CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

A Branch-and-Benders-Cut Method for Nonlinear Power Design in Green Wireless Local Area Networks

Bernard Gendron
Maria Grazia Scutellà
Rosario G. Garroppo
Gianfranco Nencioni
Luca Tavanti

September 2014

CIRRELT-2014-42

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Branch-and-Benders-Cut Method for Nonlinear Power Design in Green Wireless Local Area Networks

Bernard Gendron[1*], Maria Grazia Scutellà[2], Rosario G. Garroppo[3], Gianfranco Nencioni[3], Luca Tavanti[3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[2] Dipartimento di Informatica, Università di Pisa, Largo B. Pontecorvo, 3, I-56127 Pisa, Italy

[3] Dipartimento di Ingegneria dell'Informazion, Università di Pisa, Via Caruso, 16, 56122 Pisa, Italy

Abstract. We consider a problem arising in the design of green (or energy-saving) Wireless Local Area Networks (WLANs). In this context, decisions on powering-on a set of access points, via the assignment of one power level to each opened access point, and decisions on the assignment of the user terminals to the opened access points, have to be taken simultaneously. In particular, the power level assigned to an access point affects, in a nonlinear way, the capacity of the connections between the access point and the user terminals that are assigned to it. We model this problem as an integer program with nonlinear constraints. We solve the proposed nonlinear integer programming model by means of a branch-and-Benders-cut method. The approach has been tested on a large set of instances, and compared to a more traditional Benders decomposition algorithm on a subset of the instances. The computational results show the superiority of the proposed branch-and-Benders-cut approach in terms of solution quality, scalability and robustness.

Keywords. Integer programming, Benders decomposition, branch-and-cut, green wireless local area network, network design.

Acknowledgements. The work of Bernard Gendron was funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grant 184122-2010. The work of Maria Grazia Scutellà was funded by project PRIN 2012, "Mixed-Integer Nonlinear Optimization: Approaches and Applications" (2012JXB3YF). The work of Rosario Garroppo, Gianfranco Nencioni and Luca Tavanti was partially supported by the Italian Ministry of University and Research 17 under the "GreenNet (Greening the Networks)" FIRB project. This support is gratefully acknowledged.

---

* Corresponding author: Bernard.Gendron@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1 Introduction

We address an optimization problem arising in the design of green (or energy-saving) Wireless Local Area Networks (WLANs). We focus on the design of efficient reconfiguration algorithms to reduce the power consumption of the WLAN infrastructure when the load is scarce. Most of the currently deployed enterprise WLANs are continuously operated at full power, i.e., all access points are always turned on with the transmission power set to the maximum. This produces a considerable waste of energy, because the same power is employed at the peak hours and during the off peak periods. We address this issue by proposing an optimization model that is used to take two kinds of decisions: (i) associate each user with one of the available access points and (ii) set the transmission power level of each access point.

More formally, the problem is defined on a bipartite network structure, with a set of access points (APs) that must be assigned to user terminals (UTs) in order to satisfy the user demands, without exceeding the capacity of the connections between the APs and the UTs, with the aim of minimizing the overall power consumption of the APs. Each UT must be assigned to exactly one powered-on AP. Several different power levels (PLs) are available for powering on each AP. If an AP is powered-on, then exactly one PL must be associated with it.

A key issue arises concerning the capacity of the connections between the APs and the UTs: the specific PL assigned to a (powered-on) AP affects, in a nonlinear way, the capacity of the connections between the AP and the UTs assigned to it. The only assumption is that the transmission capacity between a UT and an AP is a nonnegative nondecreasing function of the radiated power at the AP, which will be formally defined in Section 2. As a result, the optimization model is an integer nonlinear program, a class of notoriously difficult mathematical programs.

We propose to address this intrinsic difficulty by developing an exact algorithm based on Benders decomposition [4]. Since the Benders subproblem in our approach is a 0-1 program and not a linear program (LP), we use *canonical cuts* [2], as in logic-based Benders decomposition [13] and combinatorial Benders decomposition [7], instead of the classical LP duality-based Benders cuts. The resulting Benders cuts are improved by simple arguments based on the assumption that the transmission capacity functions are nondecreasing.

In a non-standard fashion, our master problem includes the variables of the Benders subproblem, but relaxes their integrality. Linear approximations of the nonlinear transmission capacity functions are also included in the formulation of the master problem. As a result, the master problem is a mixed-integer linear programming (MILP) relaxation, which we solve with a state-of-the-art MILP software tool. Instead of solving one MILP master problem at every iteration of a classical Benders decomposition approach, we use a *branch-and-Benders-cut (BBC) method*, also called Benders-based branch-and-cut method, where a single branch-and-cut (B&C) tree is constructed and the Benders cuts are added during the exploration of the B&C tree.

This algorithmic scheme has attracted the attention of many researchers recently, as it makes better use of the reoptimization capabilities of the MILP solvers than the classical Benders decomposition approach. This is discussed, for instance, in [16], which uses an interior-point method to solve a Benders reformulation in a BBC framework, applying it to facility location and network design problems. Other recent implementations of the BBC method include: [10], which compares BBC to classical Benders decomposition for a multi-layer network design problem, showing significant speedups on average; [9], which combines the generation of outer approximation and Benders cuts in a BBC method for the single allocation hub location problem under congestion; [5], where a BBC method is used to solve a hop-constrained survivable network design problem; [1], which uses BBC algorithms for solving production routing problems under demand uncertainty. In all these references, the Benders subproblem is an LP and the Benders cuts are based on LP duality, as in the approach originally proposed by Benders [4]. As mentioned above, our Benders subproblem is a 0-1 program and we make use instead of canonical cuts [2]. Another major difference between these references and our paper is that our master problem includes the variables of the Benders subproblem, but relaxes their integrality. In the above references, a traditional partitioning of the variables into master problem variables and subproblem variables is used, as in [4].

This paper is a follow-up on an earlier contribution by the same authors [12], where it was assumed that the power consumed by an AP does not depend on the demands assigned to that AP. In the present paper, we extend the model proposed in [12] to include a linear dependency between the power consumed by an AP and the total demands assigned to that AP, which yields a more realistic formulation that now includes assignment costs between UTs and APs, called *UT assignment costs*. In [12], a classical Benders decomposition method has been proposed, which corresponds to a cutting-plane approach where feasibility cuts are iteratively added to the master problem, thanks to the information provided when solving the Benders subproblem. The latter is a feasibility problem, because of the absence of UT assignment costs. This is in contrast with the Benders subproblem defined in the present paper, which is an optimization problem, given the inclusion of UT assignment costs. This is a major difference, as the presence of such additional assignment costs prevents a straightforward extension of the classical Benders decomposition approach used in [12], as we clarify in Section 3.5. Another notable difference is that the master problem in [12] does not include the variables of the Benders subproblem, as it is the case in the present paper. The decomposition adopted in [12] thus follows a traditional variable partitioning approach as in the original Benders method [4], where the variables of the master problem and those of the subproblem do not overlap. In Section 4, we compare the performance of the two methods on instances without UT assignment costs, on which the two approaches can be compared. The computational results show the superiority of the proposed BBC approach in terms of solution quality, scalability and robustness.

