

Optimal deployment of indoor wireless local area networks

Antoine Oustry^{1,3}, Marion Le Tilly², Thomas Clausen³, Claudia D'Ambrosio³, and Leo Liberti³

¹Ecole des ponts ParisTech, Champs-sur-Marne, France

²Ecole polytechnique fédérale de Lausanne, Lausanne, Switzerland

³LIX CNRS, École polytechnique, Institut Polytechnique de Paris, Palaiseau, France

We present a two-phase methodology to address the problem of optimally deploying indoor wireless local area networks. In the first phase, we use Helmholtz's equation to simulate electromagnetic fields in a typical environment such as an office floor. The linear system which results from the discretization of this partial differential equation is solved with a state-of-the-art library for sparse linear algebra. In the second phase, we formulate the network deployment problem in the setting of Binary Linear Programming. This formulation employs the simulator output as input parameters, and jointly optimizes the number of Access Points, their locations, and their emission channels. We prove that this optimization problem is NP-Hard, and use mathematical programming based techniques and heuristics to solve it. We present numerical experiments on medium-sized buildings.

1 Introduction

Optimization problems related to Wireless Networks deployment have attracted considerable interest in the Operations Research (OR) and Mathematical Programming (MP) literature, especially since the 1990s. A particularly important problem, that of optimally deploying a Wireless Local Area Network (WLAN) in indoor environments, was mostly studied in the literature from the more technological point of view of computer networks, rather than using the algorithmic approaches afforded by OR/MP methods. This paper intends to bring the benefits of OR/MP to the OPTIMAL WLAN DEPLOYMENT (OWLD) problem. We note that the input data for the OWLD should be a precise description of the electromagnetic field at every point of the volume of interest. The approach proposed in this paper estimates the field intensity from a solution of the differential equations of the field.

1.1 OR methods in wireless networks

We first provide a minimal survey of the impact of the OR/MP culture in some optimization problems related to wireless networks. The earliest models for radio network planning were related to the MINIMUM DOMINATING SET (MDS) problem [1]. More realistic and application-specific models followed, opening different research subdomains depending on the targeted application.

- Wireless Sensor Networks (WSN) are a set of spatially distributed sensors collecting and exchanging data and recording physical conditions [2]. The RELAY NODE PLACEMENT (RNP) problem is one of the major questions arising in the design of WSN. Connectivity being necessary for flow routing, RNPs are related to fundamental problems such as MINIMUM CONNECTED DOMINATING SET (MCDS) [3], MINIMUM STEINER TREE (MST) [4, 5], and their variants [6, 7, 8]. In particular, finding the maximal data-rate of wireless connections subject to edge capacities leads to the EDGE CAPACITATED MST [9]. More technological and applied works focus on deployment processes improving network performance [10], lifetime [11, 12], and energy-efficiency [13].
- The FREQUENCY ASSIGNMENT PROBLEM (FAP) is the problem of assigning a frequency to each emitter of a wireless network while maximizing the Quality of Service (QoS) [14]. According to [15], “the FAP is probably the telecommunication application which has attracted the largest attention in the OR literature, both for its practical relevance and for its immediate relation to classical combinatorial optimization problems”. Depending on technological objective and constraints, the FAP may assume very different forms. The main variants of this problem are MAXIMUM SERVICE FAP, the MINIMUM ORDER FAP, the MINIMUM SPAN FAP, and the MINIMUM INTERFERENCE FAP [14].
- The Universal Mobile Telecommunications System (UMTS) is a mobile cellular system for networks based on the GSM standard. Due to the large operational cost for the mobile service operators, there is a huge need of optimizing at regional scale the base stations locations and configurations. This requires solving problems related to frequency assignment, emission power, antenna height/tilt/orientation, and more. The first models were simplified and based on the MDS [1] and the CAPACITATED FACILITY LOCATION (CFL) [16] problems. In the 2000s, much work has been carried out in order to jointly optimize the location and configuration of the base stations using MP tools [17, 18, 19, 20, 21, 22]. The overall model presented in [15] takes into account numerous technological constraints, such as uplink and downlink minimal Signal-to-Interference Ratio (SIR), maximum emission power, antenna height constraints, antenna tilt and assumes a SIR-based power control mechanism.
- The importance of the OWLD problem arises with the deployment of large-scale public and private WLANs, such as in airports, hotels or other buildings. This problem is related to the UMTS planning problem because (a) the location and the configuration of the Access Points (APs) may be optimized jointly and (b) interferences impact on the QoS. WLAN deployment, however, differs from UMTS planning in the following ways.
 - The configuration possibilities are more limited for WLAN deployment than UMTS.
 - The number of available frequencies is more limited for WLAN deployment [14]: this is why interference constraints are even more important.
 - The communication protocols used are different.
 - Indoor radio propagation is far more difficult to predict than outdoor and long-range propagation of GSM frequencies: in a building sev-

eral physical effects, such as, e.g., reflections, diffractions, and self-interference, may impact the radio propagation. Hence, computing the electromagnetic field generated by an emitter in a building requires advanced simulators, based on the precise knowledge of the building's architecture.

The last point is a possible explanation as to why WLAN deployment problems have not been extensively studied from the point of view of OR/MP methods. Most of works devoted to optimize WLAN deployments appear to have been proposed by researchers in computer networking.

In this paper, we propose a global approach for the deployment of a WLAN network, which combines simulation and optimization. First, we provide a mathematical model for radio propagation in a building floor in order to perform accurate predictions via simulation. We overcome some remarkable theoretical and computational obstacles posed by the indoor environments, which possess a very high number of reflection and diffraction sources. The precision of the model and the corresponding simulation is critical for OWLD, since optimization using wrong input data would obviously yield wrong solutions. Based on the qualitative input data provided by the simulation tool, we then propose an optimization model for the OWLD problem, which brings together the most important features found in the literature.

1.2 The state of the art on the OWLD problem

Many different approaches exist in order to streamline WLAN deployment processes in indoor space. In this subsection we try to summarize the main ingredients in optimizing WLAN deployments: decision variables, input data, optimization criteria, constraints, and algorithmic approaches.

1.2.1 Decision variables

The decisions involved in deploying a WLAN are:

- AP positions, either from a continuous or discrete optimization point of view;
- AP antenna orientation;
- AP emission power;
- AP emission frequencies.

The simultaneous decision of AP placement and frequency assignment is discussed in the literature to some extent [23, 24, 25, 26]. Depending on the approach, the spatial network coverage may be given or subject to optimization.

1.2.2 Simulating indoor radio-wave propagation

Optimizing any wireless network, and particularly a WLAN infrastructure in a building, requires a detailed knowledge of the way the electromagnetic waves propagate in the building, so as to predict the coverage area of each AP. Since there is an uncountable number of spatial point in a Euclidean space model of a building, it is obviously impossible to obtain such knowledge through empirical measurements. This is why it is necessary to have a reliable model to predict how electromagnetic waves propagate in a complex indoor environment. This step is

crucial since it will be the input of the optimization problem. For more details, the strong impact of the Physical Layer (PHY) modeling on network performance prediction is illustrated in [27].

Next, we briefly discuss some relevant propagation models for WLAN signals in building floors consisting of rooms and aisles.

- *Empirical approaches.* These are based on statistical models of propagation [28, 29, 30]; they predict the average behavior of waves in typical environments, but fail to provide accurate predictions in each room of the WLAN coverage area. Such approaches are widely used in network design because of their low computational requirements. They model the path-loss L (in dB) between emitter e and receiver r , which is equivalent to the power gain ratio expressed in logarithmic scale: $L = 10 \log(\frac{P_e}{P_r})$, where P_e is the power emitted by e and P_r the power received by r , both expressed in watts (W).
 - One-slope model. The path-loss L (in dB) is simply a function of the distance d between transmitter and receiver antennas:

$$L(d) = L_0 + 10n \log(d),$$

where L_0 (in dB) is a reference loss value for a unit (1m) distance, n is a power decay factor, and d is a distance (expressed in m). L_0 and n are empirical parameters for a given environment, which fully control the prediction. In a barrier-free environment, theoretical values of L_0 and n can be determined for a given wavelength. In practice, there are always obstacles to signal propagation and there is then no other way to choose values for L_0 and n than calibration from measured data, using linear regression such that the difference between the measured and estimated path losses is minimized in a mean square error sense [31].

- Multi-wall model. A semi-empirical multi-wall model provides much better accuracy than the one-slope model, and the results are more site-specific. The path-loss L (in dB) is a function of the distance between transmitter and receiver antennas but also of simple architectural properties. We have

$$L(d) = L_0 + 10n \log(d) + \sum_{i=1}^N k_i L_i + \kappa_f \Lambda_f,$$

where k_i is the number of walls of type i (for i ranging over all considered wall types) between transmitter and receiver antennas, L_i (in dB) is the attenuation factor for the i -th wall type, N is the number of wall types, κ_f is the number of floors between transmitter and receiver, and Λ_f (in dB) is the floor attenuation factor. There are also no universal values for the parameters L_i and Λ_f , which have to be calibrated by means of a measurement campaign.

These models are easy to use and can produce their output quickly. However, their accuracy is limited. The loss of accuracy is mostly due to the lack of architecture details they take into account; specifically, they neglect the physical effects of diffraction, self-interference, and so on.

- *Geometrical optics based modelling.* Ray-tracing techniques are among the most popular methodologies for indoor radiowave propagation simulation

[32, 33, 34, 35]. Ray-tracing simulations describe the physical wave propagation process, based on geometrical optics and the uniform geometrical theory of diffraction. Using Fermat’s least time principle, a ray-tracing simulation determines a ray’s trajectory between source point and some given field locations, which yields the propagation loss at these locations. Unlike empirical approaches, ray-tracing simulations take physical effects and indoor architecture into account, thus giving more precise results. Yet, their computational complexity is proportional to the number of rays *launched* by the source and grows exponentially with the number of reflections each ray undergoes. Therefore, the number of simulated reflections is usually limited. This leads to some undesirable effects due to the angular discretization. An attempt to decrease these effects, proposed by [34], consists in simulating “tubes” rather than “rays”. Geometrical optic based methods are often presented in a three-dimensional (3D) setting; it turns out, however, that it is just as accurate, but more computationally efficient, to work with multiple two-dimensional (2D) simulations [36]. Most of the works in the field of geometrical optic based simulation are dedicated to the reduction of the computational complexity by designing efficient structures to represent the physical environment, so as to speed-up the computation of ray intersections [37, 38].

- *Finite differences modelling.* There exists a family of propagation models based on Maxwell’s equations. The first indoor radio propagation simulations with finite differences were been proposed in [39, 40]. Another discrete approach, called “ParFlow”, based on the cellular automaton formalism, and applied to urban micro-cellular GSM simulations, was proposed in [41]. The main advantage of such methods is to naturally model all propagation effects including reflection and diffraction. The price to pay is a high computational load, specially in a 3D setting. However, as explained in [42], such approaches appear to be well-suited to indoor environments since their computational complexity does not depend on the number of objects and reflections. Moreover, they can handle any shape of obstacle. In [42, 43, 44], the ParFlow theory is transposed to the frequency domain in order to compute steady-state propagation; the Multi-Resolution Frequency Domain ParFlow (MR-FDPF) algorithm allows the efficient solution of the associated linear system of equations. This method is presented either in a 2D or 3D setting; a so-called “2.5D” variant proposed in [43] enables the efficient simulation of radio propagation in multi-floor buildings. MR-FDPF was the first finite differences method capable of simulating electromagnetic fields in realistic buildings, with a prediction accuracy assessed by measurement campaigns [44].

1.2.3 Deployment optimization criteria

There is a large variety of WLAN deployment quality evaluation functions in the literature. These functions are sometimes used as optimization criteria, and sometimes they are imposed as constraints to be satisfied.

- *Cost criteria.* A first optimization criterion for the deployment of a wireless network is naturally the economic cost, which mainly depends on the number of APs used and site-specific installation costs [46, 47].

- *Coverage criteria.* This family of criteria, which is very common in the literature on WLAN optimization, includes indicators that describe how well the network can cover the target area in terms of space and signal strength. One can mention the covered surface criterion or the numbers of covered clients [49]. The “hard cover” constraint refers to the fact that each client must receive a signal with an intensity higher than a given threshold [23, 46, 47, 50, 51]. An optimization objective used in [50, 52] is the sum of power received by each client from the best AP server, whereas authors in [49, 52] use the min - max criterion of received power from the best server at the worse client location.
- *Interference and Quality of Service criteria.* Another family of criteria focuses on the QoS for the clients rather than the absolute strength of the received signal. The QoS mainly depends on the signal strength but also on the interference sources, this is why the SIR or the Signal Interference-plus-Noise Ratio (SINR) are often used. These criteria are either involved as a hard QoS constraint to enable data rate guarantee for every or are used as an aggregated criteria (average). Some articles distinguish uplink and downlink SIR [46, 47, 51]. Another QoS metric is the Maximum of Channel Utilization, which designates the maximum traffic loads assigned to an AP. This is a practical metric for the wireless network performance, because it explains qualitatively its congestion [23]. More sophisticated approaches use simulation tools to compute QoS metrics depending on the selected networking protocol [53].
- *Client positioning accuracy criterion.* In some cases, a deployment criterion may be the ability of the system to locate client devices by a triangulation process based on Wi-Fi communication protocol [54].
- *Multiobjective approach and aggregate criteria.* Some authors are interested in finding deployment solutions with balanced characteristics between different criteria among those mentioned above. In [55], a multiobjective approach is implemented to obtain such a balanced solution. Other approaches use aggregated criteria, which often are a linear combination of some of the above-mentioned criteria [49, 52, 53]. The weights before each criterion are often chosen heuristically.

