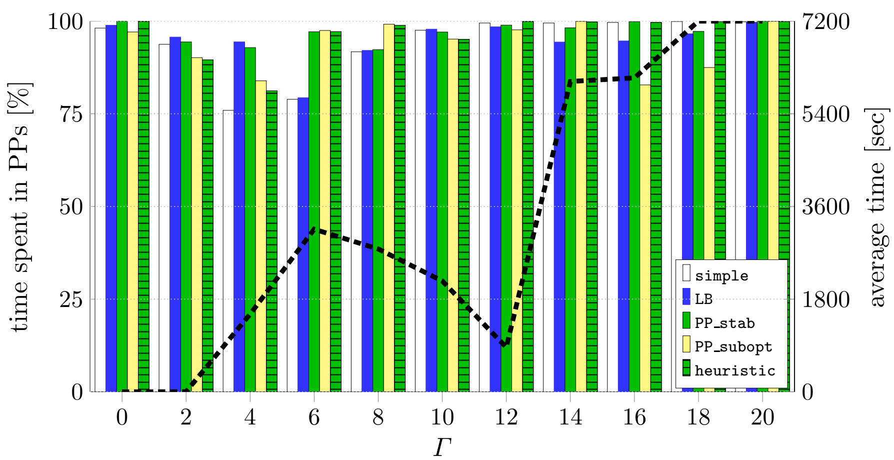

Fig. A.3 Percentage of total solving time which is spent in the PPs for all five settings (bars) and the average solving time (dotted line) for scenario scen_20_200a, displayed only for $\Gamma \in \{0, 2, 4, \ldots, 20\}$.

Table A.1 Solving time (in sec.) at the root node for scen_20_200a–c and the different settings.

	Γ	simple	LB	PP_stab	PP_subopt
scen_20_200a	0	1	0.91	0.24	0.19
	1	0.32	0.32	0.51	0.28
	2	0.58	0.56	0.45	0.46
	3	38.77	13.44	12.73	6.05
	4	7200.16	13.84	15.52	6.02
	5	7200.2	7200.26	7200	127.36
	6	7200.01	7200.14	562.07	179.52
	7	7200.3	7201.07	1474.67	429.37
	8	4963.32	4744.16	2823.13	68.69
	9	7204.61	7202.38	7201.19	419.3
	10	2396.38	1925.7	5583.15	45.02
	11	7204.1	7209.42	7208.48	404.76
	12	440.53	68.86	87.44	39.87
	13	209.24	66.71	141.02	41.48
	14	210.45	117.76	90.36	92.63
	15	390.63	133	128.94	101.39
	16	386.01	92.78	108.8	60.97
	17	319.96	109.67	112.66	72.87
	18	240.78	125.11	128.3	63.16
	19	288.3	86.17	123.87	85.08
	20	410.47	182.86	111.21	61.74
scen_20_200b	0	0.16	0.16	0.13	0.12
	1	0.37	0.35	0.21	0.23
	2	0.91	1.07	0.56	0.53
	3	1.07	1.14	0.61	0.6
	4	0.66	0.73	0.66	0.65
	5	0.8	0.78	0.53	0.64
	6	1.44	1.45	0.77	0.57
	7	1.12	1.15	0.66	1.05
	8	1.16	1.21	1.17	0.62
	9	1.01	1.03	0.87	0.72
	10	3.44	3.53	2.46	1.96
	11	19.39	7.35	15.98	7.53
	12	23.23	10.47	8.7	6.21
	13	53.42	22.06	23.63	14.79
	14	57.02	24.91	29.44	15.49
	15	55.11	25.39	36.94	12.83
	16	51.87	31.5	39.7	17.02
	17	93.27	19.03	30.69	17.32
	18	61.96	22.37	27.54	17.25
	19	172.07	39.43	99.49	11.79
	20	331.46	52.37	113.71	23.52
scen_20_200c	0	0.51	0.37	0.4	0.38
	1	0.49	0.46	0.37	0.38
	2	1.28	1.25	0.53	0.54
	3	0.98	0.92	0.65	0.59
	4	1.13	1.04	0.97	0.93
	5	0.5	0.49	0.35	0.36
	6	1.61	2.02	0.46	0.45
	7	2.23	2.15	0.87	0.9
	8	2.29	2.21	1.05	1.12
	9	3.08	2.75	1.99	1.37
	10	6.78	5.08	3.81	2.74
	11	48.65	7.02	5.16	3.1
	12	42.78	5.77	5.93	3.31
	13	42.27	6.34	8.5	5.54
	14	48.14	9.26	5.86	5.01
	15	68.18	9.26	10.54	5.67
	16	59.19	14.22	13.1	8.86
	17	82.88	18.73	17.96	7.4
	18	73.85	16.3	20.2	10.17
	19	103.41	20.37	26.15	19
	20	83.67	20.06	31.7	11.34

Table A.2 Number of pricing rounds and number of added variables at the root node for scen_20_200a–c and the different settings.