The paper is organized as follows. In Section 2, we describe our problem, which we denote as GWLANP, and we present the integer nonlinear programming model we propose for the GWLANP. The BBC method is described in Section 3. Computational results from experiments on randomly generated realistic instances are reported in Section 4. The conclusion summarizes our findings and identifies promising research directions.

2 Problem Description and Formulation

In order to state the GWLANP in a formal way, we need to characterize the energy consumed by the powered-on APs and the capacity of the connections between the APs and the UTs. First, let us denote by $\mathcal{I}$, $\mathcal{J}$ and $\mathcal{K}$ the sets of UTs, APs and PLs, respectively.

Concerning the energy consumed by the powered-on APs, the power consumed by $j \in \mathcal{J}$ is composed of a fixed component and of two variable components. The fixed component, denoted p_0, is bound to the mere fact that the device is powered-on, and therefore, it encompasses AC/DC conversion, basic circuitry powering, dispersion, etc. The first variable power component associated with $j \in \mathcal{J}$ is given by its radiated power π_j, which depends on the PL assigned to $j \in \mathcal{J}$. More precisely, if $k \in \mathcal{K}$ is assigned to $j \in \mathcal{J}$, then we have $\pi_j = p_k$, where p_k denotes the power provided by $k \in \mathcal{K}$. Regarding the second variable power component, it linearly depends on the total demands assigned to $j \in \mathcal{J}$, denoted T_j. Therefore, the energy consumed by a powered-on AP $j \in \mathcal{J}$ is given by $p_0 + \pi_j + \mu_j T_j$, where μ_j is a proportionality coefficient.

Concerning the capacity of the connections between the APs and the UTs, a key issue is that the specific PL assigned to a powered-on AP affects, in a nonlinear way, the capacity of the connections between the AP and the UTs assigned to it. The assumption is made that the transmission capacity between $i \in \mathcal{I}$ and $j \in \mathcal{J}$, denoted $r_{ij}(\pi_j)$, is a nonnegative nondecreasing function of the radiated power π_j. In practice, the transmission capacity function satisfies the following conditions:

- There exists a threshold $\gamma_{ij} > 0$ such that $r_{ij}(\pi_j) = 0$ if $\pi_j \leq \gamma_{ij}$ and $r_{ij}(\pi_j) > 0$ whenever $\pi_j > \gamma_{ij}$. Thus, $j \in \mathcal{J}$ can only be assigned to $i \in \mathcal{I}$ when its radiated power π_j remains above γ_{ij}.

- $r_{ij}(\pi_j) \leq r_{max}$ for any π_j, where r_{max} is the maximum rate achievable by any physical connection.

In all instances used in our computational experiments (see Section 4), we use the following piecewise linear transmission capacity function:

$$r_{ij}(\pi_j) = \begin{cases} 0, & \text{if } \pi_j \leq \gamma_{ij}, \\ \min\{\alpha_{ij}\pi_j, r_{max}\}, & \text{otherwise}, \end{cases} \tag{1}$$

where α_{ij} denotes a transmission loss factor between $j \in \mathcal{J}$ and $i \in \mathcal{I}$. It is important to note, however, that our BBC method does not depend on this particular function and can be generalized to any nonnegative nondecreasing transmission capacity function. The BBC method only requires an upper linear approximation $r^u_{ij}(\pi_j)$ to $r_{ij}(\pi_j)$. In the case of the function used in our instances, we simply use $r^u_{ij}(\pi_j) = \alpha_{ij}\pi_j$.

In the GWLANP, the decisions to be taken are what APs to power-on, how to assign a PL to each powered-on AP and how to assign exactly one powered-on AP to each UT. Such decisions must be taken in such a way as to satisfy the demand w_i for each $i \in \mathcal{I}$, by respecting the nonlinear transmission capacities between APs and UTs. As indicated above, the objective is to minimize the overall power consumption of the powered-on APs. The problem can be seen as a discrete location problem, where the capacity to assign to each location (which is the power level in this context) also has to be decided. In other words, the GWLANP is a particular case of a broader class of location-design problems, where both location and capacity dimensioning decisions must be taken.

To model the GWLANP, we define the following sets of binary variables:

- $x_{ij} = 1$, if AP $j \in \mathcal{J}$ is assigned to UT $i \in \mathcal{I}$; 0, otherwise; (*UT assignment variables*)
- $y_{jk} = 1$, if PL $k \in \mathcal{K}$ is assigned to AP $j \in \mathcal{J}$; 0, otherwise. (*PL assignment variables*)

Given the definitions of these variables, we derive the following relationships for the radiated power of $j \in \mathcal{J}$ and for the total demands assigned to $j \in \mathcal{J}$, respectively: $\pi_j = \sum_{k\in\mathcal{K}} p_k y_{jk}$ and $T_j = \sum_{i\in\mathcal{I}} w_i x_{ij}$. The model can then be written as follows:

$$z(GWLANP) = \min \sum_{j\in\mathcal{J}} \left\{ \sum_{k\in\mathcal{K}} (p_0 + p_k) y_{jk} + \sum_{i\in\mathcal{I}} \mu_j w_i x_{ij} \right\} \quad (2)$$

$$\sum_{j\in\mathcal{J}} x_{ij} = 1, \quad i \in \mathcal{I}, \quad (3)$$

$$\sum_{k\in\mathcal{K}} y_{jk} \le 1, \quad j \in \mathcal{J}, \quad (4)$$

$$x_{ij} \le \sum_{k\in\mathcal{K}} y_{jk}, \quad i \in \mathcal{I}, j \in \mathcal{J}, \quad (5)$$

$$\sum_{i\in\mathcal{I}|r_{ij}(\pi_j)>0} \frac{w_i x_{ij}}{r_{ij}(\pi_j)} \le 1, \quad j \in \mathcal{J}, \quad (6)$$

$$x_{ij} \in \{0,1\}, \quad i \in \mathcal{I}, j \in \mathcal{J}, \quad (7)$$

$$y_{jk} \in \{0,1\}, \quad j \in \mathcal{J}, k \in \mathcal{K}. \quad (8)$$

The objective (2) is to minimize the total power consumption, which depends on the powering-on decisions, on the power levels assigned to the powered-on APs, and on the total demands assigned to the powered-on APs. Equations (3) are the single assignment constraints that impose that exactly one AP must be assigned to each UT. Inequalities (4) impose that at most one PL can be selected for each AP. Inequalities (5) ensure that an AP cannot be assigned to any UT if the AP is powered-off. Inequalities (6) are the capacity constraints for each AP. Relations (7) and (8) define the integrality of the variables. Note that, given that at most one PL can be chosen for each AP, it is not necessary to associate further binary variables with the APs in order to state the powering-on decisions, since such decisions are captured by the terms $\sum_{k\in\mathcal{K}} y_{jk}$. This is why the fixed power cost p_0 is part of the cost associated with the y_{jk} variables in the objective function.