1.2.4 Solution methods

The diversity of criteria yields a corresponding diversity of discrete, continuous, convex, and non-convex problems, as well as solution methods. Some approaches use Binary Linear Programming (BLP) or Mixed Integer Linear Programming (MILP) formulations and employ standard branch-and-bound solvers [23, 46, 50], or propose dedicated Dantzig-Wolfe decompositions [56]. Other approaches use convex optimization techniques, such as the Simplex Algorithm [52], the Bundle Methods [49], the Nelder-Mead Algorithm [49], the Quasi-Newton BFGS Algorithm and the conjugate gradient search procedure [52, 57]. A wide range of metaheuristics are also used in the literature of WLAN deployment optimization: genetic algorithms [49, 52, 58], Simulated Annealing [52], Large-Neighbourhood and Tabu Searches [47], Termite Colony Algorithm [48] or Particle Swarm Optimization [59, 60]. Finally, a few authors use techniques from Constraint Programming [51, 61] and Black-Box Optimization [53, 62].

1.3 Positioning in relation with the state of the art

In this paper we propose an integrated approach in two phases: in the first phase we estimate the force of the magnetic field generate by a source in the building floor by solving the Helmholtz differential equations. In the second phase we formulate the OWLD optimization problem using input from the first phase, and we solve it using a number of exact and heuristic techniques based on MP. The approaches found in the literature to address the OWLD differ in either the first or the second phase or both. An integrated approach was never proposed so far to the best of our knowledge.

1.3.1 Simulation methodology

The main goal of the project that motivated this paper was to find a site-specific procedure that could take into account the shape of the building hosting the wireless network. We were looking for a simulation methodology more realistic than an empirical model; we were interested in approaches based on Maxwell's equations [39, 40, 41, 42, 43, 44] because they appeared to be the canonical point of view in physics. They took into account the physical effects of interest, i.e. *reflections, diffraction, self-interference* or *corridor effect*. The alternative point of view, provided by geometrical optics, appeared more complicated to implement. Inspired by the MR-FDPF method [42, 43, 44], we also chose to work in the Fourier domain in order to obtain a partial differential equation (PDE) without time dependence and with uncoupled frequencies: the Helmholtz equation. We therefore solved this PDE using simple finite difference schema by a state-of-the-art sparse linear solving library [63]. This way, we obtained a similar time complexity as the MR-FDPF method, with an easier implementation.

1.3.2 Modelling the deployment problem

In this section we list the techniques on which our methodology is based.

- We use Binary Linear Programming as in [23, 46, 50] to model the overall decision problem, which employs elements from facility location, frequency assignment and knapsack.
- We optimize jointly over client positions, AP positions and channel assignment.
 - Clients: we consider a certain number of given positions to place clients at. In the case of a corporate building for instance, these positions would correspond to workstations for employees, needing a WiFi connection. Each client requires a certain uploading and downloading data rate. For linear modelling purposes, we consider only two possible alternatives for a client: it is either connected and receives the demanded data rates or is disconnected.
 - Access points placement: we consider also a list of predetermined available locations to place APs. A specific cost corresponds to each of these locations, describing the installation and maintenance costs. We consider APs of only one type. We chose not to consider repeaters, which are rarely used in large-scale deployment project due to their lack of reliability.

- Available channels and interference: we consider a set \mathcal{C} of available communication channels (i.e. frequencies) at our disposal for the WiFi network deployment. The IEEE 802.11 WLAN standard defines a fixed number of channels for use by APs and mobile users. A total of 13 frequencies is available but these channels are overlapping. In practice only 3 frequencies can be used in the same physical neighborhood simultaneously without causing interference [14]. Therefore we considered here to have three available non-interfering channels: $|\mathcal{C}| = 3$.
- Abandonment of the APs orientation variables: after exchanging with network experts [45], we thought it was not realistic to optimize in such a level of detail because the impact of the antennas orientation is negligible compared to unpredictable perturbation sources such as moving obstacles.
- Presence of extra noise: our model involves an ambient noise that represents the thermal noise and the presence of perturbation sources from the surrounding environment. Taking into account ambient noise leads to more robust solutions.
- Impact of the interference and of the noise on the QoS: it is well known that the interference between devices impacts the data communication rate. We use the Shannon-Hartley theorem to express channel capacity as a function of noise and interference levels as in [46, 47, 51], for uplink and downlink connections. The theoretical capacity of a wireless channel is given by Shannon's law:

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (1)$$

where B is the available bandwidth in Hz (Hertz), S the signal strength and N the interference-plus-noise level in W (Watts). For fixed bandwidth \bar{B} and signal strength values \bar{C} and \bar{S} , the maximal interference-plus-noise level compatible with data rate \bar{C} is thus:

$$N_{\max}(\bar{C}, \bar{B}, \bar{S}) = \frac{\bar{S}}{2^{\bar{C}/\bar{B}} - 1}. \quad (2)$$

The presence of an ambient noise implies that any connection in our deployment will comply not only with a minimum SINR ratio but also with a minimum signal strength in absolute value.

- Communication protocol: an important modeling issue is the role of the communication protocol, which greatly impacts the data communication rate in practice. In particular, the main issue is the interaction of multiple client devices connected to a same AP. In this article, we refer to the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) method used in the IEEE 802.11 standard, which is the most common WLAN protocol [64]. The CSMA/CA method uses a collision avoidance mechanism based on a principle of prior negotiation and reciprocal acknowledgements between sender and receiver. The station wishing to transmit listens to the network. If the network is busy, the transmission is delayed. Otherwise, if the media is free for a given time then the station can transmit. No two clients may emit at the same time. Hence, we can assume that clients connected to the same AP do not interfere. We denote the set of any AP and its connected

clients as *cluster*. Inside a cluster, every device is tuned to the same channel. The maximal data rate implied by CSMA/CA mode for each cluster will be taken into account as a capacity γ for each AP. Two clusters tuned to different channels are assumed not to interfere with each other. By contrast, two clusters tuned to the same channel do interfere: more explicitly, in Shannon’s law Eq. (1) the variable N is a sum of an ambient noise θ and of all signals received from other clusters on the same channel, i.e. all the interfering signals.

- Economic objective: we study the trade-off between installation and maintenance costs, and the total data-rate that the infrastructure can provide, as in the UMTS network planning problem presented in [15]. We penalize each data-flow unit that is requested by a client but not provided by the network as deployed, at a cost ρ .

1.4 Outline of the article

The rest of this article is organized as follows: Section 2 presents our methodology for simulating the propagation of electromagnetic waves in an indoor environment, which is based on a finite-difference scheme of the Helmholtz PDE; Section 3 introduces a BLP formulation for the OWLD problem based on the output from the simulator, and another BLP formulation for a natural relaxation of OWLD problem; Section 4 states that the OWLD problem and its relaxation are both strongly NP-hard; Section 5 proposes heuristic algorithms to try to find better solutions in case an exact algorithm would fail to close the optimality gap; Finally, Section 6 presents our numerical experiments and the results obtained.

2 Solving the Helmholtz equations

In this section, we introduce a numerical methodology used to predict radio propagation in any building whose shape and materials are known precisely, as detailed below. Our approach is freely inspired by the MR-FDPF method of Gorce et al. [42, 43, 44] but it leads to an easier implementation based on the sparse linear algebra library SuperLU [63]. It is a simple finite difference schema to simulate Helmholtz equation, a PDE defined on complex numbers which is the transposition of the classic wave equation in frequency domain.

2.1 Why the Helmholtz equation?

2.1.1 The classical wave equation

Following the same modelling simplifications from [41] and [65], for instance, we consider a 2D environment and a transverse electromagnetic field, enabling us to model it as scalar wave. We start from the following classical two-dimensional wave equation in an heterogeneous medium:

$$\Delta u(x, y, t) - \mu \epsilon \partial_t^2 u(x, y, t) = -s(x, y, t) \quad (3)$$

where $u(x, y, t)$ is the unknown real-valued function representing the scalar wave, $s(x, y, t)$ is a source term, $\epsilon(x, y)$ is the local electric permittivity and $\mu(x, y)$ the local permeability. The physical values ϵ and μ depend on the material: they represent the architecture of the floor.

2.1.2 The frequency domain

In order to eliminate the time differential, the Fourier transform is applied to the wave equation (3), which gives the Helmholtz equation:

$$\Delta\Psi(x, y, \omega) + \omega^2\mu\epsilon\Psi(x, y, \omega) = -S(x, y, \omega) \quad (4)$$

where Ψ is the Fourier transform of u and S is the Fourier transform of s . The main advantage of this wave equation in the frequency domain is the absence of partial derivative with respect to ω , the angular frequency variable: every frequency can be studied independently, leading to the case of one single harmonic source. In the following, we assume that the source signal is only made of its carrier frequency f_c , which corresponds to the angular frequency $\omega_c = 2\pi f_c$. The Helmholtz equation can thus be written:

$$\Delta\Psi(x, y) + \omega_c^2\mu\epsilon\Psi(x, y) = -S(x, y, \omega_c). \quad (5)$$

The scalar field Ψ represents the intensity of the electromagnetic field in the steady state corresponding to frequency f_c . In such a steady state there is no propagation anymore. We let c_0 denote the light speed in vacuum and $\lambda_c = \frac{c_0}{f_c}$ denotes the carrier wavelength.

2.2 Boundary conditions and the additional diffusive term

2.2.1 Boundary conditions

In any PDE, it is crucial to set the boundary conditions of the equation in order to define the problem exactly. Here we impose a Dirichlet condition: the definition domain is a rectangle $[0, L] \times [0, \ell]$ and Ψ has to be null on the boundary. However, setting a null field value on the boundary creates fictitious reflections in the simulation. Indeed this Dirichlet condition corresponds to a situation where the boundary is a perfectly reflecting material: no outgoing energy transfer.

2.2.2 Avoiding spurious reflections

In preliminary simulations, fictitious patterns of interference were observed: the reflections on the walls and on the boundary were overestimated. A solution employed in [43] consists in adding a diffusive term in the boundary, in the walls. This diffusive term models walls energy absorption and the energy transfer to the outside. Helmholtz equation (5) can now be written:

$$\Delta\Psi(x, y) + \mu(x, y) \left(\omega_c^2\epsilon(x, y) - i\omega_c\sigma(x, y) \right) \Psi(x, y) = -S(x, y, \omega_c) \quad (6)$$

where $\sigma(x, y)$ is a fictive electric conductivity that is strictly positive in the walls and in the boundary (see Assumption 1). The complex two-dimensional equation (6) is the ultimate PDE that we simulate.

2.3 Finite difference approach

2.3.1 Discretization

The rectangle $[0, L] \times [0, \ell]$ is discretized into a grid $\{0, \dots, N_x - 1\} \times \{0, \dots, N_y - 1\}$, using the same step $h > 0$ for both dimensions. We let d denote the integer

$N_x N_y$ and index the grid by the set $\{0, \dots, d-1\}$. We assume that $N_x \geq 3$. We will discretize the field Ψ so that, for any $(p, q) \in \{0, \dots, N_x-1\} \times \{0, \dots, N_y-1\}$ the scalar Ψ_{jN_x+i} is an approximation of $\Psi(hp, hq)$. We define $\mathcal{G} = (\{0, \dots, d-1\}, \mathcal{E})$ the unoriented graph associated to grid $\{0, \dots, N_x-1\} \times \{0, \dots, N_y-1\}$. The classic discretization of Laplacian operator in (6) leads to the following equation:

$$\forall k \in \{0, \dots, d-1\}, \quad \sum_{\substack{l=0, \dots, d-1 \\ \{k, l\} \in \mathcal{E}}} \Psi_l + (\beta^2 n_k^2 - 4 - ih^2 \omega_c \alpha_k) \Psi_k = F_k \quad (7)$$

with the following conventions:

- normalization constant: $\beta = \frac{\omega_c h}{c_0}$
- diffusive term: $\forall (p, q) \in \{0, \dots, N_x-1\} \times \{0, \dots, N_y-1\}, \quad \alpha_{qN_x+p} = \mu(hp, hq) \sigma(hp, hq)$
- refractive index: $\forall (p, q) \in \{0, \dots, N_x-1\} \times \{0, \dots, N_y-1\}, \quad n_{qN_x+p} = c_0 \sqrt{\mu(hp, hq) \epsilon(hp, hq)}$
- source term: $\forall (p, q) \in \{0, \dots, N_x-1\} \times \{0, \dots, N_y-1\}, \quad F_{qN_x+p} = -h^2 S(hp, hq, \omega)$.

