symmetry

Article

Optimizing Connectivity and Coverage for Millimeter-Wave-Based Networks

Pablo Adasme [1,*], Ali Dehghan Firoozabadi [2,*] and Sergio Cordero [1]

1 Centre for Multidisciplinary Research in Telecommunication Technologies (CIMTT), Department of Electrical Engineering, Faculty of Engineering, Universidad de Santiago de Chile, Avenida Víctor Jara 3519, Estación Central, Santiago 7800002, Chile; sergio.cordero@usach.cl
2 Department of Electricity, Universidad Tecnológica Metropolitana, Av. Jose Pedro Alessandri 1242, Santiago 7800002, Chile
* Correspondence: pablo.adasme@usach.cl (P.A.); adehghanfirouzabadi@utem.cl (A.D.F.)

Abstract: In this article, the problem of achieving the minimum backbone connectivity cost while simultaneously maximizing user coverage for 5G millimeter-wave (mmWave)-based networks is considered. Let $G = (N, E)$ be an input graph instance with a set of nodes N (base stations) and a set of edges E. It is assumed that G represents a wireless backbone network. Let M represent a set of users to be covered by G. Note that mmWave technology has been considered in the literature as an important candidate solution for 5G networks due to its low latency. However, there remain some problems to be addressed before using this technology. A serious one is that millimeter waves cannot cover large transmission distances. In this article, the proposed methodology consists of formulating mixed-integer programming models to deal with the problem from a management point of view. Our models allow the determination of which of the nodes of G should be active and connected while simultaneously maximizing the total number of covered users. The models are solved with the CPLEX solver using its branch and cut and automatic Benders decomposition algorithms. For this purpose, symmetric complete and sparse graphs are considered. Using the symmetry concept, it is considered that the distances between base stations and users and between base stations themselves are symmetrical. Finally, an efficient local search meta-heuristic is proposed that allows for finding near-optimal solutions. Our numerical experiments indicate that the problem is hard to solve optimally. Thus, instances with up to 40 nodes and 500 users have been solved to optimality so far. In particular, it is observed that one of the models presents slightly better performance in terms of CPU time. Finally, the heuristic approach allows us to obtain tight solutions with less computational effort when dealing with even larger instances of the problem.

Keywords: network planning; optimal connectivity and user coverage; 5G millimeter-wave-based networks; mixed-integer programming models; branch and cut method and Benders decomposition; local search meta-heuristic

Citation: Adasme, P.; Firoozabadi, A.D.; Cordero, S. Optimizing Connectivity and Coverage for Millimeter-Wave-Based Networks. *Symmetry* **2024**, *16*, 123. https://doi.org/10.3390/sym16010123

Academic Editor: Michel Planat

Received: 29 December 2023
Revised: 12 January 2024
Accepted: 16 January 2024
Published: 19 January 2024

1. Introduction

Using 5G technology for wireless communication can make it easier to send a lot of data quickly. It has the potential to overcome problems like limited bandwidth, access issues, and delays. This means that future technologies, like "5G and Beyond", could let us send data at speeds of hundreds of Megabits per second with very short delays, less than 1 millisecond. As a consequence, it will make it possible to connect simultaneously billions of devices and it can also lead to the creation of entirely new and innovative applications that we haven't seen before [1]. Examples of where these new applications could be used cover a wide range of areas. This includes things like mobile health, self-driving cars, manufacturing and entertainment, education, smart grids, big data analysis, smart cities and homes, aerospace, ocean exploration, emergency response, and mobile platforms, among others [1]. As can be verified from the literature, a base station (BS) can transmit data with up to a radius of approximately 200ms while using 5G mmWave technology [2]. So far, the experimental evidence suggests that deploying base stations (BSs) with a radius of at least 200 m can solve the coverage issue in outdoor areas by utilizing direct line-of-sight communication. However, it is important to note that opting for smaller radius values would mean needing a significantly larger number of base stations to ensure complete coverage for users [2,3]. In particular, note that any pair of BSs can be connected by using cables to form the backbone network structure. Consequently, it is necessary to adapt current infrastructures [4]. This will be essential for the deployment of future 5G-based networks to manage the increasing demand for data usage and broader coverage, as recommended in [2,5].