Note that the problem considered in [12] can be seen as a special case of the GWLANP where, for each $j \in \mathcal{J}$, $\mu_j = \mu \ge 0$, a proportionality coefficient that is constant over all APs. In that case, the UT assignment costs can be removed from the objective function, since

$$\sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}} \mu w_i x_{ij} = \mu \sum_{i\in\mathcal{I}} w_i \left(\sum_{j\in\mathcal{J}} x_{ij} \right) = \mu \sum_{i\in\mathcal{I}} w_i,$$

i.e., the UT assignment costs are the same, irrespective of the solution. We call this special case the *GWLANP without UT assignment costs*.

3 The Branch-and-Benders-Cut Method

In this section, we present the BBC method for solving the GWLANP. Sections 3.1 and 3.2 describe the master problem and the Benders subproblem, respectively. The different types of Benders cuts added during the course of the algorithm are introduced in Section 3.3. Section 3.4 gives a formal statement of the BBC algorithm, as well as a proof of convergence. Finally, Section 3.5 is dedicated to an extensive comparison between the BBC method and the Benders decomposition algorithm proposed in [12] for the GWLANP without UT assignment costs.

3.1 Master Problem

In order to solve the nonlinear model (2)-(8), we propose a BBC method. As in classical Benders decomposition, the approach consists in solving a master problem, which is a relaxation of model (2)-(8), to which we gradually add Benders cuts. In a non-standard way, our master problem involves both the PL assignment variables y_{jk} and the UT assignment variables x_{ij}. The master problem is denoted $M(x, y)$ and is initially defined as:

$$z(M(x,y)) = \min \sum_{j\in\mathcal{J}} \left\{ \sum_{k\in\mathcal{K}} (p_0 + p_k) y_{jk} + \sum_{i\in\mathcal{I}} \mu_j w_i x_{ij} \right\} \tag{9}$$

subject to (3), (4), (5), (8), and

$$\sum_{i\in\mathcal{I}|r^u_{ij}(\pi_j)>0} \frac{w_i x_{ij}}{r^u_{ij}(\pi_j)} \leq 1, \quad j \in \mathcal{J}, \tag{10}$$

$$x_{ij} \in [0,1], \quad i \in \mathcal{I}, j \in \mathcal{J}. \tag{11}$$

Constraints (10) define a relaxation of the nonlinear capacity constraints (6) obtained by replacing functions $r_{ij}(\pi_j)$ by upper linear approximations $r^u_{ij}(\pi_j)$. Constraints (11) define the UT assignment variables as continuous between 0 and 1. Together, these two sets of constraints, along with constraints (3), (4), (5), (8), define a MILP relaxation of model (2)-(8). During the course of the BBC algorithm, Benders cuts are gradually added to the master problem, as we see below.

3.2 Benders Subproblem

The master problem is solved by a B&C method implemented in a state-of-the-art MILP solver (we use CPLEX, version 12.5.1). Each time an integer solution $\bar{y}$ is obtained during the exploration of the B&C tree, we solve the following Benders subproblem, denoted $S(x,\bar{y})$:

$$z(S(x,\bar{y})) = \min \sum_{j\in\bar{\mathcal{J}}} \left\{ \sum_{k\in\mathcal{K}} (p_0+p_k)\bar{y}_{jk} + \sum_{i\in\mathcal{I}} \mu_j w_i x_{ij} \right\} \tag{12}$$

$$x_{ij} \in \{0,1\}, \qquad i\in\mathcal{I}, j\in\bar{\mathcal{J}}, \tag{13}$$

$$\sum_{j\in\bar{\mathcal{J}}|\bar{r}_{ij}>0} x_{ij} = 1, \qquad i\in\mathcal{I}, \tag{14}$$

$$\sum_{i\in\mathcal{I}|\bar{r}_{ij}>0} \frac{x_{ij}w_i}{\bar{r}_{ij}} \le 1, \qquad j\in\bar{\mathcal{J}} \tag{15}$$

$$x_{ij} = 0, \quad (i,j)\in\mathcal{I}\times\bar{\mathcal{J}} \mid \bar{r}_{ij}=0, \tag{16}$$

where $\bar{\mathcal{J}} \subseteq \mathcal{J}$ is the set of APs that are powered-on according to $\bar{y}$ (i.e., $\sum_{k\in\mathcal{K}} \bar{y}_{jk} = 1$), while $\bar{r}_{ij} = r_{ij}(\sum_{k\in\mathcal{K}} p_k\bar{y}_{jk})$ is the capacity of the connection between $i\in\mathcal{I}$ and $j\in\bar{\mathcal{J}}$ induced by the power level assignment given by $\bar{y}$. Note that the integrality of the x_{ij} variables is now reimposed in the Benders subproblem and that the capacity constraints (15) are now linear. Also observe that inequalities (15) have, in general, the structure of knapsack constraints, which implies that $S(x,\bar{y})$ cannot be solved as an LP. In fact, this Benders subproblem has the structure of a generalized assignment problem, which can be solved by specialized algorithms (see, for instance, [17] and the references therein). In our implementation, we use the same state-of-the-art MILP software tool as when solving the master problem. Since $S(x,\bar{y})$ is not an LP, we cannot use LP duality-based Benders cuts, and we rely instead, as explained below, on the canonical cuts for the unit hypercube, studied in [2], which are also used in logic-based Benders decomposition [13] and combinatorial Benders decomposition [7].

3.3 Benders Cuts

If $S(x,\bar{y})$ is feasible, and $\bar{x}(\bar{y})$ is the computed optimal solution, then a feasible solution $(\bar{x}(\bar{y}),\bar{y})$ to the original nonlinear formulation (2)-(8) has been determined. If the corresponding objective function value $z(S(x,\bar{y}))$ is better than the value of the current best feasible solution, denoted z_u^*, then both z_u^* and the best feasible solution are suitably updated. Note that z_u^* is not the B&C incumbent value managed by the MILP solver, since the latter corresponds to a feasible solution to $M(x,y)$, which is a relaxation of model (2)-(8), to which Benders cuts are added. In fact, to ensure the convergence of the BBC algorithm, it is necessary, as we see below in Theorem 1, that the value z_u^* is substituted to the incumbent value that would normally be stored by the MILP software tool when solving $M(x,y)$ by B&C.