We define the matrix $U \in \mathbb{R}^{d \times d}$ as the adjacency matrix associated to the grid graph \mathcal{G} and let $L = 4I_d - U$, where I_d is the $d \times d$ identity matrix. We also define the diagonal matrix $D \in \mathbb{R}^{d \times d}$ associated to vector $(\beta^2 n_k^2 - ih^2 \omega_c \alpha_k)_{0 \leq k \leq d-1}$. Defining also $A = D - L \in \mathbb{R}^{d \times d}$, the problem to be solved can be cast as following linear system

$$A\Psi = F, \quad \Psi \in \mathbb{R}^d. \quad (8)$$

Broadly speaking, the A matrix represents the architecture of the building while F represents the intensities and source positions. In the application cases we present here, we have to calculate the electromagnetic field produced from each source points, whose set is designated by $V \subset \{0, \dots, d-1\}$. For each source point $i \in V$ we need to solve a linear system with the same system matrix A but with specific right-hand side (RHS) member $F^i = P_i e_i \in \mathbb{R}^d$ where $P_i \in \mathbb{R}$ is the power of device i and $e_i \in \mathbb{R}^d$ is the canonical basis vector associated to i . In practice, P_i has only two possible values in our experiments, depending on whether i is a client device (P_{client}) or a candidate point to host an AP (P_{AP}). The electromagnetic field generated by $i \in V$ is the solution of linear system (8) with $F = F^i$; this solution is denoted Ψ^i . For any $(i, j) \in V \times V$, the power $p_{ij} \in \mathbb{R}_+$ received by j from emitter i is:

$$p_{ij} = |\Psi_j^i|$$

where Ψ^i is the solution of (8) associated to F^i . The matrix $p = (p_{ij})$ provides the input to the optimization problem introduced in Sect. 3. Before proving the invertibility of the system to be solved, we make a modeling assumption. This numerical condition avoids spurious reflections (see Sect. 2.2) and is useful to prove the invertibility.

Assumption 1. *For any cell (p, q) such that $p \in \{0, 1, N_x-2, N_x-1\}$ or $q \in \{0, 1, N_y-2, N_y-1\}$, we have $\alpha_{qN_x+p} > 0$.*

Proposition 1. *Under Assumption 1, the system matrix A is invertible.*

Proof. We take any complex vector $\Psi \in \mathbb{C}^d$ such that $A\Psi = 0$, and we are going to prove that $\Psi = 0$. This will prove the proposition since A is a square matrix.

1. First, we prove that for any $k \in \{0, \dots, d-1\}$ such that $\alpha_k > 0$, we have $\Psi_k = 0$. Decomposing matrix $D = R - iJ$, where R is the diagonal matrix associated to vector $(\beta^2 n_k^2)_{0 \leq k \leq d-1}$ and J the diagonal matrix associated to vector $(h^2 \omega_c \alpha_k)_{0 \leq k \leq d-1}$, matrix A can be written as $A = R - iJ - L$. Since $A\Psi = 0$, the following holds

$$\Psi^* R \Psi - i(\Psi^* J \Psi) - \Psi^* L \Psi = 0 \quad (9)$$

Since matrices R , J , and L are real symmetric matrices, we know that $(\Psi^* R \Psi, \Psi^* J \Psi, \Psi^* L \Psi) \in \mathbb{R}^3$. Hence, we can deduce from (9) that $\Psi^* J \Psi = 0$, by unicity of the imaginary part. Reformulating this, we have that $h^2 \omega_c \sum_{k=0}^{d-1} \alpha_k |\Psi_k|^2 = 0$. Dividing by $h^2 \omega_c > 0$, we have $\sum_{k=0}^{d-1} \alpha_k |\Psi_k|^2 = 0$. Since $\alpha_k |\Psi_k|^2 \geq 0$ for all $k \in \{0, \dots, d-1\}$, we deduce that $\alpha_k |\Psi_k|^2 = 0$ and thus $(\alpha_k > 0) \implies (\Psi_k = 0)$.

2. Based on this first point and using Assumption 1, we deduce that $\Psi_{qN_x+p} = 0$ for all (p, q) such that $p \in \{0, 1, N_x-2, N_x-1\}$ or $q \in \{0, 1, N_y-2, N_y-1\}$. This means that field vector Ψ is null in the proximity of boundaries.
3. We are now going to prove by finite strong induction over $k \in \{0, 1, \dots, d-1\}$ that $\Psi_k = 0$.

- Initialization: in the second argumentation point above, we proved that $\Psi_0 = 0$.
- Induction step: we take $k \in \{0, \dots, d-2\}$ and we assume that $\Psi_l = 0$ for all $l \in \{0, \dots, k\}$. We decompose k as $qN_x + p$. In a first case, we have that $p \in \{0, 1, N_x-2, N_x-1\}$ or $q \in \{0, 1, N_y-2, N_y-1\}$, and thus $\Psi_k = 0$ (see second argumentation point). In a second case, we have that $(p, q) \in \{2, \dots, N_x-3\} \times \{2, \dots, N_y-3\}$. We apply then the system equation (7) with a null RHS member and for index $\tilde{k} = k+1 - N_x$, which gives

$$\Psi_{k+2-N_x} + \Psi_{k-N_x} + \Psi_{k+1} + \Psi_{k+1-2N_x} + (\beta^2 n_k^2 - 4 - ih^2 \omega_c \alpha_k) \Psi_{k+1-N_x} = 0. \quad (10)$$

We recall that $N_x \geq 3$ by assumption, thus the integers $k+2-N_x, k-N_x, k+1-2N_x, k+1-N_x$ are less than or equal to k . Using the strong induction hypothesis, we know that $\Psi_{k+2-N_x} = \Psi_{k-N_x} = \Psi_{k+1-2N_x} = \Psi_{k+1-N_x} = 0$. Hence, equation (10) gives that $\Psi_{k+1} = 0$.

By strong induction, we proved that $\Psi_k = 0$ for all $k = 0, \dots, d-1$. □

2.3.2 Solution algorithm and time complexity

We need to solve $|V|$ linear systems sharing the same matrix $A \in \mathbb{R}^{d \times d}$ but corresponding to different RHS member $F \in \mathbb{R}^d$. Hence, it is worth computing the LU factorization of this matrix as a first step in order to speed up the solution time of every single linear system thereafter, see the next paragraph for the complexity details. One can benefit from the non-zero structure of the entries of A by ordering

the eliminations such that as few non-zeros as possible are generated in L and U, because the number of non-zeros determines the storage and time complexities of the triangular solves [66, 67]. For a 2-D regular grid like matrix A , we need $O(d^{\frac{3}{2}})$ floating point operations for a sparse factorization, and the L,U factors one obtains have $O(d \log d)$ non-zeros, see [68, 69]. Hence, based on the LU decomposition, a system is solved in $O(d \log d)$ floating point operations. Considering that $N_x \geq N_y$, the time complexity of the two algorithmic steps are thus the following:

- $O(N_x^3)$ for the LU factorization of the system
- $O(N_x^2 \log N_x)$ to compute the system solution based on the LU factorization.

Since we want to compute the electromagnetic field generated by $|V|$ different sources, we can see that the preliminar factorization enables to achieve it in $O(N_x^3 + |V|N_x^2 \log N_x)$ instead of $O(|V|N_x^3)$. The resulting time complexity is similar to the complexity of 2D MR-FDPF claimed in [42] but our approach lead to a more direct implementation by using a state-of-the-art linear algebra library.

2.4 The “2.5-th” dimension

To make the model fit reality, it is crucial to model indoor radio wave propagation in 3D environment. Transposing this finite-difference method in 3D would yield an excessive increase of the time required to solve the systems. Instead, we consider the 2.5D empirical approach presented in [43], which relies on the projections of the field in the floor $k - 1$ and $k + 1$ to compute the field in the floor k , using one of these alternatives:

- *Field Projecting* models the 3D propagation by projecting the field map (of the floor $k - 1$ and $k + 1$) through the ceilings with an attenuation coefficient depending on the nature of the ceilings.
- *Source Projecting* consists in projecting the sources (on floors $k - 1$ and $k + 1$) in floor k with an attenuation factor and then in computing the 2D propagation in floor k from this virtual source.
- A linear combination thereof.

3 Optimal WLAN deployment: MP formulations

In this section, we introduce a new WLAN deployment optimization problem that addresses many of the challenges that we found in the literature. We also introduce a simple relaxation of this deployment problem and we prove that both the problem and this relaxation are strongly NP-hard. Finally, we introduce in this section several solution algorithms.

3.1 Binary linear programming formulation

We denote by (P) the BLP formulation introduced in this subsection. This is the OWLD problem.

Sets of points We consider subsets of a 2D or 3D simulation grid:

- I defines the set of users (clients) to be covered.

- J defines the set of eligible positions for APs. A point which has to be covered and which is also a potential AP position is duplicated. We can therefore assume that $I \cap J = \emptyset$.
- $V = I \cup J$.

Set of channels \mathcal{C} denotes the set of available frequencies.

Parameters Table 1 introduces the parameters involved in our BLP formulation. We make two assumptions: (a) the data flows u_i and d_i are scaled by the available bandwidth, which is known to be the same for every channel; (b) the different channel carrier frequencies are distinct enough not to interfere, but close enough to assume that the signal strength p_{ij} does not depend on the channel. These assumptions are verified for any 3-frequency deployments with channels 1, 7, 13 of the 802.11g protocol [14]. Moreover, the fact that the signal strength does not significantly depend on the channel can be numerically checked with the simulation tool. Finally the last parameter of this optimization problem is a "big-M" constant defined as $M = \theta + \sum_{(i,j) \in V^2} p_{ij}$.

Notation	Index set	Meaning
$p_{ij} \in \mathbb{R}_+$	$V \times V$	Signal emitted by i and received by j computed with methodology presented in Sect. 2
$c_j \in \mathbb{R}_+$	J	AP installation cost at position j
$u_i \in \mathbb{R}_+$	I	Data flow (normalised by the bandwidth) client i wants to upload
$d_i \in \mathbb{R}_+$	I	Data flow (normalised by the bandwidth) client i wants to download
$K_{ij}^u \in \mathbb{R}_+$	$I \times J$	Maximal interference-plus-noise level at i for uplink connection between client i and AP j
$K_{ij}^d \in \mathbb{R}_+$	$I \times J$	Maximal interference-plus-noise level at j for downlink connection between client i and AP j
$\gamma \in \mathbb{R}_+$		Flow capacity of any AP
$\theta \in \mathbb{R}_+$		Ambient noise level
$\rho \in \mathbb{R}_+$		Marginal penalty cost for non-provided flow unit

Table 1: Parameters

Set of edges Given an ambient noise level $\theta > 0$, the set

$$E_\theta = \{(i, j) \in I \times J \mid (\theta \leq K_{ij}^u) \wedge (\theta \leq K_{ij}^d)\}$$

is the set of potential wireless connections, i.e. the set of all the user-AP pairs such that the signal strength is strong enough to enable uplink and downlink flow in spite of the ambient noise. We actually consider the undirected graph induced by $G = (V, E_\theta)$, which we denote in the same way with a slight abuse of notation. This undirected graph is bipartite.

Decision variables Table 2 introduces the decision variables of our BLP formulation. Having $x_i = 1$ means that device i is on, $w_{ic} = 1$ means that device i is tuned on channel c and $y_{ij}^c = 1$ describes that client i is served by the AP j on channel c . In this BLP problem, the total number of decision variables is $(1 + |\mathcal{C}|) \times |V| + |\mathcal{C}| \times |E_\theta|$.

Notation	Indices set	Meaning
$x_i \in \{0, 1\}$	V	Device i is on/off
$w_{ic} \in \{0, 1\}$	$V \times \mathcal{C}$	Device i emits on channel c
$y_{ij}^c \in \{0, 1\}$	$E_\theta \times \mathcal{C}$	Client i communicates with AP j on channel c

Table 2: Decision variables

Objective Minimize the sum of the device installation costs, and the cost for not covering some of the clients:

$$\sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i).$$

Constraints One coupling constraint:

- Each device is tuned to a unique channel:

$$\forall i \in V, \sum_{c \in \mathcal{C}} w_{ic} = x_i$$

The other constraints are channel dependant (uncoupled):

- Each client communicates with one AP:

$$\forall c \in \mathcal{C}, \forall i \in I, \sum_{\substack{j \in J \\ (i,j) \in E_\theta}} y_{ij}^c = w_{ic}$$

- Maximal capacity of an AP:

$$\forall c \in \mathcal{C}, \forall j \in J, \sum_{\substack{i \in I \\ (i,j) \in E_\theta}} (d_i + u_i) y_{ij}^c \leq \gamma w_{jc}$$

- A client can communicate with an AP only if both are tuned to the same channel:

$$\begin{aligned} \forall c \in \mathcal{C}, \forall (i, j) \in E_\theta, y_{ij}^c &\leq w_{ic} \\ \forall c \in \mathcal{C}, \forall (i, j) \in E_\theta, y_{ij}^c &\leq w_{jc} \end{aligned}$$

- Interference-plus-noise constraints at clients, to enable downlink flow:

$$\forall c \in \mathcal{C}, \forall (i, j) \in E_\theta, \theta + \sum_{\substack{k \in I \\ k \neq i}} p_{ki} (w_{kc} - y_{kj}^c) + \sum_{\substack{k \in J \\ k \neq j}} p_{ki} w_{kc} \leq K_{ij}^d + M(1 - y_{ij}^c) \quad (11)$$

- Interference-plus-noise constraints at candidates, to enable uplink flow:

$$\forall c \in \mathcal{C}, \forall (i, j) \in E_\theta, \theta + \sum_{\substack{k \in I \\ k \neq i}} p_{kj} (w_{kc} - y_{kj}^c) + \sum_{\substack{k \in J \\ k \neq j}} p_{kj} w_{kc} \leq K_{ij}^u + M(1 - y_{ij}^c). \quad (12)$$

Constraints (11) enable to certify that, if $y_{ij}^c = 1$ i.e. if client i is connected to AP j on channel c , the sum of the ambient noise and the interference from other clusters tuned on the same channel will not exceed the maximal bearable level K_{ij}^d . On the contrary if $y_{ij}^c = 0$, this constraint is necessarily satisfied by definition of constant M . Constraints (12) are similar to constraints (11) but looking at the maximal interference-plus-noise level at APs so as to make the uplink flows possible.