In this article, the problem of achieving the minimum backbone connectivity cost while simultaneously maximizing user coverage using 5G mmWave technology is considered. For this purpose, let us consider $G = (N, E)$ to be an input graph instance with a set of nodes (BSs) $N = \{1, \ldots, n\}$ and a set of edges (or connection links) $E = \{\{i, j\}\ i, j \in N, (i < j)\}$. Thus, it is assumed that G represents a wired or wireless backbone network. Also, let $M = \{1, \ldots, m\}$ represent a set of users to be covered by G. Note that mmWave technology has been considered in the literature as an important candidate solution for 5G networks due to its low latency. Nevertheless, there remain some practical problems to overcome before using this technology for real network applications. A serious one is that millimeter waves cannot cover large radial transmission distances. Consequently, in this article, the main contributions are to propose mixed-integer programming models to deal with this problem. In particular, our models allow the determination of which of the nodes of G should be active and connected at the lowest connectivity cost while simultaneously maximizing the total number of covered users. More precisely, two flow-based models, a Miller–Tucker–Zemlin-based one and an exponential one, are proposed [6]. In general, these models mainly differ in the way a set of constraints forms a spanning tree backbone. In our research article, we consider scenarios where network nodes remain stationary, which is a common characteristic of sensor networks, for instance, or in catastrophic scenarios where it might be required to set up a wireless network communication rapidly, such as in earthquakes or pandemic situations. If a particular user is not covered, they will disconnect from the backbone network, as occurs, for example, on rural highways. This underpins our assertion that the novel models we introduce have direct applicability to wireless networks. This assumption aligns well with the prevalent conditions in sensor network deployments. However, we recognize the more intricate scenario involving mobile BSs and users, capable of traversing various application domains. Note that even in dynamic scenarios, our proposed models retain their utility, offering an alternative avenue to derive optimal network configurations. This optimization task is facilitated by considering optimization tools such as the CPLEX solver, an industry-leading solution in the realm of operations research [7]. Its efficacy is underscored by its inclusion of exact approaches like branch and cut (B&C) and Benders algorithms, aspects well-documented in the literature [7,8]. Within the context of wireless sensor networks (WSNs), a myriad of applications has been explored in the existing body of work. It is also observed that additional metrics such as users' and BS's mobility, quality of service requisites, and security provisions will certainly lead to future modeling approaches. Nevertheless, our models and algorithms address the pivotal issue of distance—both between BSs and users and among BSs themselves. This treatment is rooted in the understanding that many metrics are contingent on distance, often exhibiting degradation as distances increase. Note that even when both BSs and users are in motion, our models and algorithms remain efficient, consistently yielding near-optimal solutions remarkably, as demonstrated through our numerical experiments. This adaptability positions them not only as practical tools but also as a benchmark for the development of novel and more efficient algorithmic approaches. While they may not be amenable to exact solutions within a one-hour CPU time limit in certain cases, the extensive use of rigorous upper and lower bounds ensures that the solutions attained are indeed near-optimal. Our contribution, therefore, lies not just in the solutions themselves but in the robust methodology employed to ascertain their quality. In summary, the contributions outlined in our novel models exhibit a remarkable degree of generality, making them adaptable to a range of contexts, contingent upon the specific application domain. This flexibility underscores their value as a versatile toolset for addressing diverse wireless network scenarios. Finally, in a situation where the BSs and users move, there are always alternative ways of finding feasible solutions faster. For this purpose, our second algorithm uses an iteration parameter *MaxIter* which allows it to run for a predefined number of iterations. The latter can be handled efficiently to find good solutions, as is seen from the velocities reported in the last two figures, where the algorithm reaches near-objective values very fast. Since all our models are solved with the CPLEX solver using its branch and cut and its automatic Bender decomposition algorithmic options, a set of complete and sparse connected input graphs to form the backbone network is considered. So, the assumption is that the positions of all users and base stations BSs are randomly distributed within a square area. Additionally, the assumption is made that in an outdoor setting, all users and base stations BSs can communicate using a direct line-of-sight channel. For this purpose, the maximum CPU time limit allowed for the CPLEX solver is arbitrarily set to be at most one hour when solving each of the instances, and also the CPLEX Mipgap option is set to be zero and less than one percent [7]. Finally, an efficient local search meta-heuristic algorithm is proposed that allows for finding tight solutions in significantly low CPU time when compared to the proposed models either for small- or large-sized instances of the problem. As far as we know, our proposed models and solution approaches are new to the literature and therefore enrich the existing literature to deal with the problem from a management point of view. Note that our proposed models allow us to incorporate a new aspect to the problem which is considering the distance between users and base stations BSs, and the optimal connectivity between all BSs. Finally, we mention that our proposed models could cover all users when the maximum user coverage metric becomes more relevant than considering the connectivity cost of the backbone network. Total coverage could also be ensured by using a larger amount of candidate sites for the BSs to be activated in the network.

Consequently, a direct comparison with other methods is not straightforward as the modeling and algorithmic approaches are completely new. From the literature, only one article was found to be as close as possible to the connectivity problem, where a flow-based formulation is proposed. In this article, that flow model is adapted to our new more intricate problem. Unfortunately, its performance in terms of obtaining optimal solutions when compared to our new proposed models is poor. Moreover, all our proposed models allow us to determine the optimal number of BSs and which of them should be active to form the backbone network structure. This consideration also takes into account the characteristics of 5G mmWave technology. It is worth noting that the literature has not yet explored the aspect of wireless networks with random deployments, such as sensor networks. Note that the rationale behind employing the CPLEX solver primarily rests on its exceptional, unique, and potent algorithmic capabilities, as evidenced by its success in tackling challenging optimization problems in the literature [7]. Finally, we mention that this work corresponds to a larger version of the article presented at the IEEE conference [9].

The organization of the article is as follows. In Section 2, first, some recent studies from the literature that are closer to the optimal network planning problem considered in this article are reviewed. Then, in Section 3, the optimization problem at hand is briefly explained, and each proposed mathematical formulation is presented in detail. Subsequently, in Section 4, the algorithmic approaches used to solve the optimization problem are presented and explained. More precisely, an exact algorithm that consists of adding sub-tour elimination constraints to our exponential model until the optimal solution is obtained without cycles is presented. Then, how the Bender decomposition algorithm finds the optimal solutions and the automatic Benders CPLEX version as well are briefly explained [7]. Finally, in this section, we introduce and clarify the proposed meta-heuristic. Moving forward to Section 5, we carry out extensive numerical experiments, presenting and discussing the results obtained from the proposed models and algorithms. The comparison is based on factors such as CPU time and the optimal or best solutions achieved. Lastly, in Section 6, we conclude the article while discussing insights for potential future research.

2. Related Work

The BS placement problem using 5G mmWaves for outdoor areas has not been investigated so far with sufficient depth in the literature. Some recently published works using mmWaves technology which are closely related to this network planning problem can be described as follows. In [10], the authors utilize mmWave radar technology to unobtrusively detect human falls. To this end, they collect data from healthy young volunteers with the radar mounted on the side wall or overhead of an experimental room. They consider a set of features that are manually extracted from the data and apply multilayer perceptron, random forest, k-nearest neighbor, and support vector machine classifiers on the features. Subsequently, they devise a convolutional neural network and conclude from their numerical results that the development of a hidden monitoring system for fall detection using mmWave radar is feasible. Finally, the authors mention that the optimal placement of the radars is unknown, and consequently, they locate them intuitively. Similarly, in [11], the authors propose a novel 3D geometry-based framework for deploying mmWave base stations in urban environments and also provide a solution for the optimum deployment of passive metallic reflectors to extend the radial coverage to non-line-of-sight areas. Specifically, they formulate their network planning problem as two independent optimization problems to maximize the coverage area, and minimize the deployment cost, while ensuring a desired quality-of-service level. Finally, they test the efficacy of their approach by using a generic map. The study conducted by the authors concludes that their network planning approach, utilizing mmWave technology, effectively reduces the need for passive metallic reflectors in non-line-of-sight areas, consequently lowering deployment costs. In [12], the authors investigate the practicality of relaying and caching in mmWave-based networks, assuming the coexistence of base stations and relay nodes (RNs). Their findings suggest that increasing the number of RNs and expanding the cache size in mmWave networks presents a more cost-effective solution compared to simply increasing BS density for similar backhaul off-loading performance. Another work [13], explores advanced computational geometry concepts for placing numerous mmWave wall-mounted BSs in large urban areas. The authors assert that their study is the first to use mmWave cells on buildings, and their approach enables accurate modeling for small-cell mmWave-based networks, providing gigabit capacities for densely populated urban areas.