		# pricing rounds				# variables			
	Γ	`simple`	`LB`	`PP_stab`	`PP_subopt`	`simple`	`LB`	`PP_stab`	`PP_subopt`
scen_20_200a	0	60	60	28	28	122	122	89	89
	1	15	15	32	24	75	75	138	107
	2	27	27	31	32	156	156	149	171
	3	446	205	261	251	1672	1123	1239	1200
	4	29160	211	281	211	31223	1147	1354	1094
	5	7766	7813	7448	1191	9593	9641	9346	3198
	6	20559	20189	1991	1431	23813	23431	4100	3657
	7	7834	7824	2101	2387	10239	10229	4312	5085
	8	8493	7705	6971	641	14262	13078	9868	2505
	9	1848	1856	1888	2041	5218	5238	3795	4553
	10	2307	2091	5301	401	6537	6141	8745	2192
	11	1251	1251	1414	1711	4555	4555	3755	3668
	12	444	228	261	281	2327	1680	1693	1657
	13	385	241	331	281	2332	1896	1921	1852
	14	398	293	271	411	2314	2018	1738	2226
	15	454	299	301	371	2534	2005	1849	2101
	16	381	203	251	321	2142	1565	1641	1846
	17	429	253	301	341	2200	1857	1889	2023
	18	346	265	281	301	2088	1864	1751	1781
	19	358	204	281	411	2103	1557	1776	2148
	20	499	310	281	341	2653	2101	1818	1956
scen_20_200b	0	11	11	14	14	41	41	40	40
	1	22	22	19	19	63	63	69	69
	2	43	43	59	59	96	96	116	116
	3	42	42	52	52	159	159	197	197
	4	27	27	67	67	77	77	144	144
	5	29	29	39	58	71	71	100	110
	6	58	58	73	47	152	152	126	105
	7	56	56	47	68	230	230	142	251
	8	66	66	93	41	187	187	160	109
	9	56	56	70	58	171	171	136	122
	10	96	96	73	85	582	582	360	411
	11	194	142	151	170	1103	908	829	838
	12	183	120	111	148	977	849	670	817
	13	262	159	181	191	1374	1151	1024	1064
	14	194	151	171	206	1213	1097	1004	1003
	15	179	120	161	175	962	868	815	856
	16	192	153	161	197	1072	998	868	865
	17	192	122	161	214	1036	904	908	903
	18	194	130	151	203	1089	979	902	956
	19	231	143	221	174	1087	979	940	809
	20	290	192	221	272	1364	1228	943	967
scen_20_200c	0	28	27	31	31	86	86	73	73
	1	25	24	32	32	94	94	80	80
	2	47	46	38	38	134	134	115	115
	3	33	32	43	43	113	113	117	117
	4	50	49	57	57	209	209	214	214
	5	21	20	21	21	149	149	141	141
	6	61	60	22	24	376	376	199	205
	7	67	66	49	49	388	388	276	267
	8	41	40	39	42	452	452	342	368
	9	40	37	41	41	520	494	497	494
	10	47	40	47	56	697	609	577	665
	11	143	56	51	61	1369	830	661	785
	12	121	47	51	61	1351	713	663	785
	13	117	50	61	81	1288	708	785	1040
	14	118	60	51	81	1398	875	664	1034
	15	129	63	71	91	1439	865	911	1092
	16	145	68	71	121	1675	1012	894	1345
	17	155	78	81	111	1630	1123	991	1253
	18	149	74	91	111	1598	1024	1117	1298
	19	161	92	101	161	1658	1209	1180	1600
	20	149	90	101	141	1677	1247	1194	1479

Table A.3 Solving time (in sec.) at the root node for scen_30_300a–c and the different settings.