3.2 Relaxing the interference-plus-noise constraints

In the case where the number of available channels is large (i.e. greater than the number of deployed APs), there is no interference problem between “clusters” anymore since each used channel is assigned to a unique AP. In such a case one can model the problem with less variables. We let (R) be the relaxation of (P) obtained by removing the channel bound constraints.

Decision variables Table 3 introduces the decision variables of this relaxed problem.

Notation	Indices set	Meaning
$x_i \in \{0, 1\}$	V	Device i is on/off
$y_{ij} \in \{0, 1\}$	E_θ	Client i communicates with AP j

Table 3: Decision variables

Objective Minimize the sum of the installation and penalty cost:

$$\sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i).$$

Constraints

- Each client communicates with one AP:

$$\forall i \in I, \sum_{\substack{j \in J \\ (i,j) \in E_\theta}} y_{ij} = x_i$$

- Maximal capacity of an AP:

$$\forall j \in J, \sum_{\substack{i \in I \\ (i,j) \in E_\theta}} (d_i + u_i) y_{ij} \leq \gamma x_j$$

- A client can communicate with an AP only if both are on:

$$\begin{aligned} \forall (i, j) \in E_\theta, y_{ij} &\leq x_i \\ \forall (i, j) \in E_\theta, y_{ij} &\leq x_j. \end{aligned}$$

We underline that the problem (R) depends on signal strength data computed with the simulator and on the parameters K_{ij}^u, K_{ij}^d through the definition of the set of edges E_θ . This relaxation has the advantage of providing a better lower bound than the continuous relaxation. Admittedly, we will show in the Sect. 4 that this optimization problem is NP-hard, but in practice it is easier to solve than the initial problem. Hence, we use this relaxation to build heuristic algorithms, see Sect. 5.

4 Complexity results

4.1 Complexity of the interference-free relaxation

Proposition 2. *The decision problem associated with (R) is strongly NP-complete.*

Proof. Table 4 introduces (R_D) , the decision problem associated with (R) , whereas Table 5 introduces the classic (BinPacking) problem, which is known to be strongly NP-complete [70]. We now show a polynomial reduction of (BinPacking) to prob-

Problem:	(R_D)
Instance:	Sets I, J Rational numbers $p_{ij}, c_j, u_i, d_i, K_{ij}^u, K_{ij}^d, \gamma, \theta, \rho$ as detailed in Table 1 Rational number Q
Question:	Does there exist a feasible solution of (R) with cost less than or equal to Q ?

Table 4: Decision problem (R_D)

Problem:	(BinPacking)
Instance:	Positive integers N, K with $K \leq N$ Positive integers s_1, \dots, s_N, L
Question:	Can we pack N objects of size s_1, \dots, s_N in K boxes of capacity L ?

Table 5: Decision problem (BinPacking)

lem (R_D) , which prove the proposition since (R_D) , as a BLP feasibility problem, is in NP. For any instance $\mathcal{I} = (N, K, s_1, \dots, s_N, L)$ of (BinPacking), we define an instance $\Phi(\mathcal{I})$ of (R_D) as:

- Sets: $I = \{1, \dots, N\}$ (clients), $J = \{N + 1, \dots, N + K\}$ (APs), $V = I \cup J$
- Parameters:
 - $\forall (i, j) \in V^2, p_{ij} = 1$
 - $\forall j \in J, c_j = 0$
 - $\forall i \in I, u_i = d_i = \frac{s_i}{2}$
 - $\forall (i, j) \in I \times J, K_{ij}^u = K_{ij}^d = 1$
 - $\gamma = L$
 - $\theta = 0$
 - $\rho = 1$

In such an instance, the graph G_θ is the complete bipartite graph associated to sets I and J .

- Level parameter $Q = 0$

With such a definition, it is clear that \mathcal{I} is a YES instance of (BinPacking) if and only if $\Phi(\mathcal{I})$ is a YES instance of (R_D) :

- If \mathcal{I} is a YES instance of (BinPacking), we can define the binaries y_{ij} associated to the presence of object i in box j . We also define $x_i = 1$ for all $i \in V$. Since $\sum_{i \in I} s_i y_{ij} \leq L$ for all $j \in J$, we have $\sum_{i \in I} (u_i + d_i) y_{ij} \leq \gamma x_j$ for all $j \in J$. Hence, (\mathbf{x}, \mathbf{y}) is a feasible solution of problem (R) associated to parameters of instance $\Phi(\mathcal{I})$, and has an objective value equal to $0 \leq Q$. This is why (\mathbf{x}, \mathbf{y}) certifies that $\Phi(\mathcal{I})$ is a YES instance of (R_D) .

- If $\Phi(\mathcal{I})$ is a YES instance of (R_D) , it exists (\mathbf{x}, \mathbf{y}) feasible in (R_D) and with value equal to zero. This means, in particular, that
 - $x_i = 1$ and $\sum_{j \in J} y_{ij} = 1$ for all $i \in I$
 - $\sum_{i \in I} (u_i + d_i) y_{ij} = \sum_{i \in I} s_i y_{ij} \leq Lx_j$ for all $j \in J$

Thus, the vector \mathbf{y} is an explicit allocation of N objects of size s_1, \dots, s_N in K boxes of capacity L , and certifies that \mathcal{I} is a YES instance of (BinPacking).

Moreover, $\Phi(\mathcal{I})$ is computable with a time complexity polynomial in N and, thus, polynomial in the coding size of \mathcal{I} , since $N \leq |\mathcal{I}|$. \square

4.2 Complexity of the OWLD problem

From the previous complexity result, the complexity of the OWLD problem can now be addressed.

Corollary 3. *The decision problem associated with (P) is strongly NP-complete.*

Proof. Table 6 introduces (P_D) , the decision problem associated with (P) . We are going to show a polynomial reduction of (R_D) to problem (P_D) , which will prove the proposition since (P_D) , as a BLP feasibility problem, is in NP.

Problem:	(P_D)
Instance:	Sets I, J, \mathcal{C} Rational numbers $p_{ij}, c_j, u_i, d_i, K_{ij}^u, K_{ij}^d, \gamma, \theta, \rho$ as detailed in Table 1 Rational number Q
Question:	Does there exist a feasible solution of (P) with cost less than or equal to Q ?

Table 6: Decision problem (P_D)

For any instance $\mathcal{I} = (I, J, p_{ij}, c_j, u_i, d_i, K_{ij}^u, K_{ij}^d, \gamma, \theta, \rho, Q)$ of (R_D) , we define an instance $\Psi(\mathcal{I})$ of (P_D) with same coefficients and $\mathcal{C} = J$. With such a definition, it is clear that \mathcal{I} is a YES instance of (R_D) if and only if $\Psi(\mathcal{I})$ is a YES instance of (P_D) :

- If \mathcal{I} is a YES instance of (R_D) , it exists a solution $(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^V \times \{0, 1\}^{E_\theta}$ feasible for this instance of problem (R_D) and with value less than or equal to Q . In order to construct a feasible solution of (P) , each cluster will be assigned its own channel. Formally, we define the binary vectors $\tilde{\mathbf{y}} \in \{0, 1\}^{E_\theta \times \mathcal{C}}$ and $\mathbf{w} \in \{0, 1\}^{V \times \mathcal{C}}$ as follow:
 - $\tilde{y}_{ij}^c = y_{ij} \delta_{jc}$ for all $(i, j) \in E_\theta$ and $c \in \mathcal{C}$, where δ_{jc} denotes the Kronecker product between j and c
 - $w_{ic} = x_i \delta_{ic}$ for all $i \in J$ and $c \in \mathcal{C}$
 - $w_{ic} = \sum_{\substack{j \in V \\ (i,j) \in E_\theta}} \tilde{y}_{ij}^c$ for all $i \in I$ and $c \in \mathcal{C}$.

By construction $(\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{w})$ is feasible for the instance $\Phi(\mathcal{I})$ of problem (P) and has value $\sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i)$ which is lower or equal to Q by definition of (\mathbf{x}, \mathbf{y}) . It certifies that $\Phi(\mathcal{I})$ is a YES instance of (P_D) .

- If $\Phi(\mathcal{I})$ is a YES instance of (P_D) , it exists then a solution $(\mathbf{x}, \tilde{\mathbf{y}}, \mathbf{w}) \in \{0, 1\}^V \times \{0, 1\}^{E_\theta \times \mathcal{C}} \times \{0, 1\}^{V \times \mathcal{C}}$ of this instance of problem (P_D) with value less than or equal to Q . We define then vector $\mathbf{y} \in \{0, 1\}^{E_\theta}$ as: $y_{ij} = \sum_{c \in \mathcal{C}} \tilde{y}_{ij}^c$ for all $(i, j) \in E_\theta$. By construction, (\mathbf{x}, \mathbf{y}) is feasible for the instance

\mathcal{I} of problem (R) , and has value $\sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i)$ which is lower or equal to Q . It certifies that \mathcal{I} is a YES instance of (R_D) .

Moreover the time complexity to compute $\Psi(\mathcal{I})$ is clearly polynomial in coding size of instance \mathcal{I} . \square

One can legitimately question the interest of introducing relaxation (R) which is NP-hard, as the original problem (P) is. In fact, we will see in Sect. 6 that the commercial branch-and-bound algorithm that we used solves this relaxation considerably faster than (P) , due to its smaller size and due to the absence of “big-M” constant in (R) . This provides good and fast lower bounds on the value of problem (P) .

5 Solution algorithms

To solve problem (P) , we first used a standard MILP solver implementation (see Sect. 6). It turns out, however, that this commercial solver cannot solve all the tested instances to optimality with a practically reasonable time limit. This is why we also propose several heuristic algorithms, in order to find better solutions for the cases where the MILP solver does not close the optimality gap in due time.

5.1 Greedy heuristics

The two first heuristic algorithms proposed here are based on a greedy approach. The second algorithm is a “multi-start” variant of the first one.

5.1.1 First greedy heuristic

This algorithm sequentially treats the different available frequencies. For a given frequency, the following steps are repeated as long it is possible to generate a new cluster with a positive score:

- For each AP that is not turned-on yet, the clients that can be associated with it are selected in a greedy way. The construction of each potential cluster must respect the bandwidth limitation of the AP, and the interference caused by these new devices must not prevent pre-existing connections on the same frequency. A score is assigned to each potential cluster, made of gains of connecting the selected clients minus the AP cost.
- The cluster with highest score is selected, the corresponding devices are turned-on if and only if the score is nonnegative. If the score is negative, the channel is considered “full”.

When the channel is “full”, all devices switched on this channel are filtered out and the process is repeated for a new channel. In order to present a pseudo-code of this first greedy heuristic, we need to introduce three auxiliary procedures. In all these procedures, the variable `ChannelEnv` stands for “channel environment” and designates the set of APs already selected and switched on this channel, jointly with the client devices connected to them. Given a device index $k \in V$ and a “channel environment”, i.e. a set of pairs (client, AP), the first auxiliary procedure (Algorithm 1) aims at checking if turning-on device k is possible in the sense that it does not hinder a connection present in the environment. Given

Algorithm 1 Procedure **CompatibleDevice**

```
1: procedure COMPATIBLEDEVICE( $k$ , ChannelEnv)
2:   for each pair  $(i, j)$  of client-AP connection in ChannelEnv do
3:     Let  $N_{\text{up}}$  be the interference-plus-noise level for uplink connection  $(i, j)$  in
     ChannelEnv
4:     Let  $N_{\text{down}}$  be the interference-plus-noise level for downlink connection  $(i, j)$  in
     ChannelEnv
5:     if  $N_{\text{up}} + p_{kj} > K_{ij}^u$  then  $\triangleright$  If the AP  $k$  hinders the  $(i, j)$  uplink connection.
6:       return FALSE
7:     end if
8:     if  $N_{\text{down}} + p_{ki} > K_{ij}^d$  then  $\triangleright$  If the AP  $k$  hinders the  $(i, j)$  downlink connection.
9:       return FALSE
10:    end if
11:  end for
12:  return TRUE
13: end procedure
```

an AP that we denote by a , given a set of possible client devices and a “channel environment”, the second auxiliary procedure (Algorithm 2) aims at checking if AP a may be turned-on without impacting the existing connections already set on this channel, and then compute a set of client devices (a cluster) in a greedy approach, knowing that the connection possibilities also depend on the “channel environment”. This procedure returns the “score” of this cluster, i.e. its contribution to the objective function.

Thanks to the ScoreCluster procedure, the third auxiliary procedure (Algorithm 3) tests all the possible new clusters to add on a current channel and select the “best” one, in a greedy approach too. With this auxiliary procedures, we can now introduce the pseudo-code of the greedy heuristic (Algorithm 4). In this algorithm, the size of array `patterns` is $|\mathcal{C}|$, the number of available channels. At the end of the algorithm, `patterns[i]` contains the list of (client, AP) pairs set on this channel.

5.1.2 Multistart greedy heuristic

The idea of this second algorithm comes from the observation that the first selected AP in the previous greedy heuristic highly influences the final results. Hence, it is relevant to run several times the greedy heuristic with a different AP selected first in the process (Algorithm 5).

5.2 Relaxation-based heuristics

We now present two heuristic algorithms based on the fact that the interference-free relaxation problem (R) presented in Sect. 3.2 is more efficiently solved by the used MILP solver than the original problem (P).