In addition to the mentioned works, recent publications on network planning issues for future communication networks include references [6,14]. In the specific case of [14], the authors tackle the broader challenge of maximizing user coverage while adhering to facility location and radial distance constraints in the context of 5G/6G wireless communication networks. They propose mixed-integer linear programming models, drawing inspiration from the classical p-median problem in optimization literature. Notably, the authors introduce non-overlapping radial distance constraints for various pairs of base stations (BSs). Their experiments involve solving instances with up to 100 antennas and 1000 users. The study concludes that incorporating a larger number of radius values enhances the flexibility and accuracy of solutions, albeit at a higher computational cost. In a similar vein, in [6] the author proposes generic and novel mixed-integer linear programming models for the p-median problem while imposing ring, tree, and star backbone topology constraints on the facility locations. The author also mentions that the proposed models find applications in wired and wireless network design, computer networks, transportation, water supply, and electrical networks, to name a few. The author further proposes a variable neighborhood search meta-heuristic for each network backbone structure. Finally, the author concludes, based on substantial numerical experiments, that the ring models are harder to solve with CPLEX than the tree and star ones. Ultimately, the proposed meta-heuristics in [6] proved to be highly efficient as they allowed better feasible solutions to be obtained compared to CPLEX, and with significantly less computational effort. Other related works can be consulted in references [2,15,16] and in references therein. As can be observed from these previous works, none of them takes into account the radial transmission distance limitation of 5G mmWave technology so far. Consequently, the proposed mathematical models in this article change in structure significantly to deal with the network planning problem from a management point of view.

To conclude this section, it is mentioned that we observe only a few works in the literature that are directly related to the network planning problem using 5G mmWave technology. Consequently, our work will contribute to providing a new dimension of the problem, i.e., taking into account optimal connectivity. Since the objective function of our proposed models contains conflicting objectives, i.e., maximizing user coverage and minimizing backbone connectivity cost simultaneously, a weighting parameter $\alpha \in [0;1]$ is introduced to balance the degree of importance of each objective. Finally, it is mentioned that the direct implication for the mmWave networks is to at least find very good solutions, including optimal configurations.

3. Optimization Problem and Mathematical Formulations

In this section, first, the optimization network planning problem we are dealing with is introduced. For this purpose, an example of an input graph network and the optimal solution obtained is presented. Subsequently, each proposed model is presented and explained in detail.

3.1. Optimization Problem

As mentioned in Section 1, a network deployment that is composed of users and BSs that are randomly located inside a square area of 1 km^2 is considered. It is also assumed that each user can connect to at least one BS located within a predefined radial transmission symmetric distance. For this purpose, it is assumed that the distances between base stations and users and between base stations themselves are symmetric. Thus, the main goal is to cover the maximum number of users in the network with at least one of the BS (nodes) while simultaneously minimizing the total connectivity cost of the BSs. In particular, the main focus is put on the context of 5G mmWave since it is assumed that the distances between users and BSs are particularly small. Note that the latter implies using a larger number of BSs to cover the entire network.

In Figures 1 and 2, the input backbone network together with the optimal solutions, one for a sparse graph and another one for a complete graph, are presented. Note that the input graphs are given as digraphs as they will be required to solve the proposed models, whereas the optimal backbone solutions are drawn employing undirected graphs. The digraph version of each undirected input graph is, thus, denoted hereafter by $H = (N, A)$, where A represents a set of directed arcs, i.e., the set $E = \{\{i, j\}\ i, j \in N, (i < j)\}$ is transformed into the set $A = \{(i, j)\ i, j \in N, (i \neq j)\}$. Note that the arcs obtained in the optimal solutions can be simply dropped and replaced by edges instead. As can be seen in Figure 1, the input backbone network is connected. It is also seen that eight nodes of the optimal solution obtained are active and connected while forming a tree backbone topology. Only two BSs are not included in the solution. This means that the maximum number of covered users, which is 30 out of 50 for the example of Figure 1, can connect with at least one of the active nodes. Similarly, from Figure 2, it is observed that two BSs are not part of the optimal solution, and thus, not required to cover users. In this example, the number of covered users is 27 out of 50.

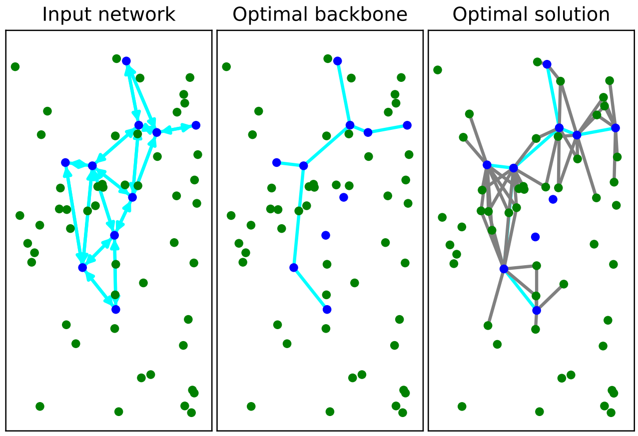

Figure 1. Input network, the optimal backbone solution of a sparse graph, and the users connected to the active BSs. The solution is obtained using $\alpha = 0.5$ for $m = 50$ users. The network is composed of $n = 10$ nodes using radial transmission distances of 300 ms between BSs. The radial transmission distance between users and BSs is 200 ms. BSs are colored blue and users are colored green.