	Γ	simple	LB	PP_stab	PP_subopt
scen_30_300a	0	5.79	5.87	6.25	4.2
	1	4.94	5.08	4.35	2.47
	2	20.81	19.3	12.12	6.78
	3	53.79	41.67	39.43	23.18
	4	404.89	122.77	92.51	62.93
	5	370.62	98.94	89.76	77.56
	6	812.97	167.54	127.23	99.21
	7	994.85	142.77	132.83	154.27
	8	7201.17	132.89	108.56	171.91
	9	1098.13	150.61	210.7	272.11
	10	1656.2	162.38	160.43	242.74
	11	2219.32	200.67	164.84	259.85
	12	3480.41	185.14	195.71	300.43
	13	3844.43	183.21	361.51	170.88
	14	7205.77	197.29	350.88	153.26
	15	7213.72	226.69	400.07	194.14
	16	7225.15	261.97	516.1	202.14
	17	7239.68	424.97	415.46	288.56
	18	7215.24	467.35	450.52	380.06
	19	7207.25	839.58	953.47	293.37
	20	7318.04	651.85	1233.53	343.69
scen_30_300b	0	3.32	3.35	2.26	3.17
	1	6.36	6.3	3.56	5.16
	2	9.42	9.28	8.27	5.31
	3	51.59	20.47	10.26	16.79
	4	350.57	27.5	29.65	26.79
	5	380.05	50.36	38.58	38.61
	6	380.87	50.43	45.46	32.41
	7	772.48	53.56	54.81	59.57
	8	886.83	66.74	56.37	87.34
	9	1387.7	70.82	69.64	74.29
	10	3540.22	74.84	86.18	145.67
	11	2951.13	112.77	102.55	143.8
	12	7311.42	109.78	115.26	190.53
	13	7279.55	111.16	118.13	203.26
	14	7478	147.39	155.68	131.35
	15	7298.85	136.22	164.33	211.09
	16	7207.55	246.41	205.13	171.23
	17	7379.37	176.84	177.65	241.09
	18	7307.65	161.33	234.84	210.09
	19	5186.8	159.06	222.86	218.42
	20	7291.33	276.16	179.6	193.35
scen_30_300c	0	37.06	36.15	18.2	19.16
	1	445.56	145.56	1634.8	360.83
	2	513.25	246.95	298.01	91.42
	3	399.28	144.58	205.9	169.34
	4	543.35	192.82	216.6	175.54
	5	405.85	159.21	225.04	183.23
	6	973.72	323.71	185.84	264.88
	7	401.81	163.01	171.55	180.55
	8	789.02	227.75	199.51	231.08
	9	729.71	196.38	204.42	207.94
	10	3141.04	243.26	208.2	239.47
	11	768.33	307.41	272.26	418.99
	12	2524.82	480.24	535.88	390.54
	13	1281.99	725.34	1508.57	467.31
	14	1432.04	476.46	609.89	415.74
	15	7240.74	1580.5	537.01	422.71
	16	3616.67	856.08	708.81	382.8
	17	7301.61	742.45	1805.54	353.46
	18	7278.9	2001.25	3181.37	432.88
	19	6855.84	1825.28	2099.54	405.14
	20	7278.73	5473.3	634.72	431.13

Table A.4 Number of pricing rounds and number of added variables at the root node for scen_30_300a–c and the different settings.

		# pricing rounds				# variables			
	Γ	`simple`	`LB`	`PP_stab`	`PP_subopt`	`simple`	`LB`	`PP_stab`	`PP_subopt`
scen_30_300a	0	116	116	131	147	1387	1387	1079	1179
	1	63	63	76	61	907	907	942	746
	2	151	125	160	121	2113	2075	1899	1463
	3	239	239	251	238	3475	3475	2959	2866
	4	642	285	321	371	8734	4873	4172	4672
	5	597	254	271	391	6994	4334	3766	4928