5.2.1 First relaxation-based heuristic

The principle of this algorithm is to use the solution of the interference-free relaxation (R) to build a solution of the original problem (P), by assigning a channel to

Algorithm 2 Procedure **ScoreCluster**

```
1: procedure SCORECLUSTER( $a, I, \text{ChannelEnv}$ )
2:   if compatible-device( $a, \text{ChannelEnv}$ ) then
3:     Sort clients by decreasing order of  $d_i + u_i$ 
4:      $S \leftarrow c_a$ 
5:      $\text{ChannelEnv2} \leftarrow \text{ChannelEnv} \cup \{a\}$ 
6:      $\Gamma \leftarrow \gamma$ 
7:     for  $i$  in  $I$  do ▷ Loop adding client devices in the cluster.
8:       if  $d_i + u_i < \Gamma \wedge \text{CompatibleDevice}(i, \text{env2})$  then
9:         Let  $N_{\text{up}}$  be the interference-plus-noise level for the  $(i, a)$  uplink connection in env2
10:        Let  $N_{\text{down}}$  be the interference-plus-noise level for the  $(i, a)$  downlink connection in env2
11:        if  $N_{\text{up}} \leq K_{ia}^u \wedge N_{\text{down}} \leq K_{ia}^d$  then ▷ If the connection  $(i, a)$  is possible.
12:           $S \leftarrow S - \rho(d_i + u_i)$ 
13:           $\Gamma \leftarrow \Gamma - (d_i + u_i)$ 
14:           $\text{ChannelEnv2} \leftarrow \text{ChannelEnv2} \cup \{i\}$ 
15:        end if
16:      end if
17:    end for
18:    return  $S, \text{ChannelEnv2}$ 
19:  else
20:    return 0,  $\text{ChannelEnv}$ 
21:  end if
22: end procedure
```

Algorithm 3 Procedure **ClusterSelection**

```
1: procedure CLUSTERSELECTION( $I, J, \text{ChannelEnv}$ )
2:    $S \leftarrow 0$ 
3:    $\text{ChannelEnv2} \leftarrow \text{ChannelEnv}$ 
4:   for  $a$  in  $J$  do
5:      $S_{\text{aux}}, \text{ChannelEnvAux} \leftarrow \text{ScoreCluster}(a, I, \text{ChannelEnv})$ 
6:     if  $S_{\text{aux}} < S$  then
7:        $S \leftarrow S_{\text{aux}}$ 
8:        $\text{ChannelEnv2} \leftarrow \text{ChannelEnvAux}$ 
9:     end if
10:  end for
11:  return  $S, \text{ChannelEnv2}$ 
12: end procedure
```

each cluster and possibly switching off the devices to make the solution feasible. Thus, this heuristic consists in successively solving two easier BLP problems:

1. **Positioning step:** Choose the AP-hosting candidate points, select the client devices to serve, and design "clusters" (set of client devices connected to a same AP) by solving the interference-free relaxation (R). This allows us to obtain a solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \{0, 1\}^V \times \{0, 1\}^{E_\theta}$ that represents clusters.
2. **Frequency assignment step:** Assigning channels to clusters and potentially turning-off devices if needed. We define this frequency assignment

Algorithm 4 Procedure **GreedyHeuristic1 (GH1)**

```
1: procedure GREEDYHEURISTIC1( $I, J, \mathcal{C}$ )
2:    $\text{val} \leftarrow \rho \sum_{i \in I} (d_i + u_i)$ 
3:   Declare patterns as an array indexed by  $\mathcal{C}$ 
4:   for  $c$  in  $\mathcal{C}$  do ▷ Loop over the channels.
5:     ChannelEnv  $\leftarrow \emptyset$ 
6:      $S, \text{ChannelEnv} \leftarrow \text{ClusterSelection}(I, J, \text{ChannelEnv})$ 
7:     while  $S < 0$  do
8:        $\text{val} \leftarrow \text{val} + S$ 
9:       patterns[ $c$ ]  $\leftarrow \text{ChannelEnv}$ 
10:       $I \leftarrow I \setminus \text{clients}(\text{ChannelEnv})$ 
11:       $J \leftarrow J \setminus \text{APs}(\text{ChannelEnv})$ 
12:       $S, \text{ChannelEnv} \leftarrow \text{ClusterSelection}(I, J, \text{ChannelEnv})$ 
13:    end while
14:  end for
15:  return val, patterns
16: end procedure
```

Algorithm 5 Procedure **GreedyHeuristic2 (GH2)**

```
1: procedure GREEDYHEURISTIC2( $I, J, \mathcal{C}, \text{triesNb}$ )
2:   TabuList  $\leftarrow []$ 
3:   for  $i = 1, \dots, \text{triesNb}$  do
4:     Run GreedyHeuristic( $I, J, \mathcal{C}$ ), but forbidding the APs in the TabuList for the
     first AP selection of the first channel.
5:      $A \leftarrow$  the first selected AP in the computed solution
6:     Append TabuList with  $A$ 
7:   end for
8:   return Best encountered solution
9: end procedure
```

problem as the initial problem (P) with the $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ -based additional constraints:

$$\begin{aligned} \forall i \in V, x_i &\leq \bar{x}_i, \\ \forall (i, j) \in E_\theta, y_{ij} &\leq \bar{y}_{ij}. \end{aligned}$$

These constraints reduce the combinatorial nature of the problem by specifying that only devices and connections existing in $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ may be used in the final deployment. Introducing the set of allowable edges $E(\bar{\mathbf{y}}) = \{(i, j) \in E_\theta : \bar{y}_{ij} = 1\}$, this frequency assignment problem ($P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}}$) can be written in a

Algorithm 6 Procedure **RelaxHeuristic1 (RH1)**

- 1: **procedure** RELAXHEURISTIC1(I, J, \mathcal{C})
 - 2: Solve interference-free relaxation (R).
 - 3: Let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be an optimal solution of the relaxation.
 - 4: Solve the frequency assignment problem $P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}}$.
 - 5: Let **sol** be the optimal solution found and **val** be its value.
 - 6: **return** **val**, **sol**
 - 7: **end procedure**
-

compact way:

$$\begin{aligned}
\min \quad & \sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i) \\
\text{s.t.} \quad & \forall i \in V, x_i \leq \bar{x}_i \\
& \forall i \in V, \sum_{c \in \mathcal{C}} w_{ic} = x_i \\
& \forall c \in \mathcal{C}, \forall (i, j) \in E(\bar{\mathbf{y}}), w_{ic} \leq w_{jc} \\
& \forall (i, j) \in E(\bar{\mathbf{y}}), \forall c \in \mathcal{C}, \theta + \sum_{\substack{(k,l) \in E(\bar{\mathbf{y}}) \\ l \neq j}} p_{ki} w_{kc} + \sum_{\substack{k \in J \\ k \neq j}} p_{ki} w_{kc} \leq K_{ij}^d + M(1 - w_{ic}) \\
& \forall (i, j) \in E(\bar{\mathbf{y}}), \forall c \in \mathcal{C}, \theta + \sum_{\substack{(k,l) \in E(\bar{\mathbf{y}}) \\ l \neq j}} p_{kj} w_{kc} + \sum_{\substack{k \in J \\ k \neq j}} p_{kj} w_{kc} \leq K_{ij}^u + M(1 - w_{ic}) \\
& \mathbf{x} \in \{0, 1\}^V, \mathbf{w} \in \{0, 1\}^{V \times \mathcal{C}}.
\end{aligned}$$

This optimization problem has a number of variables and constraints linear in $|\mathcal{C}| \times |V|$. Indeed we have $|E(\bar{\mathbf{y}})| \leq V$. We underline that problem $(P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}})$ does not only aim at assigning frequencies to each device of the relaxation solution $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$, but also at turning off devices to get a feasible solution of (P) . From any (\mathbf{x}, \mathbf{w}) solution of $(P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}})$, one can deduce a solution (\mathbf{x}, \mathbf{y}) of (P) by setting $y_{ij}^c = \bar{y}_{ij} w_{ic}$. This property is used in the following algorithms: for a matter of simplicity, we consider (\mathbf{x}, \mathbf{w}) and (\mathbf{x}, \mathbf{y}) as being equivalent, if the definition of $\bar{\mathbf{y}}$ is unambiguous.

With those definitions, we may now introduce the pseudocode of this first relaxation-based heuristic (Algorithm 6).

5.2.2 Second relaxation-based heuristic

We introduce an iterative variant of the relaxation-based heuristic consisting in repeating positioning step and frequency assignment step alternately. The APs and clients positioning step is modified to benefit from the feedback of the frequency assignment step. We assume we have a threshold value $\tau \in \mathbb{R}_+$ and a non-negative vector $\phi \in \mathbb{R}_+^V$ and we are looking for a solution (\mathbf{x}, \mathbf{y}) of the relaxation problem (R) whose objective value is lower than τ and with minimal score

$$\sum_{\substack{i \in V \\ x_i = 1}} \phi_i \quad \sum_{\substack{j \in V \\ \neg \exists H \in \mathcal{F}(\mathbf{y}) \ i, j \in H}} p_{ji} \tag{13}$$

where $\mathcal{F}(\mathbf{y})$ denotes the set of clusters associated to deployment \mathbf{y} . We recall that a cluster is a set gathering an AP and all clients connected to it. Hence, the set $\mathcal{F}(\mathbf{y})$ is formally defined as

$$\mathcal{F}(\mathbf{y}) = \{ \{j\} \cup \{i \in I | y_{ij} = 1\} \mid j \in J, x_j = 1 \}. \tag{14}$$

The objective function (13) is chosen to come up with solution of the relaxation that avoid the overlapping of clusters. The problem of looking such a solution can be formulated as a BLP problem:

$$\begin{aligned}
\min \quad & \sum_{i \in V} \phi_i n_i \\
s.t. \quad & \sum_{j \in J} c_j x_j + \rho \sum_{i \in I} (d_i + u_i)(1 - x_i) \leq \tau \\
& \forall i \in I, \quad \sum_{j \in J | (i,j) \in E_\theta} y_{ij} = x_i \\
& \forall j \in J, \quad \sum_{i \in I | (i,j) \in E_\theta} (d_i + u_i) y_{ij} \leq \gamma x_j \\
& \forall (i,j) \in E_\theta, y_{ij} \leq x_j \\
& \forall (i,j) \in E_\theta, y_{ij} \leq x_i \\
& \forall i \in J, \quad \sum_{j \in J, j \neq i} p_{ji} x_j + \sum_{j \in I} p_{ji} (x_j - y_{ji}) \leq n_i + M(1 - x_i) \\
& \forall i \in I, \quad \sum_{j \in J} p_{ji} (x_j - y_{ij}) + \sum_{(j,k) \in E_\theta} p_{ji} (y_{jk} - y_{ik}) \leq n_i + M(1 - x_i) \\
& \mathbf{x} \in \{0, 1\}^V, \mathbf{y} \in \{0, 1\}^{E_\theta}, \mathbf{n} \in \mathbb{R}_+^V
\end{aligned} \tag{R_{\tau, \phi}}$$

In this BLP model, the variable n_i represents the sum of the signal powers received by i and emitted by devices j that are not in the same cluster as i . Now that problem $(R_{\tau, \phi})$ is defined, we are able to present the second relaxation-based heuristic (Algorithm 7), that depends on two parameters: a parameter `nb_it` denoting a maximal number of iterations and a multiplicative factor $r > 1$ to increase the penalty associated to a node that is turned off during the frequency assignment step. During this process, the parameter τ stores the current target

Algorithm 7 Procedure **RelaxHeuristic2 (RH2)**

- 1: **procedure** RELAXHEURISTIC2($I, J, \mathcal{C}, \text{nb_it}, r$)
 - 2: Solve interference-free (and channel free) relaxation (R) .
 - 3: Let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be the optimal solution found and τ be its value.
 - 4: Solve the frequency assignment problem $P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}}$ associated with $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be its optimal solution found and v be its value.
 - 5: $\text{Gap} \leftarrow v - \tau$, $\delta \leftarrow \text{Gap}/\text{nb_it}$, $\phi \leftarrow \mathbf{1}$
 - 6: **if** $\text{Gap} = 0$ **then**
 - 7: **return** v , $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$
 - 8: **end if**
 - 9: **for** $i = 1, \dots, \text{nb_it}$ **do**
 - 10: Solve $(R_{\tau, \phi})$ and let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be the optimal solution found.
 - 11: Solve the frequency assignment problem $P_{\bar{\mathbf{x}}, \bar{\mathbf{y}}}$ associated with $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ and let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be its optimal solution found and v be its value.
 - 12: **if** $v \leq \tau$ **then return** v , $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$
 - 13: **else**
 - 14: $\tau \leftarrow \tau + \delta$
 - 15: For any $i \in V$ such that $\bar{x}_i = 1$ and $\tilde{x}_i = 0$, $\phi_i \leftarrow r\phi_i$
 - 16: **end if**
 - 17: **end for**
 - 18: **return** Best encountered solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ and its value.
 - 19: **end procedure**
-

for a solution value.

6 Computational experiments

This last section is dedicated to computational experiments that we led to assess the proposed methodology. We present the numerical setup for the simulation and the optimization parts, as well as the performance of both algorithmic steps.