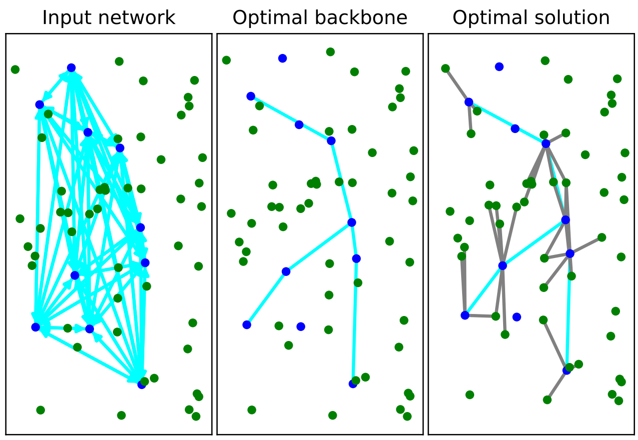

Figure 2. Input network, the optimal backbone solution of a complete graph. The solution is obtained using $\alpha = 0.5$ for $m = 50$ users. The network is composed of $n = 10$ nodes. The radial transmission distance between users and BSs is 200 ms. BSs are colored blue and users are colored green.

3.2. Mathematical Formulations

To formulate a first mixed-integer linear flow-based model, let $M = \{1, \ldots, m\}$ and $N = \{1, \ldots, n\}$ denote the corresponding sets of users and BSs, respectively. The binary variable x_j for each user $j \in M$ equals one when user j is covered by some of the active BSs, otherwise $x_j = 0$. Subsequently, the binary variable y_i should be equal to one if BS $i \in N$ is active, otherwise $y_i = 0$. Consequently, the first linear flow-based model for the 5G mm-Wave-based network, which is denoted hereafter by M_1, consists of maximizing the total number of covered users while simultaneously minimizing the total backbone connectivity cost of the active BSs. The model can be written as follows:

$$M_1 : \max_{\{x,y,z,f\}} \quad \left\{ \alpha \sum_{j \in M} x_j - (1 - \alpha) \sum_{(i,j) \in A} D_{ij} z_{ij} \right\} \tag{1}$$

$$\text{st} : \quad \sum_{i \in N} C_{ij} y_i \geq x_j, \forall j \in M \tag{2}$$

$$\sum_{(i,j) \in A} z_{ij} = \sum_{i \in N} y_i - 1 \tag{3}$$

$$\sum_{j|(r,j)\in A_r} z_{rj} = 1 \quad (4)$$

$$\sum_{j|(r,j)\in A_r} f_{rj} = \sum_{i\in N} y_i \quad (5)$$

$$\sum_{i|(i,j)\in A\cup A_r} f_{ij} - \sum_{i|(j,i)\in A} f_{ji} = y_j, \quad \forall j \in N \quad (6)$$

$$f_{ij} \leq n z_{ij}, \quad \forall (i,j) \in A \cup A_r \quad (7)$$

$$z_{ij} + z_{ji} \leq 1, \quad \forall (i,j) \in A \quad (8)$$

$$f_{ij} \leq n y_i, \quad \forall (i,j) \in A \quad (9)$$

$$f_{ij} \leq n y_j, \quad \forall (i,j) \in A \quad (10)$$

$$x \in \{0;1\}^m, y \in \{0;1\}^n, z \in \{0;1\}^{|A\cup A_r|},$$
$$f \in [0;\infty)^{|A\cup A_r|} \quad (11)$$

In model M_1, the variable z_{ij} for all $(i,j) \in A$ equals one if the connection link between nodes i and $j \in N$ are part of the backbone, otherwise it equals zero. Similarly, the non-negative variable f_{ij} for all $(i,j) \in A \cup A_r$ can take positive values if and only if variable $z_{ij} = 1$, otherwise f_{ij} should be equal to zero. For this purpose, it is considered an expanded digraph of $G = (N, E)$, as $H = (N \cup r, A \cup A_r)$, where the set of nodes in H considers an additional dummy node r and a set of arcs A_r composed of the arcs in A plus additional arcs with zero costs going from r to every node $j \in N$. Next, in the objective function (1), the parameter $\alpha \in [0;1]$ represents a weighting parameter that balances the degree of importance of each term. Each entry in the input matrix $C = (C_{ij})$, for all $i \in N$, $j \in M$ is a 0–1 value indicating whether user j is covered or not by BS i. In particular, note that it is possible to compute each of these entries by imposing a predefined radial transmission distance. If the distance of BS $i \in N$ reaches user $j \in M$, then $C_{ij} = 1$; otherwise, $C_{ij} = 0$. On the other side, each entry in the input matrix $D = (D_{ij})$ denotes the existing distance between BSs i and $j \in N$.

Observation 1. *Note that when $\alpha = 1$ in the objective function (1), the model M_1 reduces only to maximizing the total sum of covered users without taking into account the connectivity costs between BSs. On the other hand, when $\alpha = 0$ the output solution of the problem reduces to a trivial minimum-cost spanning tree. Furthermore, note that the solution-spanning tree to be obtained will be composed of one isolated node with an objective function value equal to zero.*

Thus, the underlying idea of using parameter $\alpha \in [0;1]$ is to find the maximum possible number of covered users while simultaneously connecting a spanning tree backbone of minimum cost which is formed with the active BSs. Note that distances are used for each entry in the input matrix $D = (D_{ij})$ for all i and $j \in N$ in (1). As a commonly recognized fact, the signal quality experiences a notable improvement with the transmission distance in a wireless communication channel. In simpler terms, when the emitter and receivers are in closer proximity, the power consumption tends to be lower, as highlighted in the study by [6].