	6	661	256	301	401	6323	4400	4060	5024
	7	799	271	291	511	6931	4881	4014	6031
	8	2700	228	221	491	8370	4079	3214	5542
	9	513	236	321	651	5643	4090	4369	6147
	10	547	228	251	541	5392	4002	3587	5380
	11	548	243	241	541	5414	4110	3455	5606
	12	513	223	251	577	5186	3841	3538	5429
	13	513	225	281	391	5560	4259	3919	4901
	14	491	224	281	351	5262	4029	3946	4757
	15	496	244	301	421	5452	4264	4148	5231
	16	487	249	321	431	5508	4362	4311	5257
	17	518	306	311	551	6083	5191	4426	6221
	18	648	327	321	701	6939	5667	4465	7031
	19	739	375	371	611	7457	6149	4850	6837
	20	770	354	391	670	7182	5822	5142	7110
scen_30_300b	0	92	92	79	121	860	860	731	882
	1	99	99	91	141	1097	1097	977	1069
	2	157	157	151	119	1558	1558	1631	1329
	3	283	124	111	191	2917	1819	1393	2178
	4	692	133	151	221	5974	1988	2033	2852
	5	513	146	151	221	4990	2169	2083	2913
	6	460	145	151	191	4714	2219	2114	2578
	7	441	144	151	251	4473	2254	2214	3309
	8	428	147	141	271	4456	2298	2089	3628
	9	478	155	151	281	4529	2455	2253	3776
	10	449	147	161	341	4499	2362	2407	4570
	11	433	157	171	341	4767	2579	2624	4501
	12	484	166	171	351	4538	2669	2622	4655
	13	395	165	171	431	4357	2676	2606	5216
	14	394	189	191	281	4674	3033	2933	4018
	15	368	173	191	421	4482	2863	2948	5219
	16	419	185	191	341	4870	3155	2957	4658
	17	366	193	201	401	4447	3165	3110	5169
	18	455	183	211	371	4701	3030	3292	4951
	19	403	180	221	381	4609	3040	3467	5088
	20	425	206	191	371	4748	3441	3077	5046
scen_30_300c	0	240	229	233	262	3333	3315	2112	2127
	1	1104	540	2311	1641	10957	7475	12775	10616
	2	834	509	901	781	9705	7296	6666	6200
	3	867	456	661	951	8638	6482	5541	7159
	4	675	375	581	901	8216	6215	5311	7365
	5	638	350	531	801	7608	5751	4892	6568
	6	713	399	411	961	7600	6053	4110	6976
	7	561	338	411	641	7128	5604	4374	5839
	8	559	304	381	691	6991	5362	4321	6085
	9	595	323	381	631	6751	5268	4433	6280
	10	749	351	351	591	6859	5281	4097	5656
	11	520	368	361	801	6584	5675	4224	6277
	12	662	389	441	761	7261	5931	5025	6634
	13	508	415	541	698	7249	6756	6003	6741
	14	552	367	401	603	7215	6248	5244	6301
	15	506	416	351	514	6502	6222	4797	6026
	16	509	360	391	511	6837	6304	5191	5622
	17	517	402	411	506	7303	6553	5453	5911
	18	436	362	431	535	6130	5881	4909	5655
	19	569	424	441	491	6519	6158	5003	5432
	20	431	417	311	521	5399	5337	4284	5338

Table A.5 Number of instances for which each of the four settings `simple`, `LB`, `PP_stab` and `PP_subopt` gives the best results per scenario and in total for scen_20_200a–c.