Furthermore, instead of fathoming the B&C node corresponding to the integer solution $\bar{y}$, the following canonical cut is added to $M(x,y)$ and the B&C algorithm is restarted at that node:

$$\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}|\bar{y}_{jk}=0} y_{jk} + \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}|\bar{y}_{jk}=1} (1-y_{jk}) \ge 1. \tag{17}$$

The rationale behind cut (17) is that, since the best solution $\bar{x}(\bar{y})$ for the given configuration $\bar{y}$ has been determined, we can cut all solutions of the form $(x, \bar{y})$. Note that cut (17) is not valid for the original formulation, but since it removes only the feasible solutions of the form $(x, \bar{y})$, for which we have already computed the best solution $(\bar{x}(\bar{y}), \bar{y})$, then no optimal solution can be missed. Since the cut is not valid in general, $M(x, y)$ is no more a relaxation of the original model, but rather a relaxation of the model representing the original set of feasible solutions with the exclusion of the solutions of the form $(x, \bar{y})$. To the best of our knowledge, such a simple cut generation strategy has never been used in a BBC or Benders decomposition approach. We further discuss this issue in Section 3.5, where we compare our BBC algorithm to the Benders decomposition method presented in [12] for the GWLANP without UT assignment costs.

When $S(x, \bar{y})$ is infeasible, we could generate cut (17), which is now obviously valid for the original model, given that it removes only the solution $\bar{y}$, which cannot yield a feasible solution. However, we can strengthen this cut by using simple arguments based on the assumption that the transmission capacity functions $r_{ij}(\pi_j)$ are nondecreasing. We first define the *strengthened feasibility cut* as follows:

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K} | p_k > \bar{\pi}_j} y_{jk} \geq 1, \tag{18}$$

where $\bar{\pi}_j = \sum_{k \in \mathcal{K}} p_k \bar{y}_{jk}$.

Lemma 1 *If $S(x, \bar{y})$ is infeasible, then (18) is a valid inequality to (2)-(8).*

Proof: An infeasible $S(x, \bar{y})$ implies that the PLs assigned to the powered-on APs, according to $\bar{y}$, do not provide enough capacity to satisfy the demands of the UTs. Therefore, it is necessary to increase at least one of the values $\bar{r}_{ij}$, i.e., we must raise the PL of at least one AP (which follows from the fact that the transmission capacity functions $r_{ij}(\pi_j)$ are nondecreasing functions of π_j, $j \in \mathcal{J}$).

We can further strengthen the Benders cut in case $S(x, \bar{y})$ is infeasible by first solving an auxiliary Benders subproblem $S(x, \tilde{y})$, where $\tilde{y}$ is defined as follows:

$$\tilde{y}_{jk} = \begin{cases} 1, & \text{if } k = k_{max} \text{ and } \sum_{k \in \mathcal{K}} \bar{y}_{jk} = 1, \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

where k_{max} is the index of the PL providing the maximum radiated power.

Indeed, if both $S(x, \bar{y})$ and $S(x, \tilde{y})$ are infeasible, we define the *maximally strengthened feasibility cut* as:

$$\sum_{j \in \mathcal{J} | \bar{\pi}_j = 0} \sum_{k \in \mathcal{K}} y_{jk} \geq 1. \tag{20}$$

Lemma 2 *If $S(x, \bar{y})$ and $S(x, \tilde{y})$ are infeasible, then (20) is a valid inequality to (2)-(8).*

Proof: According to $\tilde{y}$, the maximum possible radiated power is associated with the group of APs that are powered-on in solution $\bar{y}$. An infeasible $S(x, \tilde{y})$ thus implies that no feasible solution exists that uses only such a subset of APs. Therefore, at least one AP that is powered-off in $\bar{y}$ (i.e., an AP j such that $\bar{\pi}_j = 0$) must be powered-on.

Note that cut (20) coincides with cut (18), if we consider $\tilde{y}$ in place of $\bar{y}$. It is, in general, a stronger cut, since $\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}|p_k>\bar{\pi}_j} y_{jk} = \sum_{j\in\mathcal{J}|\bar{\pi}_j=0}\sum_{k\in\mathcal{K}} y_{jk} + \sum_{j\in\mathcal{J}|\bar{\pi}_j>0}\sum_{k\in\mathcal{K}|p_k>\bar{\pi}_j} y_{jk} \geq \sum_{j\in\mathcal{J}|\bar{\pi}_j=0}\sum_{k\in\mathcal{K}} y_{jk} \geq 1$.

Whether $S(x,\bar{y})$ is feasible or not, the B&C node corresponding to $\bar{y}$ should not be fathomed, as its descendants might contain another feasible solution to the GWLANP with a better objective function value. That is why, in both cases, the cut is added and the B&C algorithm is restarted at the current node. This point is made more precise in the following section.

3.4 Convergence of the Algorithm

The BBC algorithm can be summarized as follows:

1. $z_u^* = \infty$ (z_u^* is the best known upper bound on $z(GWLANP)$).
2. Using z_u^* as incumbent value, perform B&C for solving $M(x,y)$ until an integer solution $\bar{y}$ is found or the B&C search is completed (the currently best known lower bound on $z(M(x,y))$ is z_l^*).
3. If the B&C search is completed, then STOP the BBC algorithm.
4. Solve $S(x,\bar{y})$.
5. If $S(x,\bar{y})$ is feasible, then:
   (a) If $z(S(x,\bar{y})) < z_u^*$, then $z_u^* = z(S(x,\bar{y}))$ and store the optimal solution $(\bar{x}(\bar{y}),\bar{y})$ to $S(x,\bar{y})$.
   (b) If $z_u^* \leq z_l^*$, then STOP the BBC algorithm.
   (c) Add the canonical cut (17).
6. If $S(x,\bar{y})$ is infeasible:
   (a) Solve $S(x,\widetilde{y})$.
   (b) If $S(x,\widetilde{y})$ is infeasible, then add the maximally strengthened feasibility cut (20).
   (c) If $S(x,\widetilde{y})$ is feasible, then add the strengthened feasibility cut (18).
7. Go to Step 2 (restarting B&C at the current node).

Theorem 1 *The BBC algorithm identifies an optimal solution to model (2)-(8), if there is one.*

Proof: Assume that model (2)-(8) is feasible. Note that at least one feasible solution to (2)-(8) is identified by the BBC algorithm. Indeed, the initial master problem $M(x,y)$ is a relaxation of model (2)-(8) and, consequently, its set of feasible solutions includes all feasible solutions to (2)-(8). When the B&C algorithm for solving $M(x,y)$ identifies an integer solution $\bar{y}$, either a feasibility cut is added or a feasible solution to (2)-(8) is identified. This last alternative will necessarily arise after adding a finite number of feasibility cuts, since model (2)-(8) is feasible and the y solutions to $M(x,y)$ coincide with the ones to (2)-(8).