Constraints (2) allow a particular user $j \in M$ to be covered if and only if at least one of the BSs is active and the user is located inside its sensing radius. Next, constraint (3) ensures that the backbone spanning tree obtained contains several edges equal to the number of nodes minus one. Constraint (4) ensures that only one arc goes from r to a unique node $j \in N$. Subsequently, constraints (5) ensure that an amount of $\sum_{i\in N} y_i$ units of flow enter into the unique arc (r,j) of variable z_{ij}, $(i,j) \in A \cup A_r$. Similarly, constraints (6) ensure that one unit of flow equals the incoming total flow minus the outgoing total flow for each active BS. Constraints (7) indicate that at most n units of flow should traverse the arc $(i,j) \in A \cup A_r$ if and only if variable $z_{ij} = 1$, i.e., if the nodes i and $j \in N$ are connected. Next, constraints (8) ensure that either arc (i,j) or arc (j,i) is part of the output solution of the input digraph. Constraints (9) and (10) are valid inequalities forcing the fact that variable f_{ij} cannot be positive if neither i nor j is an active BS. Finally, constraints (11) are domain constraints for the decision variables.

Another equivalent flow-based formulation adapted from [2] can be written as follows:

$$
\begin{aligned}
M_2: \max_{\{x,y,s,f,z,\theta\}} \quad & \left\{ \alpha \sum_{j \in M} x_j - (1-\alpha) \sum_{(i,j) \in A} D_{ij} z_{ij} \right\} \\
\text{st}: \quad & \sum_{i \in N} C_{ij} y_i \geq x_j, \forall j \in M \\
& \sum_{i \in N} s_i = 1 && (12) \\
& \sum_{j \in N} (f_{ji} - f_{ij}) + y_i = \sum_{j \in N} \theta_{ij}, \quad \forall i \in N && (13) \\
& \theta_{ij} \leq y_j, \forall i,j \in N && (14) \\
& \theta_{ij} \leq s_i, \forall i,j \in N && (15) \\
& \theta_{ij} \leq y_j + s_i - 1, \forall i,j \in N && (16) \\
& f_{ij} \leq n y_i, \quad \forall (i,j) \in A \\
& f_{ij} \leq n y_j, \quad \forall (i,j) \in A \\
& f_{ij} \leq n z_{ij}, \quad \forall (i,j) \in A \\
& z_{ij} + z_{ji} \leq 1, \quad \forall (i,j) \in A \\
& x \in \{0;1\}^m, y \in \{0;1\}^n, s \in \{0;1\}^n, \\
& z \in \{0;1\}^{|A|}, \theta \in [0;1]^{n^2}, f \in [0;\infty)^{|A|} && (17)
\end{aligned}
$$

In model M_2, the variable s_i for all $i \in N$ is used to represent the source node from which $\sum_{j \in N} y_j - 1$ units of flow must traverse the network formed with the active BSs. Otherwise, it equals zero. On the other hand, when $s_i = 0$, then the incoming minus outgoing flow of node $i \in N$ must be equal to -1. Note that this fact is simultaneously implied by constraints (12) and (13). Constraints (14)–(16) are standard linearization constraints [17]. More precisely, the binary quadratic products $\theta_{ij} = s_i y_j$, for all $i,j \in N$ are linearized with the constraints (14)–(16). Finally, the domain constraints of the decision variables are referenced in (17). The remaining constraints are similar to the ones presented in model M_1.

A Miller–Tucker–Zemlin-based constrained formulation can also be written as follows [18]:

$$
\begin{aligned}
M_3: \max_{\{x,y,u,z\}} \quad & \left\{ \alpha \sum_{j \in M} x_j - (1-\alpha) \sum_{(i,j) \in A} D_{ij} z_{ij} \right\} \\
\text{st}: \quad & \sum_{i \in N} C_{ij} y_i \geq x_j, \forall j \in M \\
& \sum_{(i,j) \in A} z_{ij} = \sum_{i \in N} y_i - 1 \\
& u_i \leq \sum_{j \in N} y_j, \forall i \in N && (18) \\
& u_i \geq y_i, \forall i \in N && (19) \\
& \sum_{i|(i,j) \in A} z_{ij} \leq y_j, \forall j \in N && (20) \\
& u_j - u_i - (n-1) z_{ij} - (n-3) z_{ji} \geq 2 - n, \forall (i,j) \in A && (21) \\
& z_{ij} + z_{ji} \leq y_i, \forall (i,j) \in A && (22) \\
& z_{ij} + z_{ji} \leq y_j, \forall (i,j) \in A && (23) \\
& x \in \{0;1\}^m, y \in \{0;1\}^n, z \in \{0;1\}^{|A|} && (24)
\end{aligned}
$$

where constraints (18)–(21), which are polynomials in number, avoid cycles in the output solution of the problem. Note that, in particular, constraint (20) ensures that each node of the backbone tree has at most one incoming arc. Next, constraints (22) and (23) are valid inequalities forcing the fact that either arc (i,j) or $(j,i) \in A$ is equal to one since the input graph is a digraph. And for this purpose, both nodes i and $j \in N$ must be active, which is reflected by variables y_i and y_j, respectively.

The last formulation proposed has an exponential number of constraints in (25). More precisely, for each subset $S \subseteq N$,a constraint is imposed that avoids forming a cycle with the nodes belonging to S. The model reads

$$
\begin{aligned}
M_4: \max_{\{x,y,z\}} \quad & \left\{ \alpha \sum_{j \in M} x_j - (1-\alpha) \sum_{(i,j) \in A} D_{ij} z_{ij} \right\} \\
\text{st}: \quad & \sum_{i \in N} C_{ij} y_i \geq x_j, \forall j \in M \\
& \sum_{(i,j) \in A(S)} z_{ij} \leq |S| - 1, \forall S \subseteq N \qquad (25) \\
& z_{ij} + z_{ji} \leq y_i, \forall (i,j) \in A \\
& z_{ij} + z_{ji} \leq y_j, \forall (i,j) \in A \\
& \sum_{(i,j) \in A} z_{ij} = \sum_{i \in N} y_i - 1 \\
& \sum_{i | (i,j) \in A} z_{ij} \leq y_j, \forall j \in N \\
& x \in \{0;1\}^m, y \in \{0;1\}^n, z \in \{0;1\}^{|A|} \qquad (26)
\end{aligned}
$$

The rest of the constraints are the same as in the above models, thus not labeled. Finally, the domain constraints of the decision variables are written in (26).