		`simple`	`LB`	`PP_stab`	`PP_subopt`
scen_20_200a	time	0	0	2	19
	rounds	4	12	4	6
	vars	1	5	8	9
scen_20_200b	time	0	1	3	17
	rounds	6	15	4	3
	vars	5	5	9	8
scen_20_200c	time	0	1	5	15
	rounds	0	15	6	2
	vars	2	6	14	5
total	time	0	2	10	51
	rounds	10	42	14	11
	vars	8	16	31	22
	total	18	60	55	84

Table A.6 Number of instances for which each of the four settings `PP_subopt`, `added_cols` 1, `added_cols` 5 and `added_cols` 10 gives the best results per scenario and in total for scen_20_200a–c.

		`PP_subopt`	`added_cols` 1	`added_cols` 5	`added_cols` 10
scen_20_200a	time	12	4	1	4
	rounds	12	3	2	6
	vars	2	17	1	1
scen_20_200b	time	11	1	5	5
	rounds	10	1	5	6
	vars	3	11	7	0
scen_20_200c	time	11	0	1	9
	rounds	11	0	3	11
	vars	0	17	2	2
total	time	34	5	7	18
	rounds	33	4	10	23
	vars	5	45	10	3
	total	72	54	27	44

Table A.7 Solving time (in sec.), number of pricing rounds and number of added variables at the root node for scen_20_200a–c and the setting `added_cols` with different numbers of added pricing variables (given in the second line).

		time			# pricing rounds			# variables		
	Γ	1	5	10	1	5	10	1	5	10
scen_20_200a	0	0.53	0.51	0.34	64	47	35	63	89	92
	1	0.73	0.55	0.34	51	39	23	50	147	77
	2	6.02	0.74	0.56	361	42	32	360	176	136
	3	165.25	9.51	11.89	4091	271	331	4090	1230	1330
	4	57.54	13.21	9.76	1461	291	261	1460	1353	1255
	5	119.7	65.9	100.02	1511	661	871	1510	2281	2896
	6	118.07	418.88	209.1	1841	3221	1521	1840	5716	4148
	7	136.5	364.37	360.42	1271	2107	1779	1270	4483	4405
	8	127.67	345.76	673.07	1331	2417	4041	1330	4784	6839
	9	195.65	489.78	695.16	1681	2151	2871	1680	4540	5355
	10	258.67	49.18	149.53	1981	431	871	1980	1985	3192
	11	222.95	414.82	723.31	1551	1561	2441	1550	3697	4745
	12	160.43	60.01	40.28	1061	358	221	1060	1581	1447
	13	238.99	43.24	63.17	1561	281	361	1560	1379	1985
	14	309.44	121.95	72.63	1811	511	361	1810	2216	2011
	15	504.25	133.25	85.81	2081	541	391	2080	2439	2221
	16	224.78	70.02	81.66	1321	381	381	1320	1774	2074
	17	327.77	79.41	59.84	1661	431	321	1660	1977	1883
	18	311.73	88.8	84.49	1601	361	381	1600	1629	2104
	19	293.22	88.18	60.6	1601	411	321	1600	1898	1904
	20	371.01	72.63	71.19	1911	411	381	1910	1930	2064
scen_20_200b	0	0.38	0.12	0.13	49	13	13	48	25	32
	1	0.71	0.29	0.39	85	28	34	84	68	110
	2	0.79	0.48	0.76	53	34	61	52	92	124
	3	1.04	0.61	0.9	65	51	73	64	149	244
	4	1.02	0.35	0.45	92	34	42	91	81	94
	5	0.45	0.55	0.69	39	43	57	38	95	105
	6	1.14	0.68	0.6	83	52	48	82	101	100
	7	2.62	1.24	1.19	165	95	79	163	219	249
	8	1.57	0.93	0.88	77	71	69	76	145	129
	9	1.86	0.99	1.02	149	73	80	148	169	148
	10	9.74	2.84	2.75	321	72	117	319	347	495
	11	25.72	7.37	4.46	561	171	131	560	679	646
	12	30.35	6.96	5.54	581	167	141	580	723	772
	13	68.09	12.49	10.34	851	225	203	850	1001	1097
	14	83.2	14.01	15.18	941	207	192	940	890	963
	15	91.33	18.42	23.24	971	225	269	970	854	1028
	16	81.25	17.07	18.41	871	210	215	870	904	932
	17	59.76	17.05	10.22	791	227	158	790	864	855
	18	84.34	16.94	28.96	971	211	268	970	818	977
	19	75.15	14.91	22.66	978	228	264	977	890	987
	20	83.07	24.84	16.15	992	256	225	991	839	977
scen_20_200c	0	0.7	0.24	0.3	70	25	28	68	66	73
	1	1.03	0.76	0.21	69	61	22	67	125	61
	2	1.79	0.43	0.4	101	36	33	99	103	82
	3	1.1	0.66	0.74	90	49	48	88	131	119
	4	2.69	1.21	0.8	129	76	51	127	220	186
	5	2.31	0.54	0.38	99	21	21	97	94	141
	6	3.57	0.85	0.46	145	41	23	143	185	195
	7	3.12	1.2	1.3	138	43	48	136	180	272
	8	6.76	1.52	1.32	198	51	38	196	245	305
	9	13.03	3.03	1.96	301	81	41	300	395	390
	10	18.84	3.55	2.22	371	91	51	370	450	494
	11	18.12	4.17	3.47	421	101	71	420	500	695
	12	21.66	5.24	4.3	461	111	61	460	550	599
	13	23.01	6.9	4.75	481	111	71	480	550	700
	14	37.81	7.69	4.45	611	131	71	610	650	700
	15	36.7	12.76	5.52	671	171	101	670	850	999
	16	53.26	13.27	10.84	791	201	131	790	1000	1271
	17	58.39	17.47	13.92	881	201	151	880	1000	1426
	18	54.15	16.79	11.37	791	221	141	790	1100	1324
	19	57.75	21.05	14.62	871	261	171	870	1300	1535
	20	89.26	16.56	11.13	1091	231	151	1090	1150	1382