6.1 Instances generation

6.1.1 Map generation and simulation

The first experimental step was to design building plans. The solution we chose to produce examples for the moment was to design building maps with a graphical raster editor. Table 7 introduces the physical coefficients for the simulation, that we used for all the maps. We produced 6 different types of floors. The dimensions

Parameter	f_c	h	α_{air}	α_{mur}	n_{air}	n_{mur}	β	P_{AP}	P_{client}
Maps 1,2 and 4	2.4 GHz	3 cm	0 s/m ²	15 s/m ²	1	4	1	2 W	0.1 W
Maps 3,5 and 6	2.4 GHz	3 cm	0 s/m ²	10 s/m ²	1	3	1	2 W	0.1 W

Table 7: Simulation parameters

of such 2D maps are detailed in the Sect. 6.2. For each floor, we produce a one-level, a two-level and a three-level building based on this (duplicated) floor: we come up with 18 different buildings. In order to calculate the field generated by devices on another floor, we use the field projection method (see Sect. 2.4) with an attenuation gain of -15 dB per floor. For each building, the client and APs positions (2D coordinates and floor) are chosen at random. The choice of the discretization step h is consistent with the common knowledge in simulation, of using a step smaller than a quarter wavelength.

6.1.2 Parameters of the optimization problems

We produced instances with uniform AP costs ($c_j = 10$) and uniform clients uplink and downlink normalised demanded data rates ($u_i = d_i = 0.5$). The maximum data rate that a AP can process is set to $\gamma = 8$. For each client, it corresponds to a balanced situation between uplink and downlink flow with a total data flow equals to the bandwidth. According to Shannon's law (see Sect. 1.3) and using notations introduced in (2), the parameters K_{ij}^u and K_{ij}^d are taken in accordance to the normalised data rates and the signal strengths:

$$\begin{aligned} K_{ij}^u &= N_{\max}(u_i, 1, p_{ij}) = \frac{p_{ij}}{2^{u_i-1}}, \\ K_{ij}^d &= N_{\max}(d_i, 1, p_{ji}) = \frac{p_{ji}}{2^{d_i-1}}. \end{aligned} \quad (15)$$

Each of the 3 sets of parameters presented in Table 8 was tested for the 18 building configurations, giving us a total of 54 cases. We now discuss the implementation and the results of the simulations in Sect. 6.2 and then the optimization algorithms in Sect. 6.3.

6.2 Simulation experiments

6.2.1 Implementation

The numerical experiments presented here were ran on a computer with following characteristics:

Set of parameters	θ	ρ
Set 0	0.0001	100
Set 1	0.0001	10
Set 2	0.001	100

Table 8: Sets of parameters

- CentOS Linux 7 operating system;
- 32 processors Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz;
- 64 GB of RAM.

The LU factorization is computed with a variant of the Gilbert and Peierls algorithm for sparse gaussian elimination, implemented in the *SuperLU* library [63]. Based on this LU factorization, the linear systems are solved with the same library. This library is accessed through the Scipy wrapper [71] for Python.

6.2.2 Results

In the following table, the columns "LU time" and "Average solution time" correspond respectively to the LU factorization time and the average system solution time for one source and for a given floor. The average time for the projection step on an other floor (see Sect. 2.4) is presented in the column "Average projection time".

Floor type	Ground surface	N_x	N_y	LU time	Average solution time	Average projection time
Map 1	153 m ²	523	325	106.5 s	0.13 s	0.04 s
Map 2	293 m ²	1000	325	177.6 s	0.25 s	0.08 s
Map 3	472 m ²	898	584	240.4 s	0.41 s	0.13 s
Map 4	683 m ²	1200	632	198.6 s	0.59 s	0.18 s
Map 5	593 m ²	1067	618	92.2 s	0.47 s	0.15 s
Map 6	817 m ²	1230	738	102.1 s	0.72 s	0.21 s

Table 9: Simulation results

The calculation times displayed here correspond to sequential computations: in the implementation of the SuperLU library that we used, the multiple processors of the machine are not exploited. A direct way to exploit multithreading would be, given the LU decomposition, to run parallel processes to compute propagation for different sources. The largest building treated here (Map 6 duplicated on three floors) corresponds to an office building of intermediate size.

6.3 Optimization experiments

6.3.1 Implementation

All the optimization algorithms have been implemented in the C++ programming language and ran on the same machine that the one used for the simulation experiments. We used a commercial MILP solver, namely IBM ILOG CPLEX 12.8 [72] that we called through the Concert Technology API for C++. In practise, we did not use the "big-M" constant in MILP problems (P), ($P_{\mathbf{x},\bar{y}}$) and ($R_{\tau,\phi}$) but

a dedicated CPLEX feature to handle such constraints. At most 16 processors were allocated to CPLEX, and we set a time limit of 600 seconds.

6.3.2 CPLEX performance

CPLEX was able to solve 72% of the instances to optimality within the time frame of 600s per instance. For the instances that CPLEX solved within 600s, the Figure 1 shows an exponential relationship between the solution time and the total number of clients and candidates. The computation time and the lower and upper bounds for each instance are given in the Appendix. Of the 15 cases that

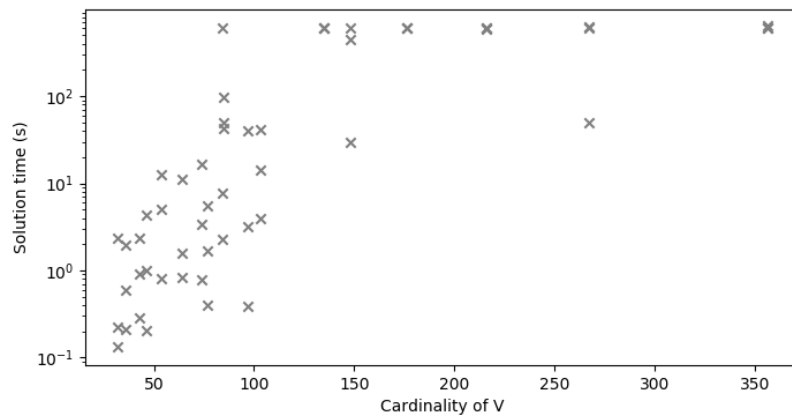


Figure 1: Performance of CPLEX's branch-and-bound solver: solution time

were not solved to optimality by CPLEX, the optimality gap is greater than 20% in 8 cases. Figure 2 illustrates that the largest instances show a high optimality gap of more than 80% for 4 cases. These very large gaps illustrate the difficulty of the OWLD problem, even for relatively small instances; therefore it calls for heuristic algorithms.

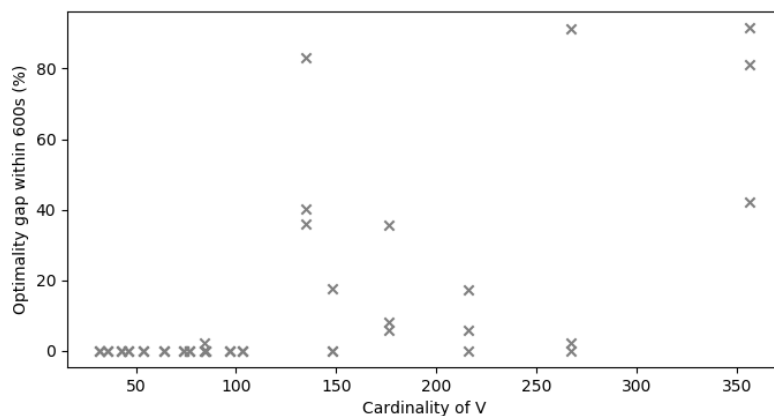


Figure 2: Performance of CPLEX's branch-and-bound solver: optimality gap within 600 sec

6.3.3 Heuristics performance

The performance of the heuristic algorithms are presented in comparison with the time and solution results of CPLEX branch-and-bound algorithm. Detailed results for each instance are presented in the Appendix. Figure 3 is a performance profile, plotting in the x-axis the CPU time ratio compared to CPLEX's CPU time and in the y-axis the % of instances for which the heuristic does better than this ratio. For example, GH1's CPU time is 1000 times shorter than CPLEX's for 55% of the instances. According to this same figure, the two greedy heuristics are significantly faster than other algorithms. Both relaxation-based heuristics have generally shorter computation times than CPLEX, but in a significant part of the cases they are longer. The parameters used for the iterative versions of

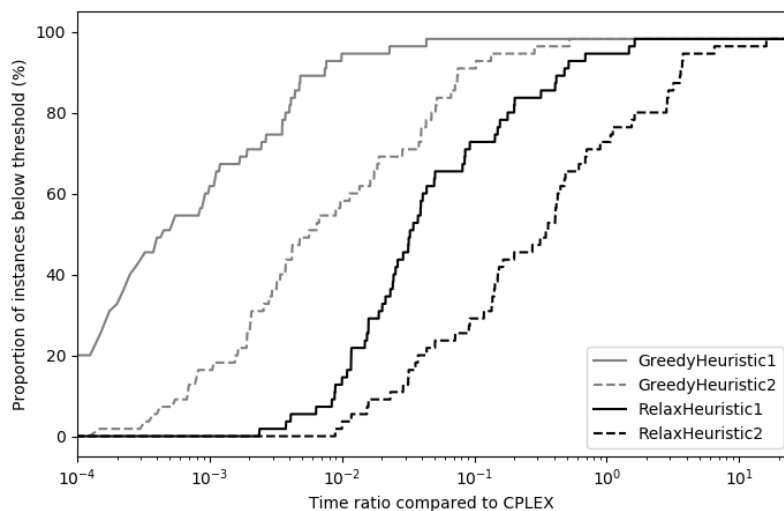


Figure 3: Performance of heuristic algorithms: solution time compared to CPLEX

greedy and relaxation-based heuristics are the following:

- (GH2) $\text{triesNb} = 10$
- (RH2) $r = 1$, if $|V| \leq 70$, $\text{nb_it} = 15$ else $\text{nb_it} = 5$.

For both type of heuristics, greedy (GH) and relaxation-based (RH), it appears that the iterative version of the heuristic has a longer computing time but finds much better solutions: the performance profile is improved. Globally, all heuristics present a certain lack of robustness in that extent that, in 10% to 20% of the instances depending on the heuristic, they failed to find a solution under a +100% quality gap compared to CPLEX's solution. The iterative relaxation-based heuristic presents the best rate (63%) of instances where the solution found is as good as the solution found by CPLEX or better. In this perspective, it is the most robust heuristic. However, the multistart greedy heuristic clearly outperforms all other heuristic algorithms in term of solution quality for big instances, as Table 10 shows. It greatly outperforms the CPLEX solution for 3 of these 6 instances. To that extent, the multistart greedy heuristic is more complementary to the exact branch-and-bound algorithm than the iterative relaxation-based heuristic.

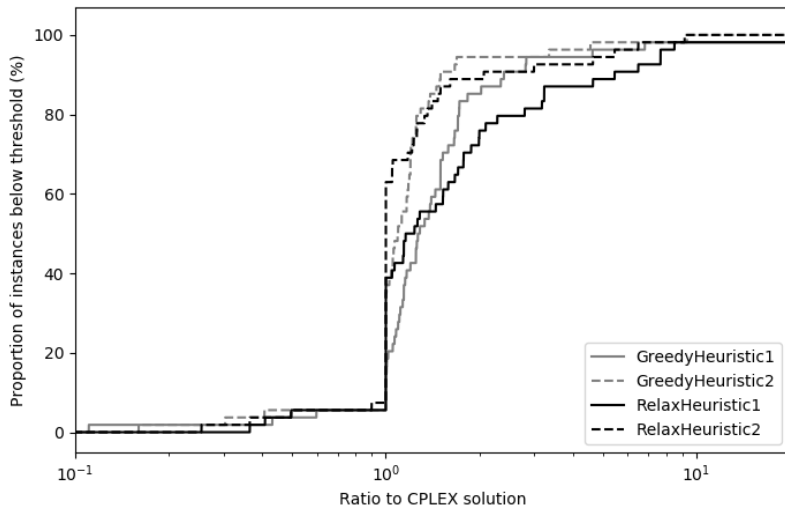


Figure 4: Performance of heuristic algorithms: solution quality compared to CPLEX

Instance name	$ I $	$ J $	Solution Value				Lower bound (B&B)	
			B&B	GH1	GH2	RH1		RH2
Instance_MAP6_1_1	155	112	440	490	480	500	440	430
Instance_MAP6_1_2	155	112	2570	3710	3060	4390	3500	2570
Instance_MAP6_1_3	155	112	7690	850	850	3140	1960	676.25
Instance_MAP6_2_1	175	181	1810	780	550	660	660	342.50
Instance_MAP6_2_2	175	181	4070	5160	4230	6790	5420	2350
Instance_MAP6_2_3	175	181	14490	8660	5890	7180	7180	1221.25

Table 10: Performance of B&B and heuristics on the six largest instances

7 Conclusion

We proposed a protocol to optimize the deployment of a WLAN, taking full account of the building’s architecture, which we tested on mid-size buildings with a floor area of up to 1000m² and several floors. The strength of our approach is a sophisticated but fast to code simulator that is based on the physical equations of radio wave propagation. This simulator allows us to calculate the electromagnetic field produced by any source in the building. We then introduce a stylized network deployment optimization problem, which consists of optimizing the positioning and frequency assignment of the APs to provide WiFi access to customers whose positions are given. This problem, which is formulated as a BLP, has the particularity of fully exploiting the data produced by the radio propagation simulator. We show that this problem is strongly NP-difficult, and the numerical experiments that we have carried out with a standard MILP solver implementation have confirmed the difficulty of the problem. This commercial solver was typically able to solve instances within ten minutes, if the combined number of clients and APs is about 100. Within this time limit, the optimality gap could exceed 80% for the largest instances encountered, with more than 300 nodes. This is the reason why we also proposed several heuristic algorithms: a greedy heuristic and its iterative variant; and another heuristic based on a natural relaxation

of the original problem, and its iterative variant. Although not the most robust heuristic among the four, the iterative variant of the greedy heuristic turned out to be the most attractive as it allowed to find significantly better solutions than the solver for some of the largest instances. From this perspective, it is the most complementary to the exact approach.