Lemmas 1 and 2 show that the feasibility cuts (18) and (20) are valid for (2)-(8), and therefore their addition to $M(x, y)$ cannot eliminate any feasible solution to model (2)-(8). Concerning the canonical cuts (17), observe that they are added to $M(x, y)$ when the optimal feasible solution $(\bar{x}(\bar{y}), \bar{y})$ corresponding to $\bar{y}$ has been determined. Since the objective function value $z(S(x, \bar{y}))$ of $(\bar{x}(\bar{y}), \bar{y})$ is used to improve the best known upper bound z_u^* on $z(GWLANP)$, no optimal solution to model (2)-(8) can be discarded by the addition of (17).

To conclude, observe that the number of feasible $\bar{y}$ configurations is finite. Therefore, after a finite number of cut additions, the BBC algorithm must end, due to either one of the following reasons:
1) The B&C search in Step 2 is completed. In this case, we have identified a feasible solution to (2)-(8) of objective function value z_u^*. Now, assume that this solution is not optimal. This implies that there is an optimal solution to (2)-(8), say (x^*, y^*) of objective function value $z(x^*, y^*) < z_u^*$, for which the corresponding configuration y^* has not been generated when solving $M(x, y)$ by B&C. This, in turn, implies that there exists some node p that has been fathomed, but would have yield configuration y^* after a finite number of branchings. Node p has been fathomed by the lower bound test, i.e., $z^l(p) \geq z_u^*$, where $z^l(p)$ is the lower bound associated with node p (recall that z_u^* is the incumbent value used in B&C). Furthermore, the fact that node p would have yield configuration y^* after a finite number of branchings implies that $z(x^*, y^*) \geq z^l(p)$, node p being a relaxation of subproblem $S(x, y^*)$ for which an optimal solution is x^*. Collecting together these facts, we obtain: $z_u^* > z(x^*, y^*) \geq z^l(p) \geq z_u^*$, a contradiction. Hence, the best feasible solution identified at the end of the BBC algorithm is necessarily optimal. Note that this part of the proof relies on the fact that z_u^* is substituted to the incumbent value that would be normally used when performing the B&C method in Step 2. Failure to perform this substitution would result into an algorithm that is not necessarily exact.
2) The condition $z_u^* \leq z_l^*$ in Step 5b is verified. This case implies that any feasible solution (x, y) to the GWLANP that could still be generated by performing the B&C method for solving $M(x, y)$ has an objective function value $z(x, y) \geq z_l^* \geq z_u^*$, and therefore cannot improve upon z_u^*.

Hence, the BBC algorithm ends with an optimal solution to model (2)-(8).

3.5 Comparison with Benders Decomposition for a Special Case

In this section, we describe the Benders decomposition (BD) algorithm [12] developed for the GWLANP without UT assignment costs. Our objective is to state the similarities and the differences between this BD algorithm and the BBC method described above.

One of the main differences between the two methods lies in the way the two algorithms perform B&C on the master problem: while the BBC method explores a single B&C tree, adding the Benders cuts during the exploration of that tree, the BD approach performs B&C at every iteration, adding the Benders cuts only after the B&C has completed its exploration. Hence, the BD algorithm explores several B&C trees, with the master problem being gradually augmented with Benders cuts.

Another main difference between the two approaches is the way they define master problems. In the BD algorithm, the master problem includes only the AP assignment variables y_{jk}. Since the UT assignment constraints (3) are then relaxed, the following valid inequalities are introduced in the master problem:

$$\sum_{j \in \mathcal{J}} r_{ij}^u(\pi_j) \geq w_i, \quad i \in \mathcal{I}. \tag{21}$$

The master problem in the BD method, denoted $M_{BD}(y)$, can therefore be formulated as follows:

$$z(M_{BD}(y)) = \min \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} (p_0 + p_k) y_{jk} \tag{22}$$

subject to (4), (8), (21) and the Benders cuts added so far during the course of the algorithm. In order to generate a good set of initial Benders cuts, the BD method first solves the relaxation corresponding to $M(x, y)$, the BBC master problem, which is further strengthened by imposing the integrality of the UT assignment variables x_{ij}. This type of initialization strategy, involving the solution of a relaxation of the problem to generate a good set of initial Benders cuts, is well-known in the Benders decomposition literature (see, for instance, [15] for an early contribution on this topic). All subsequent iterations solve the classical Benders master problem $M_{BD}(y)$.

At every iteration of the BD algorithm, the master problem is solved until an optimal solution $\bar{y}^0$ is obtained. All other integer solutions, say $\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^n$, found during the exploration of the B&C tree, are also collected. For each solution $\bar{y} = \bar{y}^q, q = 0, 1, \ldots, n$, the Benders subproblem $S(x, \bar{y})$ is solved, as in the BBC algorithm. Note, however, that $S(x, \bar{y})$ is no more an optimization problem, but is rather a feasibility problem, since there are no UT assignment costs.

If $S(x, \bar{y})$ is infeasible, a Benders feasibility cut (18) or (20) is generated (the BD algorithm also solves subproblem $S(x, \tilde{y})$, with $\tilde{y}$ defined as in (19)). If $S(x, \bar{y})$ is feasible, and $\bar{x}(\bar{y})$ is the computed feasible solution, then a feasible solution $(\bar{x}(\bar{y}), \bar{y})$ to the GWLANP is obtained. If the corresponding objective function value $z(S(x, \bar{y}))$ is better than the value of the current best feasible solution, denoted z^*_u, then both z^*_u and the best feasible solution are updated. Whenever $z^*_u \leq z^*_l$, where z^*_l is the optimal value of the master problem, we can conclude that an optimal solution to the GWLANP has been identified. This is the case when $S(x, \bar{y})$ is feasible and $\bar{y} = \bar{y}^0$, the optimal solution to the master problem: the optimality of $\bar{y}$ for the master problem and the feasibility of the Benders subproblem suffice to conclude to the optimality of any feasible solution to $S(x, \bar{y})$, because of the absence of UT assignment costs.

Such a conclusion cannot be derived for the general case of the GWLANP with UT assignment costs. This is why we rely on the addition of the canonical cuts (17) in the BBC algorithm, which then take the place of the usual Benders optimality cuts, i.e., they cut the solutions of the form $(x, \bar{y})$ when the Benders subproblem $S(x, \bar{y})$ is feasible. For the GWLANP without UT assignment costs, these cuts are not needed. In fact, they are simply replaced by updating the B&C incumbent value with z^*_u. This, in effect, cuts all the feasible solutions $(\bar{x}(\bar{y}), \bar{y})$ such that $z(S(x, \bar{y})) \geq z^*_u$.