Table A.8 Solving time (in sec.), number of pricing rounds and number of added variables at the root node for scen_30_300a–c and the setting `added_cols` with different numbers of added pricing variables (given in the second line).

		time			# pricing rounds			# variables		
	Γ	1	5	10	1	5	10	1	5	10
scen_30_300a	0	29.19	7.24	3.91	899	213	101	897	945	856
	1	80.28	8.02	5.26	1903	211	116	1901	1045	1070
	2	87.82	24.44	8.83	1693	350	162	1691	1740	1396
	3	324.49	66.61	33.22	3749	633	328	3748	3057	2998
	4	527.33	125.54	89.56	3581	811	481	3580	4050	4781
	5	525.35	191.67	102.94	3201	921	471	3200	4600	4698
	6	767.53	218.74	133.39	3461	841	501	3460	4200	4856
	7	857.3	282.93	155.31	3471	1041	561	3470	5200	5412
	8	1159.79	418.22	202.1	3651	1011	581	3650	5050	5396
	9	1392.44	350.16	266.42	3861	1011	681	3860	5050	5858
	10	1219.73	327.58	264.35	3441	881	611	3440	4400	5418
	11	1120.33	241.57	196.58	3241	721	491	3240	3600	4729
	12	1221.21	375.12	178.48	3221	881	461	3220	4400	4525
	13	1350.25	335.18	247.58	3351	801	541	3350	4000	4976
	14	1426.49	312.6	191.49	3581	811	471	3580	4050	4620
	15	1471.76	304.04	229.8	3821	771	511	3820	3850	5004
	16	1734.67	383.19	286.6	4241	921	601	4240	4600	5445
	17	1914.71	358.33	320.47	4391	911	666	4390	4550	6304
	18	2028.74	454.97	304.67	4881	1061	671	4880	5300	6353
	19	2159.34	599.35	315.57	5111	1221	681	5110	6100	6361
	20	2100.99	526.43	349.39	5241	1191	631	5240	5950	5909
scen_30_300b	0	14.93	2.42	4.37	775	103	147	773	487	851
	1	21.51	8.6	7.93	733	228	145	731	830	993
	2	33.14	11.35	9.09	911	270	163	910	1253	1498
	3	65.74	19.15	12.29	1161	261	171	1160	1292	1691
	4	155.55	37.11	28.23	1601	351	221	1600	1750	2200
	5	223.26	60.13	38.56	1841	431	261	1840	2150	2599
	6	264.66	75.23	56.58	1731	451	281	1730	2250	2800
	7	428.73	123.5	60.51	1991	471	281	1990	2350	2800
	8	365.3	172.84	85.14	1841	541	351	1840	2700	3500
	9	480.98	105.45	97.19	2071	501	351	2070	2500	3500
	10	661.54	127.15	115.53	2701	511	401	2700	2550	4000
	11	777.27	218.33	145.12	2641	641	391	2640	3200	3900
	12	892.14	193.51	141.23	2581	601	331	2580	3000	3300
	13	921.32	222.3	108.45	2851	611	301	2850	3050	3000
	14	1086.49	247.04	177.85	2941	671	431	2940	3350	4295
	15	1116.79	295.05	154.36	2991	741	361	2990	3700	3600
	16	1311.19	251.54	201.82	3321	631	471	3320	3150	4700
	17	1396.62	327.39	247.56	3261	751	491	3260	3750	4889
	18	1354.28	288.03	162.42	3401	731	391	3400	3650	3900
	19	1425.4	361.52	277.24	3401	781	551	3400	3900	5449
	20	1463.95	389.24	219.91	3681	901	481	3680	4500	4796
scen_30_300c	0	157.97	16.41	13.48	3234	385	270	3232	1823	2060
	1	385.42	192.98	154.95	5631	1421	1051	5630	7097	7385
	2	520.35	171.9	173.33	5801	1431	1131	5800	7135	8042
	3	403.43	155.14	179.11	4401	1231	1041	4400	6150	7908
	4	635.23	207.3	114.28	4791	1371	771	4790	6839	6063
	5	831.48	221.63	192.91	5271	1171	871	5270	5848	6287
	6	919.1	221.36	225.97	4471	1001	871	4470	5000	6531
	7	993.66	223.76	179.33	4471	921	721	4470	4600	5815
	8	1121.94	268.22	194.79	4221	931	621	4220	4650	5356
	9	1691.43	356.77	314.61	4751	1111	771	4750	5550	6155
	10	1829.49	438.27	243.56	5041	1108	641	5040	5515	5200
	11	1884.03	653.36	328.81	4351	1311	721	4350	6400	5603
	12	2465.74	554.99	315.46	5051	1111	611	5050	5539	5174
	13	3654.6	673.38	558.54	5591	1108	831	5590	5453	6314
	14	3686	804.66	575.29	5251	1183	786	5250	5767	6580
	15	3884.02	954.59	551.64	5491	1226	615	5490	6073	5440
	16	3681.85	859.42	431.29	5161	1166	531	5160	5637	5053
	17	5307.87	885.91	480.5	7081	1231	667	7080	5909	5957
	18	3782.95	743.88	461.48	5351	1027	593	5350	5075	4981
	19	3362.97	773.78	367.05	4901	971	511	4900	4833	4602
	20	3423.02	763.9	338.16	4924	1084	471	4923	4968	4390