We mention that another relevant metric that could be considered as part of future research in the proposed models is the interference between small cells, as pointed out in [5]. It is also possible to consider maximizing the capacity and minimizing power consumption, leading to harder nonconvex nonlinear formulations. Ultimately, note that these metrics can be added as constraints in the optimization models as well.

4. Algorithmic Approaches

In this subsection, the two algorithms are presented and explained. One is exact and consists of adding cycles iteratively to model M_4 until no cycle is obtained in an incumbent solution [19]. The second one is an approximate local search heuristic that is highly efficient as it allows obtaining near-optimal solutions in very short CPU times. Finally, the reader is referred to some relevant references involving the well-known exact methods of branch and cut and Benders decomposition of the CPLEX solver [7].

4.1. Exact Algorithm for Solving Model M_4

The iterative procedure to avoid having cycles in model M_4 consists of ignoring, at the beginning of the algorithm, the exponential number of constraints (25). In other words, first, model M_4 is solved without constraints (25). If an output solution containing cycles is obtained, we find one of the cycles arbitrarily, as there can be many of them, and re-optimize while adding the particular constraint avoiding that particular cycle. Then, Algorithm 1 continues with that iterative process. Note that at a certain iteration one may obtain a solution without cycles: that solution would be the optimal one. The procedure is depicted in Algorithm 1. The method is simple and quite general. Note that any cycle, if it exists, can be detected by a depth-first search procedure [20]. Also, note that the number of constraints in (25) is at most $\mathcal{O}(2^n)$. Consequently, Algorithm 1 converges to the optimal solution of the problem in at most $\mathcal{O}(2^n)$ outer iterations (see the proof of theorem 2 in [19]). In step 0, Algorithm 1 initializes the iteration variable $\nu = 0$ and it is assumed that there is at least one cycle to enter the loop. Next, it initializes the set of constraints of type (25) of M_4. Note that initially this set is empty. Subsequently, in step 1, Algorithm 1 enters the while loop that is broken only if there are no further cycles in the incumbent solution. The inner loop is self-explained, and in particular the method $SearchCycle()$ allows for finding a cycle using the depth-first search algorithm [20].

Algorithm 1: Exact iterative algorithm for solving M_4.

Data: A problem instance of M_4
Result: An optimal solution (x^*, y^*, z^*) for M_4 with objective function value f^*
Step 0: Set $\nu = 0$ and $Cycles = 1$;
Let $\mathcal{C}$ represent the set of cycle elimination constraints in (25) of M_4; $\mathcal{C} = \emptyset$;
Step 1: **while** *(Number of cycles is positive)* **do**
  Let M_{4_ν} be the problem obtained from M_4 by removing the constraints (25) at iteration ν. Solve model M_{4_ν} using all cycle elimination constraints in $\mathcal{C}$ and let (x^ν, y^ν, z^ν) be the optimal solution of value f^ν obtained at iteration ν;
  $\mathcal{C} = \mathcal{C} \cup SearchCycle(y^\nu, z^\nu)$;
  Add the corresponding constraints to avoid the accumulated cycles in $\mathcal{C}$;
  Number of current cycles $= card(\mathcal{C})$;
  Set $\nu = \nu + 1$;
$f^* = f^\nu$;
return (x^*, y^*, z^*, f^*);

4.2. Approximate-Based Local Search Heuristic

Now, the approximate-based local search heuristic, Algorithm 2, is presented and explained. This algorithm allows for obtaining near-optimal solutions in a short CPU time for the optimization problem. The algorithm receives an instance of the problem and returns a near-optimal solution. In particular, in step 0, Algorithm 2 initializes the set M, finds the optimal spanning tree T of G, and computes its objective function value. In the beginning, it performs this using all nodes of the backbone. Then, it counts the number of covered users. The latter is achieved by using matrix $C = C(i, j)$ for all $i \in N$ and for all $j \in M$. Subsequently, it calculates the cardinality of the remaining users, i.e., the number of users who are not covered by T. This allows for computing the objective function value and saving it as the best found so far. Finally, the best current solution found is saved. Next, in step 1 it enters into a while loop which is run for a maximum predefined number of iterations $MaxIter$. The process is repeated and inside the loop consists of initializing the set of users M while including all users. Next, it randomly partitions the set of users into two subsets U_1 and U_2, where $M = U_1 \cup U_2$. Next, the algorithm increments by one unit the number of iterations $Iter$. Then, it performs the following three times. It randomly generates a number $v \in (0; 1)$ and asks if $v <= 0.5$ and U_1 is not empty. If this is the case, it randomly chooses a user from U_1, removes it from U_1, and adds it to U_2. On the other hand, if $v > 0.5$ and U_2 is not empty, it randomly picks a user from U_2, removes it from U_2, and adds it to U_1. Subsequently, it performs several tasks only if U_1 is not empty. More precisely, first, it constructs a sub-graph $G_s(U_1, E(U_1))$ of G, where the set $E(U_1)$ represents all incident edges to the nodes of U_1. Then, it asks if the sub-graph G_s is connected. If this is the case, it finds the optimal spanning tree T_s of G_s and computes its objective function value. Next, it counts the set of covered users. The latter is performed for each $n \in T_s$ and $j \in M$ using matrix $C = C(n, j)$. Subsequently, it computes the cardinality of the remaining users in M and calculates the objective function value according to the new backbone tree T_s. If the new solution obtained with T_s is better than the best found so far, then it is saved as the best incumbent solution obtained. Finally, when the process ends, the algorithm returns the best solution found.