It is worth noting that, at any iteration of the BD algorithm, the master problem always defines a relaxation of the GWLANP. Hence, the stopping condition $z^*_u \leq z^*_l$ can equivalently be replaced by $z^*_u = z^*_l$, since z^*_l is then necessarily a lower bound on $z(GWLANP)$. In contrast, the master problem in the BBC algorithm is also a relaxation, but not of the GWLANP, rather of a restriction of the problem obtained by adding the canonical cuts corresponding to the feasible Benders subproblems. Hence, for the BBC method, it might happen that $z^*_u < z^*_l$ at the conclusion of the algorithm.

To further highlight the similarities and the differences between the two algorithms, we conclude this section with an outline of the BD algorithm (comparative computational results are presented in Section 4.3):

1. $z^*_u = \infty$ (z^*_u is the best known upper bound on $z(GWLANP)$).

2. Using z^*_u as incumbent value, perform B&C for solving $M_{BD}(y)$ (or, initially, $M(x, y)$ with the integrality imposed on the x_{ij} variables) until an optimal solution $\bar{y}^0$ of value

z_l^* is found; let $\bar{y}^1, \bar{y}^2, \ldots, \bar{y}^n$ be the other integer solutions obtained during B&C.

3. For each solution $\bar{y} = \bar{y}^q, q = 0, \ldots, n$, do:
   (a) Solve $S(x, \bar{y})$.
   (b) If $S(x, \bar{y})$ is feasible, then:
      i. If $z(S(x, \bar{y})) < z_u^*$, then $z_u^* = z(S(x, \bar{y}))$ and store the feasible solution $(\bar{x}(\bar{y}), \bar{y})$ to $S(x, \bar{y})$.
      ii. If $z_u^* \leq z_l^*$, then STOP the BD algorithm.
   (c) If $S(x, \bar{y})$ is infeasible:
      i. Solve $S(x, \widetilde{y})$.
      ii. If $S(x, \widetilde{y})$ is infeasible, then add the maximally strengthened feasibility cut (20).
      iii. If $S(x, \widetilde{y})$ is feasible, then add the strengthened feasibility cut (18).
4. Go to Step 2 (restarting B&C from scratch).

4 Computational Results

Our computational experiments aim to assess the effectiveness and the efficiency of the BBC method, and to stress its robustness and scalability issues. Note that the GWLANP has never been addressed in the general form studied in this paper. Therefore, no comparison with approaches from the literature can be performed. However, since the BD algorithm proposed in [12] addresses the special case without UT assignment costs, we compare the two approaches on instances of this type. The BBC and BD algorithms have been implemented in C++ using IBM ILOG CPLEX Optimization Studio 12.5.1. The experiments have been performed on a PC with 4 Intel Core i7 CPUs @ 3.07GHz (hyperthreading enabled), 8GB RAM, and an ASUS P6T DELUXE V2 Motherboard.

In Section 4.1, we describe the procedure used to generate realistic GWLANP instances. Section 4.2 summarizes our results on a large set of 360 large-scale instances. Section 4.3 is dedicated to a computational comparison of the BBC and BD algorithms, using the same set of 360 instances, but modified by setting $\mu_j = 0, j \in \mathcal{J}$. Additional results are also presented on a set of 90 larger instances, which have been generated to study the scalability of the methods.

4.1 Generation of Instances

To generate realistic GWLANP instances, we use related features extracted from real-life measurement campaigns in corporate environments [3, 14]. We first specify the values of $|I|$, $|J|$ and $|K|$. Then, the positions of the APs and of the UTs in each instance are randomly determined, as follows. First, we divide the test field into a regular grid of $|\mathcal{J}|$ squares. Then, the APs are placed one per square, with their coordinates chosen randomly within the square. The set of UTs is also split into $|\mathcal{J}|$ subsets, and the elements of each subset are randomly spread over each square. This strategy ensures enough uniformity in the placement of the UTs and the APs, so as to mimic a corporate scenario and to avoid heavily unbalanced instances.

Other relevant instance characteristics are the transmission loss factors α_{ij}, which have been computed by using a simplified version of the COST-231 multi-wall path loss model for indoor, non-LOS environments [8], and the maximum achievable rate r_{max}, set to $54Mbps$ according to the 802.11g standard. In addition, the addressed traffic demands w_i have an average value of $300kbps$, and they have been randomly generated within a variation of $\pm 10\%$.

To complete the parameter list, we set the sensitivity thresholds γ_{ij} and the power component figures p_0 and p_k according to [6]. Finally, the proportionality coefficients μ_j are selected based on the indications in [11].

4.2 Assessing the Performance of the BBC Algorithm

We first assess the performance of the BBC algorithm on 12 sets of instances obtained by combining $|I| = \{100, 150, 200\}$ with $|J| = \{10, 15\}$ and $|K| = \{3, 4\}$. Specifically, we generated and solved 30 instances in each of the 12 sets. The total number of instances is thus equal to 360. For these experiments, we set a CPU time limit equal to 3600 seconds.

For each instance, we report two performance measures:

- The final gap, in percentage, between the bounds, measured as

$$GAP = \max\{0, 100 \times (z_u^* - z_l^*)/z_l^*\}.$$

  Note that, because it might happen that $z_u^* < z_l^*$ at the end of the BBC algorithm, it is necessary to modify the usual formula for computing the gap. Even if z_l^* is not necessarily a lower bound on $z(GWLANP)$, this modified gap measure is a fair approximation of the distance between the current best known upper bound z_u^* and the optimal value $z(GWLANP)$.

- The CPU time, in seconds, denoted $TIME$.

In Table 1, for each of the 12 sets of instances, we report the average gap, in percentage, returned by the BBC method on the 30 instances in the set (AVG), and also the average CPU time, in seconds, required by the approach. For both measures, the standard deviation, STD, is also reported.

These results show that most instances with $|\mathcal{I}| = 100$ and 150 are solved to optimality within the time limit of 3600 seconds. They also show that high-quality results are obtained for all instances, including those with $|\mathcal{I}| = 200$. In fact, by considering the entire set of 360 instances, the BBC algorithm computed a feasible solution for each instance in the set. Furthermore, it was able to certify the optimality of the computed solution for all instances, except 37, for which the algorithm reached the time limit.

4.3 Comparing with the BD Algorithm for the Special Case

Now, we consider the GWLANP without UT assignment costs. We perform experiments on the same set of instances as in Section 4.2, but by setting $\mu_j = 0, j \in \mathcal{J}$ in each instance.