In future work we will be able to test our simulator for even larger buildings, albeit at the cost of longer computation times. Future work will also consist of testing the different optimization algorithms, exact or heuristic, for even larger instances so as to get closer to the thousand, regarding the cumulated number of clients and APs. It is likely that such cases would clearly demonstrate the relevance of using heuristic algorithms. Another future research line is to implement an exact approach that can better scale, as a column-generation algorithm for instance. Finally, we also plan to propose a stochastic or robust variant of this optimization problem to take into account several scenarios concerning the use of the network or the presence of moving obstacles in the building.

Acknowledgements

This work was mainly funded by the Cisco Foundation Grant URP n. 594955. Two of the authors (CD, LL) are also partly supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement n. 764759 "MINOA".

References

- [1] P. Calégari, F. Guidec, P. Kuonen and B. Chamaret and S. Ubéda and S. Josselin and D. Wagner and M. Pizarosso, *Radio network planning with combinatorial optimization algorithms*. ACTS Mobile Telecommunications Summit, 1996, pp. 707-713.
- [2] I. Akyildiz, W. Su and Y. Sankarasubramaniam and E. Cayirci, *Wireless sensor networks: a survey*, *Computer networks* **38** (2002), 393–422.
- [3] B. Das and V. Bharghavan, *Routing in ad-hoc networks using minimum connected dominating sets*, *International Conference on Communications, IEEE*, 1999, pp. 376–380.
- [4] G.-H. Lin and G. Xue, *Steiner tree problem with minimum number of Steiner points and bounded edge-length*, *Information Processing Letters* **69** (1999), 53–57.
- [5] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang and G. Xue, *Approximations for Steiner trees with minimum number of Steiner points*, *Journal of Global Optimization* **18** (2000), 17–33.
- [6] M. Min, H. Du, X. Jia, C. Huang, S. Huang and W. Wu, *Improving construction for connected dominating set with Steiner tree in wireless sensor networks*, *Journal of Global Optimization* **35** (2006), 111–119.
- [7] M. Rebai, H. Snoussi, F. Hnaïen and L. Khoukhi, *Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks*, *Computers & Operations Research* **59** (2015), 11–21.

- [8] S. Elloumi, O. Hudry, E. Marie, A. Plateau and S. Rovedakis. *Optimization of wireless sensor networks deployment with coverage and connectivity constraints*. 4th International Conference on Control, Decision and Information Technologies, IEEE, 2017, pp. 336–341.
- [9] C. Bentz, M.-C. Costa and A. Hertz. On the edge capacitated Steiner tree problem, Working paper, 2019. <https://hal.archives-ouvertes.fr/hal-01465403/document>.
- [10] G. Di Caro and E. Flushing, *Optimal relay node placement for throughput enhancement in wireless sensor networks*, 50th FITCE Congress ICT: Bridging an Ever Shifting Digital Divide, IEEE, 2011, pp. 1–6.
- [11] Y. Hou, Y. Shi, H. Sherali and S. Midkiff, *Prolonging sensor network lifetime with energy provisioning and relay node placement*, Second Annual Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE, 2005, pp. 295–304.
- [12] G. Wang, L. Huang, H. Xu and J. Li, *Relay node placement for maximizing network lifetime in wireless sensor networks*, 4th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2008, pp. 1–5.
- [13] T. Rault, A. Bouabdallah and Y. Challal, *Energy efficiency in wireless sensor networks: A top-down survey*, Computer Networks **67** (2014), 104–122.
- [14] K. Aardal, S. van Hoesel, A. Koster, C. Mannino and A. Sassano, *Models and solution techniques for frequency assignment problems*, Annals of Operations Research **153** (2007), 79–129.
- [15] E. Amaldi, A. Capone, F. Malucelli and C. Mannino, “*Optimization problems and models for planning cellular networks*”, *Handbook of Optimization in Telecommunications*, Springer, Boston, 2006, pp. 917–939.
- [16] C. Lee and H. Kang, *Cell planning with capacity expansion in mobile communications: A tabu search approach*, Transactions on Vehicular Technology **49** (2000), 1678–1691, .
- [17] R. Akl, M. Hegde, M. Naraghi-Pour and P. Min, *Multicell CDMA network design*, Transactions on Vehicular Technology **50** (2001), 711–722.
- [18] R. Mathar and M. Schmeink, *Optimal base station positioning and channel assignment for 3G mobile networks by integer programming*, Annals of Operations Research **107** (2001), 225–236.
- [19] D. Catrein, L. Imhof and R. Mathar, *Power control, capacity, and duality of uplink and downlink in cellular CDMA systems*, Transactions on Communications **52**,(2004), 1777–1785.
- [20] E. Amaldi, A. Capone, F. Malucelli and F. Signori, *UMTS radio planning: Optimizing base station configuration*, Proceedings of 56th Vehicular Technology Conference, volume 2, IEEE, 2002, pp. 768–772.
- [21] E. Amaldi, A. Capone and F. Malucelli, *Planning UMTS base station location: Optimization models with power control and algorithms*, Transactions on Wireless Communications **5** (2003), 939–952.
- [22] E. Amaldi, P. Belotti, A. Capone and F. Malucelli, *Optimizing base station location and configuration in UMTS networks*, Annals of Operations Research, **146** (2006), 135–151.

- [23] Y. Lee, K. Kim and Y. Choi, *Optimization of AP placement and channel assignment in wireless LANs*, 27th Annual Conference on Local Computer Networks, IEEE, 2002, pp. 831–836.
- [24] A. Gondran, O. Baala, A. Caminada and H. Mabed, *Joint optimization of access point placement and frequency assignment in WLAN*, 3rd International Conference in Central Asia on Internet, IEEE, 200, pp. 1–5.
- [25] A. Eisenblatter, H.-F. Geerdes and I. Siomina, *Integrated access point placement and channel assignment for wireless LANs in an indoor office environment*. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, IEEE, 2007, pp. 1–10.
- [26] A. Farsi, N. Achir and K. Boussetta, *WLAN planning: Separate and joint optimization of both access point placement and channel assignment*, Annals of Telecommunications **70** (2015), 263–274.
- [27] E. Ben Hamida, G. Chelius and J.-M. Gorce, *Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation*, Simulation, **85** (2009), 574–588.
- [28] S. Seidel and T. Rappaport, *914 MHz path loss prediction models for indoor wireless communications in multifloored buildings*, Transactions on Antennas and Propagation **40** (1992), 207–217.
- [29] S. Seidel and T. Rappaport, *Site-specific propagation prediction for wireless in-building personal communication system design*, Transactions on Vehicular Technology, **43** (1994), 879–891.
- [30] K.-W. Cheung, J.-M. Sau and R. Murch, *A new empirical model for indoor propagation prediction*, Transactions on Vehicular Technology **47** (1998), 996–1001.
- [31] T. Rappaport *et al*, *Wireless Communications: Principles and Practice*, IEEE Press, Upper Saddle River, 1996.
- [32] R. Valenzuela, S. Fortune and J. Ling, *Indoor propagation prediction accuracy and speed versus number of reflections in image-based 3D ray-tracing*, 48th Vehicular Technology Conference, vol. 1., IEEE, 1998, pp. 539–543.
- [33] S. Fortune, *Efficient algorithms for prediction of indoor radio propagation*, 48th IEEE Vehicular Technology Conference, vol. 1. IEEE, 1998, pp. 572–576.
- [34] C.-F. Yang, B.-C. Wu and C.-J. Ko, *A ray-tracing method for modeling indoor wave propagation and penetration*, Transactions on Antennas and Propagation **46** (1998), 907–919.
- [35] M. Hassan-Ali and K. Pahlavan, *A new statistical model for site-specific indoor radio propagation prediction based on geometric optics and geometric probability*, Transactions on Wireless Communications **1** (2002), 112–124.
- [36] G. Athanasiadou and A. Nix, *Investigation into the sensitivity of the power predictions of a microcellular ray tracing propagation model*, Transactions on Vehicular Technology **49** (2000), 1140–1151.
- [37] F. Agelet, A. Formella, J.-M. Rabanos, F. De Vicente and F. Fontan, *Efficient ray-tracing acceleration techniques for radio propagation modeling*, Transactions on Vehicular Technology **49** (2000), 2089–2104.
- [38] Z. Yun, Z. Zhang and M. Iskander, *A ray-tracing method based on the triangular grid approach and application to propagation prediction in urban environments*, Transactions on Antennas and Propagation **50** (2002), 750–758.

- [39] L. Talbi and G. Delisle, *FDTD characterization of the Indoor Propagation*, Journal of Electromagnetic Waves and Applications **10** (1996), 243–247.
- [40] J. Lee and A. Lai, *FDTD analysis of Indoor Radio Propagation*, Antennas and Propagation Society International Symposium, vol. 3, IEEE, 1998, pp. 1664–1667.
- [41] B. Chopard, P. Luthi and J.-F. Wagen, *Lattice Boltzmann method for wave propagation in urban microcells*, Proceedings - Microwaves, Antennas and Propagation, vol. 144, IEEE, 1997, pp. 251–255.
- [42] J.-M. Gorce, K. Jaffres-Runser and G. De La Roche, *Deterministic approach for fast simulations of indoor radio wave propagation*, Transactions on Antennas and Propagation **55** (2007), 938–948.
- [43] G. De La Roche, Radio wave propagation simulation in multipath environments for the study of wireless networks, Ph.D. thesis, INSA de Lyon, 2007.
- [44] K. Jaffres-Runser and J.-M. Gorce, *Assessment of a new indoor propagation prediction method based on a multi-resolution algorithm*, 61st Vehicular Technology Conference, IEEE, 2005.
- [45] A. Yourtchenko, Personal communications (2018).
- [46] J. Wong, M. Neve and K. Sowerby. *Uplink and downlink SIR analysis for base station placement*, Semiannual Vehicular Technology Conference, vol. 1, IEEE, 2003, pp. 112–116.
- [47] A. Bahri and S. Chamberland, *On the wireless local area network design problem with performance guarantees*, Computer Networks **48** (2005), 856–866.
- [48] P. Das, N. Chakraborty and S. Allayear, *Optimal coverage of wireless sensor network using termite colony optimization algorithm*, International Conference on Electrical Engineering and Information Communication Technology, IEEE, 2015, pp. 1–6.
- [49] F. Agelet, A. Varela, L. Alvarez-Vazquez and A. Formella, *Optimization methods for optimal transmitter locations in a mobile wireless system*, In Biennial Conference on Electromagnetic Field Computation, IEEE, 2000, pp. 441–452.
- [50] G. Mateus, A. Loureiro and R. Rodrigues, *Optimal network design for wireless local area network*, Annals of Operations Research **106** (2001), 331–345.
- [51] C. Prommak, J. Kabara, D. Tipper and C. Charnsripinyo, *Next generation wireless LAN system design*. In Military Communications Conference, vol. 1, IEEE, 2002, pp. 473–477.
- [52] Z. Ji, T. Sarkar and B.-H. Li, *Methods for optimizing the location of base stations for indoor wireless communications*, Transactions on Antennas and Propagation **50** (2002), 1481–1483.
- [53] K. Jaffres-Runser, Methodologies for Wireless LAN Planning. Ph.D. thesis, INSA de Lyon, 2005.
- [54] S. Zirazi, P. Canalda and H. Mabed and F. Spies, *Wi-Fi access point placement within stand-alone, hybrid and combined wireless positioning systems*, 4th International Conference on Communications and Electronics, IEEE, 2012, pp. 279–284.
- [55] Z. Zhang, X. Di, J. Tian and Z. Zhu, *A multi-objective WLAN planning method*, International Conference on Information Networking, 2017, pp. 86–91.

- [56] E. Amaldi, S. Bosio, F. Malucelli and D. Yuan, *Solving Nonlinear Covering Problems Arising in WLAN Design*, Operations Research **59** (2011), 173–187.
- [57] H. Sherali, C. Pendyala and T. Rappaport, *Optimal location of transmitters for micro-cellular radio communication system design*, Journal on selected areas in communications **14** (1996), 662–673.
- [58] H. Eldeeb, M. Arafa and M. Saidahmed, *Optimal placement of access points for indoor positioning using a genetic algorithm*, 12th International Conference on Computer Engineering and Systems, IEEE, 2017, pp. 306–313.
- [59] X. Du and K. Yang, *A map-assisted WiFi AP placement algorithm enabling mobile devices indoor positioning*, IEEE Systems Journal **11** (2016), 1467–1475.
- [60] N. Li, J. Chen, Y. Yuan, X. Tian, Y. Han and M. Xia, *A Wi-Fi indoor localization strategy using particle swarm optimization based artificial neural networks*, International Journal of Distributed Sensor Networks **12** (2016).
- [61] T. Fruhwirth and P. Brisset, *Placing base stations in wireless indoor communication networks*, IEEE Intelligent Systems and Their Applications, **15** (2000), 49–53.
- [62] J. He, A. Verstak, L. Watson, C. Stinson, N. Ramakrishnan, C. Shaffer, T. Rappaport, C. Anderson, K. Bae and J. Jiang *et al*, *Globally optimal transmitter placement for indoor wireless communication systems*, Transactions on Wireless Communication **3** (2004), 1906–1911.
- [63] J. Demmel, S. Eisenstat, J. Gilbert, X. Li and J. Liu, *A supernodal approach to sparse partial pivoting*, SIAM Journal on Matrix Analysis and Applications **20** (1999), 720–755.
- [64] D. Vassiss, G. Kormentzas, A. Rouskas and I. Maglogiannis, *The IEEE 802.11 g standard for high data rate WLANs*, IEEE Network **19** (2005), 21–26.
- [65] J.-M. Gorce, K. Jaffres-Runser and G. De La Roche, *The Adaptive Multi-Resolution Frequency-Domain ParFlow (MR-FDPPF) Method for Indoor Radio Wave Propagation Simulation*, Technical Report, INRIA, 2005.
- [66] T. Davis, *Direct methods for sparse linear systems*, vol. 2, SIAM, 2006.
- [67] I. Duff, A. Erisman and J. Reid, *Direct methods for sparse matrices*, Oxford University Press, 2017.
- [68] A. Hoffman, M. Martin and D. Rose, *Complexity Bounds for Regular Finite Difference and Finite Element Grids*, SIAM Journal on Numerical Analysis **10** (1973), 364–369.
- [69] A. George, *Nested Dissection of a Regular Finite Element Mesh*, SIAM Journal on Numerical Analysis **10** (1973), 345–363.
- [70] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, 2019.
- [71] E. Jones, T. Oliphant and P. Peterson *et al*, *Scipy: Open source scientific tools for Python*, 2016.
- [72] IBM ILOG, *V12.8: Users manual for CPLEX*, International Business Machines Corporation, 2017.