Table A.9 Solving time (in sec.) and optimality gap (in %) for scen_20_200a–c and the different settings.

		`simple`		`LB`		`PP_stab`		`PP_subopt`		`heuristic`	
	Γ	time	gap	time	gap	time	gap	time	gap	time	gap
scen_20_200a	0	1.06	0	0.89	0	0.3	0	0.34	0	0.28	0
	1	0.29	0	0.28	0	0.5	0	0.32	0	0.32	0
	2	2.72	0	3.48	0	0.71	0	2.13	0	0.67	0
	3	43.14	0	31.53	0	27.11	0	12.7	0	17.93	0
	4	7200.01	–	102.51	0	157.81	0	40.36	0	45.08	0
	5	7200.75	–	7200.86	4.33	7200	6.92	927.88	0	535.71	0
	6	7200.25	–	7200.22	10.48	753.62	0	322.44	0	313.83	0
	7	7200.15	–	7200.11	7.13	1404.79	0	665.02	0	675.78	0
	8	4900.42	0	4469.07	0	3232.26	0	642.17	0	630.9	0
	9	7200.96	–	7203.94	3.87	7203.11	4.61	1286.13	0	1430.88	0
	10	2250.28	0	2170.6	0	6244.83	0	52.09	0	45.77	0
	11	7205.89	–	7203.31	7.5	7200.08	45.25	3894.12	0	3016.96	0
	12	429.43	0	58.24	0	94.71	0	48.56	0	3688.71	0
	13	175.02	0	2754.55	0	117.57	0	871.46	0	263.46	0
	14	1291.22	0	7200.15	2.66	7200.71	2.21	7222.74	5.22	7229.83	5.22
	15	1209.96	0	7203.8	2.14	7200.28	2.23	7200.12	2.22	7223.91	0
	16	1629.5	0	7222.1	3.44	7204.36	5.35	7200.22	2.6	7230.18	3.76
	17	7203.27	3.27	7200	2.14	7275.58	8.79	7268.13	5.8	7241.81	5.25
	18	7200.14	4.68	7200.32	3.18	7201.18	2.92	7200.15	2.8	7350.03	4.65
	19	7201.9	2.63	7207.12	3.37	7200.34	2.92	7278.56	4.59	7207.45	4.51
	20	7250.86	2.5	7202.24	3.5	7229.13	4.41	7237.87	4.13	7248.42	4.29
scen_20_200b	0	0.18	0	0.2	0	0.19	0	0.21	0	0.14	0
	1	0.4	0	0.37	0	0.25	0	0.32	0	0.24	0
	2	0.76	0	0.74	0	0.63	0	0.6	0	0.52	0
	3	0.85	0	0.98	0	0.68	0	0.68	0	0.6	0
	4	0.67	0	0.66	0	0.77	0	0.89	0	0.68	0
	5	0.67	0	0.71	0	0.62	0	0.98	0	0.65	0
	6	1.48	0	1.38	0	1.83	0	0.6	0	0.58	0
	7	1.6	0	1.56	0	1.5	0	1.46	0	1.66	0
	8	1.65	0	1.6	0	2.66	0	0.87	0	0.61	0
	9	1.45	0	1.37	0	1.99	0	0.9	0	0.74	0
	10	5	0	4.69	0	8.22	0	2.57	0	2.6	0
	11	71.13	0	7200.02	2.68	7200.12	2.04	7200.17	2.24	7209.22	2.55
	12	41.84	0	517.12	0	535.65	0	87.92	0	11.15	0
	13	99.66	0	83.65	0	83.07	0	540.75	0	14.01	0
	14	115.88	0	7200.18	2.27	71.34	0	4691	0	7203.02	1.72
	15	74.41	0	7203.78	1.82	60.88	0	39.47	0	1033.17	0
	16	73.53	0	1457.9	0	219.18	0	5070.2	0	943.83	0
	17	92	0	7200.51	1.82	2314.26	0	536.59	0	1207.59	0
	18	84.12	0	736.64	0	151.04	0	889.48	0	1684.87	0
	19	202.01	0	904.36	0	306.88	0	16.77	0	1394.85	0
	20	322.31	0	509.17	0	1962.71	0	292.44	0	277.47	0
scen_20_200c	0	1.09	0	0.85	0	1.26	0	1.32	0	1.25	0
	1	1.47	0	1.2	0	0.91	0	0.91	0	1.31	0
	2	2.07	0	3.04	0	1.74	0	1.77	0	2.03	0
	3	5.27	0	5.81	0	2.74	0	3.25	0	9.18	0
	4	2.67	0	2.66	0	2.04	0	2.89	0	2.5	0
	5	14.04	0	2.7	0	2.77	0	2.54	0	3.53	0
	6	9.19	0	11.11	0	7.91	0	4.34	0	7.9	0
	7	15.34	0	114.88	0	49.2	0	6.71	0	25.11	0
	8	8.61	0	13.44	0	25.06	0	7.11	0	7.39	0
	9	107.74	0	68.68	0	36.05	0	13.51	0	13.65	0
	10	146.48	0	463.44	0	452.97	0	194.17	0	50.71	0
	11	168.38	0	186.09	0	206.92	0	51.3	0	140.37	0
	12	210.04	0	170.8	0	1300.21	0	225.74	0	279.33	0
	13	308.78	0	255.15	0	569	0	7200.19	2.13	146.21	0
	14	312.72	0	346.15	0	952.33	0	2273.09	0	361.28	0
	15	454.57	0	185.69	0	483.5	0	296.93	0	393.74	0
	16	448.2	0	360.89	0	3552.45	0	1145.11	0	336.94	0
	17	1526.29	0	1464.53	0	7200.44	2.13	7200.06	2.13	161	0
	18	494.12	0	490.02	0	1138.08	0	657.93	0	429.91	0
	19	925.4	0	1627.9	0	1099.78	0	137.81	0	236.92	0
	20	530.59	0	619.15	0	643.97	0	634.29	0	1109.17	0

Table A.10 Solving time (in sec.) and optimality gap (in %) for scen_30_300a–c and the different settings.