$\lvert\mathcal{I}\rvert$, $\lvert\mathcal{J}\rvert$, $\lvert\mathcal{K}\rvert$	GAP (%)		$TIME$ (s)	
	AVG	STD	AVG	STD
100, 10, 3	0	0	22	3
100, 15, 3	0	0	246	58
100, 10, 4	0	0	43	6
100, 15, 4	0	0	368	43
150, 10, 3	0.02	0.02	220	117
150, 15, 3	0	0	499	52
150, 10, 4	0.04	0.04	582	152
150, 15, 4	0	0	920	113
200, 10, 3	1.06	0.48	1374	244
200, 15, 3	0.61	0.27	1709	208
200, 10, 4	0.50	0.32	1325	175
200, 15, 4	1.06	0.30	2660	201

Table 1: BBC results on 360 instances (30 in each set) with UT assignment costs; Gap and Time (limit: 3600s): average (AVG) + standard deviation (STD)

We run the BBC algorithm and the BD method, described in Section 3.5, both with a CPU time limit of 3600 seconds.

In Tables 2 and 3, for each of the 12 sets of 30 instances, we report the average gap, in percentage, and the average computational time, in seconds, returned by the BBC approach and by the BD algorithm, respectively. For both measures, the standard deviation is also reported. For both methods, each performance measure is defined as in Section 4.2, but note that z_l^* is a lower bound on $z(GWLANP)$ for the BD method. Hence, the reported gaps are exact and not approximations, as it is the case for the BBC algorithm.

Furthermore, to better compare the quality of the solutions returned by the two approaches, we report in Table 4 the percentage gap defined as the difference between the best upper bounds determined by the two algorithms. More precisely, if we define $z_u^*(BD)$ and $z_u^*(BBC)$ as the best upper bounds on $z(GWLANP)$ found by the BD method and by the BBC algorithm, respectively, then we compute for each instance the quantity: $DIFF = 100 \times (z_u^*(BD) - z_u^*(BBC))/\min\{z_u^*(BD), z_u^*(BBC)\}$.

Table 2 shows a picture similar to that of Table 1, except that the average CPU times are always lower for the instances without UT assignment costs, which is not surprising, given that the Benders subproblems for these instances are feasibility problems that are easier to solve than the optimization problems for the general case. In particular, Table 2 shows that all instances with $|\mathcal{I}| = 100$ and 150 are solved to optimality, while high-quality results are obtained for the instances with $|\mathcal{I}| = 200$. Table 3 shows that the BD algorithm generally performs well, but that its performance deteriorates seriously when $|\mathcal{J}|$ and $|\mathcal{K}|$ increase. For instance, it is noteworthy that, when $|\mathcal{J}| = 10$, the BD algorithm outperforms the BBC algorithm, both in terms of solution quality and time. However, when $|\mathcal{J}| = 15$, the situation is generally the opposite. In particular, the average CPU times are smaller for BD when $|\mathcal{J}| = 10$, but larger when $|\mathcal{J}| = 15$. Table 4 shows that the best solutions identified by the algorithms are of similar quality for all sets of instances, except for the largest instances

	GAP (%)		$TIME$ (s)	
$\lvert\mathcal{I}\rvert$, $\lvert\mathcal{J}\rvert$, $\lvert\mathcal{K}\rvert$	AVG	STD	AVG	STD
100, 10, 3	0	0	9	1
100, 15, 3	0	0	121	30
100, 10, 4	0	0	17	2
100, 15, 4	0	0	167	17
150, 10, 3	0	0	52	7
150, 15, 3	0	0	289	38
150, 10, 4	0	0	137	18
150, 15, 4	0	0	641	71
200, 10, 3	1.76	1.03	1103	257
200, 15, 3	2.00	1.14	1369	210
200, 10, 4	0.26	0.22	919	177
200, 15, 4	0.94	0.32	2335	231

Table 2: BBC results on 360 instances (30 in each set) without UT assignment costs; Gap and Time (limit: 3600s): average (AVG) + standard deviation (STD)

	GAP (%)		$TIME$ (s)	
$\lvert\mathcal{I}\rvert$, $\lvert\mathcal{J}\rvert$, $\lvert\mathcal{K}\rvert$	AVG	STD	AVG	STD
100, 10, 3	0	0	2	0
100, 15, 3	0	0	390	115
100, 10, 4	0	0	5	1
100, 15, 4	0.78	0	964	230
150, 10, 3	0	0	23	5
150, 15, 3	0	0	292	117
150, 10, 4	0	0	107	21
150, 15, 4	3.97	0.99	1431	324
200, 10, 3	0	0	188	13
200, 15, 3	6.08	1.11	2075	320
200, 10, 4	0	0	550	62
200, 15, 4	19.05	2.47	2978	246

Table 3: BD results on 360 instances (30 in each set) without UT assignment costs; Gap and Time (limit: 3600s): average (AVG) + standard deviation (STD)

$\lvert\mathcal{I}\rvert$, $\lvert\mathcal{J}\rvert$, $\lvert\mathcal{K}\rvert$	$DIFF(\%)$ AVG	$DIFF(\%)$ STD
100, 10, 3	0	0
100, 15, 3	0	0
100, 10, 4	0	0
100, 15, 4	0.26	0.26
150, 10, 3	0	0
150, 15, 3	0	0
150, 10, 4	0	0
150, 15, 4	0	0
200, 10, 3	-0.36	0.36
200, 15, 3	-0.69	0.51
200, 10, 4	0	0
200, 15, 4	6.34	1.63

Table 4: Difference in upper bounds between BBC and BD on 360 instances (30 in each set) without UT assignment costs: average (AVG) + standard deviation (STD)

of size $|I| = 200$, $|J| = 15$ and $|K| = 4$, for which the BBC algorithm is far superior to the BD approach. Indeed, on the whole set of 360 instances, the BBC algorithm found a better quality solution for 14 out of the 360 instances, mostly on the largest instances of size $|I| = 200$, $|J| = 15$ and $|K| = 4$, whereas BD computed a better quality solution for just 3 instances. In addition, the BBC approach was able to certify the optimality of the computed solutions for all instances, except 29, while the BD algorithm was not able to certify the optimality for 57 instances, for which the time limit was reached.

We also performed additional experiments on larger instances, obtained by combining $|I| = \{200, 300, 500\}$ with $|J| = \{20, 30, 50\}$ and $|K| = 4$. For each size, we generated 10 instances, along the lines indicated above. The total number of such additional larger instances is thus 90. The objective was to emphasize the scalability and the robustness properties of the proposed BBC method, compared to the more traditional BD approach. For these experiments, a time limit of 7200 seconds was used. Tables 5, 6 and 7 report the results obtained on this set of larger instances, using a format similar to the last three tables of results.