Algorithm 2: Approximate-based local search heuristic for solving the optimization problem.

```
Data: A problem instance of the optimization problem.
Result: A near-optimal solution for the problem.
Step 0;
M = {1, ..., m};
Find the optimal spanning tree T of G and compute its objective function value;
foreach (n ∈ T) do
    foreach (j ∈ M) do
        if (C(n, j) = 1) then
            Remove user j from set M because it is covered;
Compute the cardinality of the remaining users in M;
Compute the objective function value and save it as the best value found so far;
Save the incumbent solution;
Step 1;
while (Iter <= MaxIter) do
    M = {1, ..., m};
    Partition the set of users into subsets U1 and U2 randomly where M = U1 ∪ U2;
    Iter=Iter+1;
    foreach (i ∈ [1, 2, 3]) do
        Generate a random number v ∈ (0; 1);
        if (v <= 0.5 and U1 is not empty) then
            Pick randomly a user from U1, remove it from U1 and add it to U2;
        if (v > 0.5 and U2 is not empty) then
            Pick randomly a user from U2, remove it from U2 and add it to U1;
    if (U1 is not empty) then
        Construct a sub-graph Gs(U1, E(U1)) of G;
        if (Gs is connected) then
            Find the optimal spanning tree Ts of Gs and compute its objective
              function value;
            foreach (n ∈ Ts) do
                foreach (j ∈ M) do
                    if (C(n, j) = 1) then
                        Remove user j from set M because it is covered;
    Compute the cardinality of the remaining users in M;
    Compute the objective function value accordingly and save it as a new
      objective function value;
    if (The new solution is better than the best found so far) then
        Replace the new one as best found so far;
        Save the incumbent solution;
return Best solution obtained;
```

5. Results and Discussion

In this section, substantial numerical experiments are conducted to compare the performances of the proposed models M_1, M_2, M_3, M_4, and the proposed algorithms. To this end, a Python code is implemented using the CPLEX solver [7]. In particular, the models M_1, M_2, and M_3 are solved with both the branch and cut and the Benders decomposition options of the solver. Note that model M_4 has no continuous variables and, consequently, Benders approach cannot be applied [8]. In particular, for solving the models M_1, M_2, and M_3, the CPLEX solver is used with default options. The maximum CPU time is limited to at most one hour. Consequently, if a particular reported objective value is obtained in 3600 s or more, it means that the solver is reporting the best solution obtained in one hour. Otherwise, it corresponds to an optimal solution to the problem. Note that the optimization problems of this type are NP-hard due to their discrete nature. Thus, both algorithms used by the CPLEX solver have exponential complexity [7]. The numerical experiments were performed on an Intel(R) 64-bit core (TM) i5-8400 CPU 2.81 GHz with 16 G of RAM under Windows 10. Randomly sparse and complete input graphs were generated. In the latter, it was assumed that all active BSs can be connected using cables, for instance. Whilst for the sparse graphs it was assumed that the radial transmission range between BSs was 300 ms. The dimensions of the graphs in terms of the number of nodes and the number of users were $N = \{40; 60; 100\}$ and $M = \{250; 500; 1000\}$, respectively. Finally, it was assumed that each user could connect to an active BS if and only if the user was located inside a radial transmission range of at most 200 ms. Each of the coordinates for the nodes and users was generated within a square area of 1 km^2 according to a uniform distribution function. Thus, each entry of the input distance matrix D was computed using these coordinates.

In Tables 1 and 2, numerical results are reported for the models M_1 and M_2, respectively. In particular, Table 1 reports numerical results for the sparse graphs. Whereas Table 2 reports numerical results for the complete ones. In both tables, the legends are the same. In columns 1 and 2, the instance number and the density of the graph are presented. The latter is computed by dividing the number of edges by the total number of edges of a complete graph having the same number of nodes. Column 3 reports the weighing parameter α. Next, columns 4–6 and 7–9 report the best objective value obtained, the number of branch and bound nodes, and the CPU time in seconds for model M_1 when solved with the branch and cut and Benders algorithms, respectively. Similarly, columns 10–12 and 13–15 report the same information for M_2.

Table 1. Numerical results obtained with M_1 and M_2 for sparse connected graphs.

#	Density	α	M_1 (B&C)			M_1 (Benders)			M_2 (B&C)			M_2 (Benders)		
			Best	*B&Bn*	CPU (s)	Best	*B&Bn*	CPU (s)	Best	*B&Bn*	CPU (s)	Best	*B&Bn*	CPU (s)
Using $n = 40$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs														
1	23.58	0.25	59.45	58,843	95.37	59.45	155,657	50.14	59.45	919,353	3600.09	59.45	72,499	32.81
2		0.5	121.96	36,497	63.01	121.96	80,225	23.32	121.96	623,892	3600.07	121.96	85,507	22.06
3		0.75	184.48	15,516	22.13	184.48	7384	3.25	184.48	633,256	3600.23	184.48	123,394	104.57
4		1	247.0	0	0.12	247.0	960	0.37	247.0	0	0.34	247.0	0	0.26
Using $n = 40$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs														
5	20.12	0.25	121.64	27,375	35.82	121.64	150,203	79.89	121.64	1,103,803	3600.06	121.64	66,042	31.03
6		0.5	246.43	7815	12.07	246.43	16,256	7.56	246.43	909,962	3600.07	246.43	136,101	78.51
7		0.75	371.21	3165	4.18	371.21	13,676	7.18	371.21	1,290,487	3600.06	371.21	15,607	12.1
8		1	496.0	0	0.09	496.0	1083	0.46	496.0	0	1.25	496.0	0	0.32
Using $n = 60$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs														
9	21.41	0.25	59.63	256,195	3600.06	59.47	374,124	3602.37	59.56	97,953	3600.2	59.52	452,293	3601.43
10		0.5	122.42	113,347	3600.07	122.37	439,433	3600.92	122.39	44,628	3601.76	121.93	113,698	3604.0
11		0.75	185.21	228,717	3600.42	185.14	197,159	3600.54	185.17	38,209	3606.07	185.01	88,191	3601.43
12		1	248.0	0	3.1	248.0	3292	78.78	248.0	0	1.71	248.0	0	1.71
Using $n = 60$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs														
13	21.58	0.25	116.32	345,380	3600.04	116.27	374,371	3600.11	116.32	75,110	3604.03	115.81	173,944	3628.18
14		0.5	235.55	196,124	3600.21	235.56	869,084	3602.04	235.55	46,342	3609.23	235.5	111,989	3604.75
15		0.75	354.77	496,438	3600.43	354.77	545,824	3604.79	354.76	50,419	3602.45	354.06	19,110	3601.76
16		1	474.0	0	4.29	474.0	2547	55.48	474.0	0	1.43	474.0	0	1.78

Table 2. Numerical results obtained with M_1 and M_2 for complete connected graphs.