Appendix

Instance name	$ I $	$ J $	Solution Value	Lower Bound	Optimality Gap (in %)	Computation Time (in s)
Instance_MAP1_0_1	13	19	30	30	0	0.1
Instance_MAP1_0_2	13	19	50	50	0	0.2
Instance_MAP1_0_3	13	19	30	30	0	2.3
Instance_MAP1_1_1	23	13	20	20	0	0.6
Instance_MAP1_1_2	23	13	50	50	0	0.2
Instance_MAP1_1_3	23	13	20	20	0	2.0
Instance_MAP1_2_1	24	30	40	40	0	5.0
Instance_MAP1_2_2	24	30	190	190	0	0.8
Instance_MAP1_2_3	24	30	40	40	0	12.7
Instance_MAP2_0_1	24	22	40	40	0	1.0
Instance_MAP2_0_2	24	22	310	310	0	0.2
Instance_MAP2_0_3	24	22	30	30	0	4.3
Instance_MAP2_1_1	35	39	80	80	0	3.4
Instance_MAP2_1_2	35	39	490	490	0	0.8
Instance_MAP2_1_3	35	39	60	60	0	16.8
Instance_MAP2_2_1	56	47	80	80	0	14.4
Instance_MAP2_2_2	56	47	510	510	0	4.0
Instance_MAP2_2_3	56	47	60	60	0	41.4
Instance_MAP3_0_1	25	18	30	30	0	0.9
Instance_MAP3_0_2	25	18	170	170	0	0.3
Instance_MAP3_0_3	25	18	30	30	0	2.3
Instance_MAP3_1_1	40	45	60	60	0	42.6
Instance_MAP3_1_2	40	45	730	730	0	49.7
Instance_MAP3_1_3	40	45	60	60	0	98.9
Instance_MAP3_2_1	62	73	180	115.0	36.1	603.4
Instance_MAP3_2_2	62	73	1770	1055.7	40.4	600.9
Instance_MAP3_2_3	62	73	620	105.0	83.1	606.1
Instance_MAP4_0_1	38	26	50	50	0	1.6
Instance_MAP4_0_2	38	26	190	190	0	0.8
Instance_MAP4_0_3	38	26	40	40	0	11.3
Instance_MAP4_1_1	36	41	150	150	0	1.7
Instance_MAP4_1_2	36	41	1360	1360	0	0.4
Instance_MAP4_1_3	36	41	380	380	0	5.5
Instance_MAP4_2_1	41	56	210	210	0	3.2
Instance_MAP4_2_2	41	56	1890	1890	0	0.4
Instance_MAP4_2_3	41	56	100	100	0	40.7
Instance_MAP5_0_1	39	45	130	130	0	7.8
Instance_MAP5_0_2	39	45	630	630	0	2.3
Instance_MAP5_0_3	39	45	570	557.5	2.2	600.6
Instance_MAP5_1_1	70	78	150	123.8	17.5	602.9
Instance_MAP5_1_2	70	78	440	440	0	29.6
Instance_MAP5_1_3	70	78	110	110	0	448.0
Instance_MAP5_2_1	115	101	210	197.5	6.0	608.5
Instance_MAP5_2_2	115	101	550	455.0	17.3	603.3
Instance_MAP5_2_3	115	101	170	170	0	597.5
Instance_MAP6_0_1	84	92	170	156.3	8.1	604.6
Instance_MAP6_0_2	84	92	1740	1640	5.7	601.2
Instance_MAP6_0_3	84	92	340	218.8	35.7	608.4
Instance_MAP6_1_1	155	112	440	430	2.3	612.3
Instance_MAP6_1_2	155	112	2570	2570	0	49.5
Instance_MAP6_1_3	155	112	7690	676.2	91.2	628.5
Instance_MAP6_2_1	175	181	1810	342.5	81.1	647.2
Instance_MAP6_2_2	175	181	4070	2350	42.3	613.6
Instance_MAP6_2_3	175	181	14490	1221.2	91.6	602.5

Table 11: CPLEX 12.8 branch-and-bound algorithm: optimality gap and solution time

Instance name	I	J	GH		GH	
			Solution Value	Solution Value	Time (in s)	Time (in s)
Instance_MAP1_0_1	13	19	30	30	0.01	0.01
Instance_MAP1_0_2	13	19	50	50	< 0.01	0.01
Instance_MAP1_0_3	13	19	30	30	<0.01	0.01
Instance_MAP1_1_1	23	13	30	30	< 0.01	0.01
Instance_MAP1_1_2	23	13	60	60	< 0.01	0.01
Instance_MAP1_1_3	23	13	20	20	< 0.01	0.01
Instance_MAP1_2_1	24	30	60	50	< 0.01	0.02
Instance_MAP1_2_2	24	30	210	200	< 0.01	0.03
Instance_MAP1_2_3	24	30	60	40	< 0.01	0.03
Instance_MAP2_0_1	24	22	50	50	< 0.01	0.02
Instance_MAP2_0_2	24	22	390	310	< 0.01	0.03
Instance_MAP2_0_3	24	22	40	30	< 0.01	0.02
Instance_MAP2_1_1	35	39	110	90	< 0.01	0.06
Instance_MAP2_1_2	35	39	680	590	0.01	0.08
Instance_MAP2_1_3	35	39	170	70	< 0.01	0.05
Instance_MAP2_2_1	56	47	120	90	0.01	0.13
Instance_MAP2_2_2	56	47	880	700	0.02	0.21
Instance_MAP2_2_3	56	47	70	70	0.01	0.12
Instance_MAP3_0_1	25	18	50	30	< 0.01	0.02
Instance_MAP3_0_2	25	18	270	180	< 0.01	0.02
Instance_MAP3_0_3	25	18	30	30	< 0.01	0.01
Instance_MAP3_1_1	40	45	110	100	0.01	0.08
Instance_MAP3_1_2	40	45	1480	1090	0.01	0.10
Instance_MAP3_1_3	40	45	1740	560	0.01	0.08
Instance_MAP3_2_1	62	73	310	250	0.02	0.21
Instance_MAP3_2_2	62	73	2700	2220	0.02	0.24
Instance_MAP3_2_3	62	73	2870	2080	0.02	0.20
Instance_MAP4_0_1	38	26	70	60	< 0.01	0.04
Instance_MAP4_0_2	38	26	200	200	< 0.01	0.06
Instance_MAP4_0_3	38	26	50	40	< 0.01	0.04
Instance_MAP4_1_1	36	41	170	160	0.01	0.07
Instance_MAP4_1_2	36	41	1370	1360	0.01	0.11
Instance_MAP4_1_3	36	41	410	390	0.01	0.06
Instance_MAP4_2_1	41	56	240	210	0.01	0.15
Instance_MAP4_2_2	41	56	1900	1900	0.02	0.20
Instance_MAP4_2_3	41	56	240	120	0.01	0.11
Instance_MAP5_0_1	39	45	150	130	0.01	0.08
Instance_MAP5_0_2	39	45	720	630	0.01	0.09
Instance_MAP5_0_3	39	45	580	570	0.01	0.07
Instance_MAP5_1_1	70	78	160	150	0.03	0.33
Instance_MAP5_1_2	70	78	730	640	0.03	0.40
Instance_MAP5_1_3	70	78	310	130	0.03	0.31
Instance_MAP5_2_1	115	101	230	230	0.09	1.00
Instance_MAP5_2_2	115	101	1290	930	0.10	1.25
Instance_MAP5_2_3	115	101	290	200	0.08	0.93
Instance_MAP6_0_1	84	92	290	220	0.04	0.49
Instance_MAP6_0_2	84	92	2240	2120	0.06	0.64
Instance_MAP6_0_3	84	92	2320	1550	0.04	0.41
Instance_MAP6_1_1	155	112	490	480	0.13	1.57
Instance_MAP6_1_2	155	112	3710	3060	0.20	2.52
Instance_MAP6_1_3	155	112	850	850	0.12	1.21
Instance_MAP6_2_1	175	181	780	550	0.25	2.72
Instance_MAP6_2_2	175	181	5160	4230	0.33	4.11
Instance_MAP6_2_3	175	181	8660	5890	0.18	2.26

Table 12: Greedy heuristics: solution value and computation time

Instance name	$ I $	$ J $	RH1 Solution Value	RH2 Solution Value	RH1 Time (in s)	RH2 Time (in s)
Instance_MAP1_0_1	13	19	30	30	0.1	0.1
Instance_MAP1_0_2	13	19	50	50	0.0	0.0
Instance_MAP1_0_3	13	19	30	30	0.0	0.0
Instance_MAP1_1_1	23	13	20	20	0.0	0.0
Instance_MAP1_1_2	23	13	50	50	0.0	0.0
Instance_MAP1_1_3	23	13	20	20	0.0	0.0
Instance_MAP1_2_1	24	30	40	40	0.5	0.5
Instance_MAP1_2_2	24	30	290	200	0.1	3.0
Instance_MAP1_2_3	24	30	340	40	0.1	1.2
Instance_MAP2_0_1	24	22	40	40	0.0	0.0
Instance_MAP2_0_2	24	22	650	390	0.0	0.6
Instance_MAP2_0_3	24	22	30	30	0.0	0.0
Instance_MAP2_1_1	35	39	80	80	0.1	0.1
Instance_MAP2_1_2	35	39	1120	690	0.1	2.8
Instance_MAP2_1_3	35	39	460	60	0.4	2.0
Instance_MAP2_2_1	56	47	80	80	0.1	0.1
Instance_MAP2_2_2	56	47	960	600	0.3	12.7
Instance_MAP2_2_3	56	47	460	60	1.0	6.3
Instance_MAP3_0_1	25	18	30	30	0.0	0.0
Instance_MAP3_0_2	25	18	260	170	0.0	0.1
Instance_MAP3_0_3	25	18	30	30	0.0	0.0
Instance_MAP3_1_1	40	45	120	90	0.7	20.7
Instance_MAP3_1_2	40	45	1300	920	0.6	7.0
Instance_MAP3_1_3	40	45	1750	550	0.8	27.1
Instance_MAP3_2_1	62	73	320	290	2.3	120.5
Instance_MAP3_2_2	62	73	3530	2150	15.3	81.1
Instance_MAP3_2_3	62	73	2880	2880	9.0	248.3
Instance_MAP4_0_1	38	26	50	50	0.1	0.1
Instance_MAP4_0_2	38	26	190	190	0.6	0.6
Instance_MAP4_0_3	38	26	40	40	0.6	0.6
Instance_MAP4_1_1	36	41	150	150	0.8	0.8
Instance_MAP4_1_2	36	41	1720	1360	0.6	1.5
Instance_MAP4_1_3	36	41	470	380	2.3	4.9
Instance_MAP4_2_1	41	56	240	210	1.0	4.9
Instance_MAP4_2_2	41	56	2160	1980	0.6	6.3
Instance_MAP4_2_3	41	56	280	90	2.0	6.7
Instance_MAP5_0_1	39	45	130	130	0.3	0.3
Instance_MAP5_0_2	39	45	810	630	0.5	6.5
Instance_MAP5_0_3	39	45	660	570	3.8	43.0
Instance_MAP5_1_1	70	78	160	150	1.4	21.6
Instance_MAP5_1_2	70	78	1400	910	15.2	106.8
Instance_MAP5_1_3	70	78	600	600	8.5	186.3
Instance_MAP5_2_1	115	101	220	220	2.5	374.3
Instance_MAP5_2_2	115	101	1780	1650	5.3	189.2
Instance_MAP5_2_3	115	101	550	170	23.4	254.2
Instance_MAP6_0_1	84	92	270	250	23.8	269.8
Instance_MAP6_0_2	84	92	2520	2140	14.5	82.2
Instance_MAP6_0_3	84	92	2210	2210	12.3	219.1
Instance_MAP6_1_1	155	112	500	440	7.2	431.3
Instance_MAP6_1_2	155	112	4390	3500	9.9	321.8
Instance_MAP6_1_3	155	112	3140	1960	25.6	671.3
Instance_MAP6_2_1	175	181	660	660	21.1	719.2
Instance_MAP6_2_2	175	181	6790	5420	16.1	995.3
Instance_MAP6_2_3	175	181	7180	7180	20.8	1701.8

Table 13: Relaxation-based heuristics: solution value and computation time