These results clearly show the superiority of the BBC algorithm as the size increases. As shown in Table 7, while the BD method is still somewhat competitive for instances with $|I| = 200$, it is significantly outperformed on instances of size $|I| = 300$ and 500. On the whole set of 90 instances, it is noteworthy that the BBC method was able to find a feasible solution for all instances, while the BD algorithm could not determine any feasible solution for 5 instances out of 90 (these instances were not considered in Tables 6 and 7). The BBC approach was able to certify the optimality of the computed solutions for 5 instances out of 90, whereas the BD method was able to certify the optimality in 6 cases. However, better quality solutions were generally found by the BBC approach. Indeed, the BBC approach found a better solution for 43 out of the 90 instances, while the BD algorithm was able to return a better quality solution in just 9 cases out 90.

$\lvert\mathcal{I}\rvert, \lvert\mathcal{J}\rvert, \lvert\mathcal{K}\rvert$	GAP (%)		$TIME$ (s)	
	AVG	STD	AVG	STD
200, 20, 4	0.89	0.31	4860	853
300, 20, 4	7.18	1.48	7025	24
500, 20, 4	25.12	1.62	7167	3
200, 30, 4	2.52	0.52	6154	948
300, 30, 4	8.78	1.03	6970	4
500, 30, 4	16.53	0.82	7121	13
200, 50, 4	3.39	0.44	6809	17
300, 50, 4	6.79	1.05	6984	8
500, 50, 4	11.67	0.58	7089	13

Table 5: BBC results on 90 larger instances (10 in each set) without UT assignment costs; Gap and Time (limit: 7200 s): average (AVG) + standard deviation (STD)

$\lvert\mathcal{I}\rvert, \lvert\mathcal{J}\rvert, \lvert\mathcal{K}\rvert$	GAP (%)		$TIME$ (s)	
	AVG	STD	AVG	STD
200, 20, 4	21.50	3.85	5736	955
300, 20, 4	56.47	7.44	7173	4
500, 20, 4	154.18	24.46	7169	11
200, 30, 4	33.80	5.81	5753	587
300, 30, 4	76.11	3.35	7169	17
500, 30, 4	145.99	11.49	7180	4
200, 50, 4	69.06	7.88	6491	694
300, 50, 4	90.18	13.05	6632	549
500, 50, 4	161.01	8.50	7181	3

Table 6: BD results on 90 larger instances (10 in each set) without UT assignment costs; Gap and Time (limit: 7200s): average (AVG) + standard deviation (STD)

$\vert\mathcal{I}\vert$, $\vert\mathcal{J}\vert$, $\vert\mathcal{K}\vert$	$DIFF(\%)$ AVG	STD
200, 20, 4	0.45	0.45
300, 20, 4	10.79	4.10
500, 20, 4	29.93	11.19
200, 30, 4	0.51	0.51
300, 30, 4	4.16	1.67
500, 30, 4	10.68	3.34
200, 50, 4	2.25	0.96
300, 50, 4	0.81	1.16
500, 50, 4	7.31	3.42

Table 7: Difference in upper bounds between BBC and BD on 90 larger instances (10 in each set) without UT assignment costs: average (AVG) + standard deviation (STD)

5 Conclusion

In this paper, we considered a location-design problem that arises from the development of network reconfiguration algorithms for reducing the power consumption of Wireless Local Area Networks (WLANs). The resulting optimization problem, called the green WLAN problem, or GWLANP, was formally described and modelled. While the GWLANP was introduced in [12], we studied a non-trivial extension of the problem where the power consumed by each access point depends on the demands assigned to the access points. An exact solution method, based on the branch-and-Benders-cut framework, was developed. The results on a large set of realistic instances showed that the approach is effective and efficient, as it delivers high-quality solutions in limited computational effort. Furthermore, when comparing its performance on the special case solved by the algorithm proposed in [12], we showed that the proposed algorithm is preferable in terms of solution quality, scalability and robustness.

This work opens up interesting research perspectives. In particular, it would be interesting to generalize the proposed branch-and-Benders-cut approach to other optimization problems. Several features of the algorithm seem to be generalizable, in particular, the inclusion of the Benders subproblem variables in the formulation of the master problem and the addition of cuts that exclude feasible solutions, but that are not based on the objective function value, as in classical Benders decomposition methods.

Acknowledgements

The work of Bernard Gendron was funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grant 184122-2010. The work of Maria Grazia Scutellà was funded by project PRIN 2012, “Mixed-Integer Nonlinear Optimization: Approaches and Applications” (2012JXB3YF). The work of Rosario Garroppo, Gianfranco Nencioni and Luca Tavanti was partially supported by the Italian Ministry of University and Research under the "GreenNet (Greening the Networks)" FIRB project. This support is gratefully acknowledged.

References

[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. Technical report, G-2012-57, GERAD, Montreal, Canada, 2013.

[2] E. Balas and R. Jeroslow. Canonical cuts on the unit hypercube. *SIAM Journal of Applied Mathematics*, 23(1):61–69, 1972.

[3] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proceedings of the 1st international conference on mobile systems, applications and services (MobiSys)*, pages 303–316. ACM Publisher, New York, USA, 2003.

[4] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[5] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, 25(1):13–26, 2013.

[6] Cisco. Aironet 3600 Series Access Point, `http://www.cisco.com/en/US/products/ps11983/index.html`, September 2014.

[7] G. Codato and M. Fischetti. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4):756–766, 2006.

[8] European Commission. COST 231 – Digital Mobile Radio Towards Future Generations Systems. Technical report, 1999.

[9] R.S. de Camargo, G. de Miranda Jr., and R.P.M. Ferreira. A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, 39(5):329–337, 2011.

[10] B. Fortz and M. Poss. An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359–364, 2009.

[11] A. Garcia-Saavedra, P. Serrano, A. Banchs, and G. Bianchi. Energy consumption anatomy of 802.11 devices and its implication on modeling and design. In *Proceedings of the 8th international conference on emerging networking experiments and technologies, CoNEXT '12*. ACM Publisher, New York, USA, 2012.

[12] B. Gendron, R.G. Garroppo, G. Nencioni, M.G. Scutellà, and L. Tavanti. Benders decomposition for a location-design problem in green wireless local area networks. *Electronic Notes in Discrete Mathematics*, 41:367–374, 2013.

[13] J.N. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96(1):33–60, 2003.

[14] A.P. Jardosh, K. Papagiannaki, E.M. Belding, K.C. Almeroth, G. Iannaccone, and B. Vinnakota. Green WLANs: on-demand WLAN infrastructures. *Mobile Networks and Applications*, 14(6):798–814, 2009.

[15] D. McDaniel and M. Devine. A modified Benders partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319, 1977.

[16] J. Naoum-Sawaya and S. Elhedhli. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210:33–55, 2013.

[17] A. Pigatti, M. Poggi de Aragão, and E. Uchoa. Stabilized branch-and-cut-and-price for the generalized assignment problem. *Electronic Notes in Discrete Mathematics*, 19:389–395, 2005.