		`simple`		`LB`		`PP_stab`		`PP_subopt`		`heuristic`	
	Γ	time	gap	time	gap	time	gap	time	gap	time	gap
scen_30_300a	0	346.37	0	87.87	0	1115	0	4052.8	0	1535.01	0
	1	6.83	0	280.49	0	4.38	0	3.89	0	2.84	0
	2	5158.81	0	1672.43	0	99.41	0	620.62	0	7206.88	2.33
	3	7200	6.98	7200.01	2.33	7200	2.33	159.1	0	2809.22	0
	4	2132.19	0	7200	2.74	7200.12	2.65	7200.27	2.59	7205.43	2.73
	5	7201.98	2.55	7200.46	5.69	7200.23	2.33	7200.07	3.67	7205.79	2.83
	6	3973.69	0	7200.01	5.79	2340.08	0	7200.1	2.22	7212.49	2.33
	7	4576.34	0	7200	4.52	2916.79	0	6211.15	0	3225.44	0
	8	7201.19	–	7200.02	2.22	7200.01	7.78	6828.41	0	5815.42	0
	9	7211.43	19.23	7201.25	2.22	7200.01	2.22	7200.01	2.22	7220.36	2.4
	10	7207.65	14.1	7200.01	4.44	7202.64	11.11	7200.26	4.44	7210.62	4.44
	11	7256.62	13.56	7203.21	9.47	7202.66	9.48	7200.55	3.01	7206.59	2.29
	12	7202.73	17.47	7200.4	11.23	7214.45	9.03	7200.51	3.87	7211.17	3.86
	13	7202.19	14.82	7202.49	12.29	7205.86	10.3	7227.84	6.03	7222.79	6.02
	14	7203.44	–	7204.61	12.74	7232.95	3.72	7200.01	8.52	7205.06	5.23
	15	7244.56	–	7201.39	14.2	7203.13	3.94	7202.18	5.6	7245.93	3.61
	16	7204.6	–	7230.44	8.08	7200.27	8.42	7200.01	6.03	7374.79	3.47
	17	7273.9	–	7200.46	5.98	7200.13	9.44	7221.16	4.26	7212.34	5.99
	18	7212.74	–	7202.35	23.05	7202.36	9.62	7200.57	6.21	7250.74	6.29
	19	7203.03	–	7280.23	5.29	7208.74	7.39	7335.77	6.16	7363.95	4.92
	20	7208.02	–	7200.85	4.77	7207.39	7.19	7200.46	7.71	7255.64	4.66
scen_30_300b	0	131.94	0	1325.9	0	46.48	0	38.88	0	7.29	0
	1	4.66	0	6.11	0	274.14	0	7200.01	8.33	7203.23	2.08
	2	4066.69	0	839.2	0	2066.93	0	6855.59	0	4921.68	0
	3	4868.11	0	7200.24	3.04	4873.64	0	7200	6.02	7203.21	3.04
	4	5044.92	0	7200.01	2.35	656.52	0	7200.31	2.29	7214.02	2.44
	5	5418.99	0	7200.01	4.7	7200	3.97	7200	3.24	7211.19	4.14
	6	6827.73	0	1967.21	0	5241.67	0	2239.2	0	7211.57	2.52
	7	4484.44	0	6066.87	0	7200.61	3.09	6548.03	0	5499.29	0
	8	7200.21	6.23	7200.4	4.28	7201.06	11.26	2445.32	0	3979.5	0
	9	7202.13	5.78	7200	3.42	7201.59	10.73	7200.74	2.36	7208.21	4.47
	10	7220.27	506.96	7200.44	12.5	7203.13	6.28	7202.6	16.45	7205.96	4.87
	11	7210.7	18.99	7204.37	13.86	7202.44	24.37	7200.01	5.12	7210.22	4.17
	12	7321.57	–	7232.35	12.28	7241.66	24.34	7211.91	6.48	7204.4	5.06
	13	7211.16	–	7201.81	8.49	7222.24	17.76	7224.44	3.17	7216.29	2.19
	14	7489.56	–	7437.84	14.78	7218.72	17.36	7200.75	6.55	7207.85	4.54
	15	7204.45	–	7210.36	10.23	7253.33	13.24	7216.01	4.72	7234.29	4.72
	16	7225.89	–	7266.25	8.89	7212.3	12.74	7242.61	3.81	7208.92	4.18
	17	7246.48	–	7229.52	18.92	7258.59	24.56	7204.66	5.4	7298.52	3.98
	18	7387.59	–	7204.19	30.76	7207.11	26.42	7215.88	4.65	7263.26	5.71
	19	7225.82	492.3	7212.85	7.31	7237.93	11.95	7200.01	7.8	7205.6	2.73
	20	7322.56	491.21	7201.6	9.28	7208.83	22.32	7208.94	7.43	7280.75	4.61
scen_30_300c	0	6770.51	0	7200.01	12.7	558.21	0	7200	10.08	7210.42	11.28
	1	3091.06	0	3114.23	0	4920.99	0	7200	2.44	857.21	0
	2	1616.77	0	7200.01	10.85	2130.94	0	6143.24	0	1781.93	0
	3	2683.38	0	7200	4.76	1977.99	0	5179.89	0	1637.3	0
	4	1020.54	0	5427.79	0	4698.29	0	7200	2.38	7017.68	0
	5	7200.46	2.59	7201.36	4.2	7202.29	9.06	7200	9.24	7204.22	4.99
	6	2661.13	0	7200.7	2.63	4965.02	0	7200.48	2.65	7213.04	3.08
	7	4139	0	7200.09	3.37	7200.19	5.97	7200.12	11.56	7211.26	3.06
	8	1675.23	0	7137.57	0	7200	4.55	1339.93	0	7210.93	4.36
	9	7200.01	2.71	7200.89	4.66	7220.35	2.46	5118.94	0	5950.55	0
	10	5398.76	0	7204.92	4.88	2440.07	0	7202.37	2.81	7223.46	3.79
	11	1450.75	0	4227.63	0	7248.39	6.67	1971.96	0	4114.18	0
	12	5312.07	0	7201.12	7.58	7222.57	3.29	7206.24	4.21	7229.78	4.14
	13	7212.48	2.59	1996.6	0	7222.93	3.55	7200.63	2.51	2411.72	0
	14	6233.14	0	7200.02	16.42	7201.15	31.64	7200.7	2.66	7200.89	3.75
	15	7222.6	–	7203.77	2.23	7201.91	10.71	7200.28	4.6	7204.16	2.46
	16	7200.93	3.47	7291.94	9.6	7205.9	11.68	7208.51	2.89	7237.53	4.13
	17	7316.37	–	7200.67	12.7	7277.25	8.76	7133.3	0	5487.71	0
	18	7203.63	–	7223.27	6.9	7206.5	15.46	4063.04	0	1088.21	0
	19	7200.09	6.91	7258.07	8.84	7234.61	21.18	7200.84	3.58	7205.1	2.87
	20	7233.51	–	7201.3	32.91	7223.33	8.11	7214.85	6.22	7220.78	5.79