#	Density	α	M_1 (B&C)			M_1 (Benders)			M_2 (B&C)			M_2 (Benders)		
			Best	***B&Bn***	**CPU (s)**	**Best**	***B&Bn***	**CPU (s)**	**Best**	***B&Bn***	**CPU (s)**	**Best**	***B&Bn***	**CPU (s)**
			Using $n = 40$, $m = 250$, and a radius of 200 m for users											
1	100	0.25	59.5	15,162	80.75	59.5	37,828	22.11	59.5	200,778	3606.42	59.5	100,908	684.62
2		0.5	122.0	12,088	82.32	122.0	21,707	14.67	121.98	178,175	3600.64	122.0	251,362	2044.93
3		0.75	184.5	23,701	142.43	184.5	7472	12.25	184.5	330,250	3600.21	184.5	25,619	53.4
4		1	247.0	0	0.32	247.0	0	0.07	247.0	0	0.23	247.0	0	0.31
			Using $n = 40$, $m = 500$, and a radius of 200 m for users											
5	100	0.25	121.06	18,239	79.29	121.06	3025	4.53	121.06	232,444	3603.07	121.06	103,016	1118.95
6		0.5	245.37	4504	33.35	245.37	3070	2.93	245.37	244,223	3600.89	245.37	45,515	432.28
7		0.75	369.68	8508	29.0	369.67	1117	2.14	369.68	494,094	3603.0	369.68	6468	93.06
8		1	494.0	0	0.21	494.0	0	0.07	494.0	0	2.71	494.0	0	0.35
			Using $n = 60$, $m = 250$, and a radius of 200 m for users											
9	100	0.25	59.63	73,332	3600.07	59.21	642,611	3600.1	59.54	49,067	3609.0	57.57	3288	3601.4
10		0.5	122.4	108,214	3600.15	121.76	1,061,065	3601.32	122.29	13,440	3602.01	120.31	2444	3601.96
11		0.75	185.2	97,016	3603.7	185.03	209,460	3600.76	185.18	14,894	3604.04	184.53	1413	3603.48
12		1	248.0	0	4.59	248.0	0	0.28	248.0	0	1.03	248.0	0	1.23
			Using $n = 60$, $m = 500$, and a radius of 200 m for users											
13	100	0.25	116.32	85,382	3601.98	116.25	338,278	3600.78	116.33	18,521	3602.01	0.0	0	3600.4
14		0.5	235.55	60,537	3600.95	235.53	469,564	3601.87	235.52	23,259	3608.82	0.0	0	3600.17
15		0.75	354.77	75,385	3601.1	354.77	348,252	3602.75	354.77	74,468	3609.1	0.0	0	3600.48
16		1	474.0	0	8.53	474.0	0	2.7	474.0	0	7.7	474.0	0	1.2

In Table 1, from the sparse graph results, it is mainly observed that, in general, M_1 outperforms M_2. First, one can note that the best solutions obtained with M_1 are the optimal ones for all the instances when using $n = 40$ nodes, either using B&C or Benders algorithms. On the other hand, it is observed that M_2 does not optimally solve the instances with the B&C algorithm. However, it can solve all the instances optimally using Benders algorithm. These observations can be confirmed by looking at the columns of the CPU times and the number of branching nodes. Subsequently, one can see that none of the graphs composed of $n = 60$ nodes can be solved optimally in one hour of CPU time. The latter shows that the graph instances become significantly harder to solve when n grows. Finally, it is observed that for the value of $\alpha = 1$, the problem is solved easily with any of the models and methods. In particular, for the complete graph instances reported in Table 2, it is possible to observe similar trends. More precisely, one can see that M_1 can solve all the instances optimally for $n = 40$ with both methods. It is also observed that none of the instances can be solved optimally in one hour of CPU time for $n = 60$ nodes. Furthermore, it is observed that M_2 obtains the optimal solutions for all the instances for $n = 40$ with Benders approach. The latter cannot be achieved with the pure B&C method. Lastly, it is seen that all the graph instances are solved optimally and very rapidly for the value of $\alpha = 1$, which is the optimization problem without taking into account the connectivity cost of the backbone. Ultimately, from both Tables 1 and 2, one can observe that the increase in users, going from 500 to 1000, does not have a significant impact on the performance when solving the problem. In Tables 3 and 4, numerical results are reported for models M_3 and M_4. In particular, Tables 3 and 4 report numerical results for the sparse and complete graphs, respectively. In both tables, the legends are the same. More precisely, in columns 1–3, the instance number, the density of the graph, and the value of the parameter α are presented. Next, columns 4–6 and 7–9 report the best objective function value obtained, the number of branch and bound nodes, and the CPU time in seconds for model M_3 when using the B&C and Benders algorithmic approaches, respectively. Finally, columns 10–13 report the best objective function value obtained with Algorithm 1, the CPU time in seconds required to obtain that solution, the number of iterations of Algorithm 1, and the gap which is obtained by subtracting the best objective value obtained with M_4 with the best objective value obtained with the remaining models and dividing this quotient by the latter value. The gap is finally multiplied by 100 to present percentages. Note that the objective values obtained with models M_1, M_2, and M_3 are lower bounds as they are feasible or in the best case optimal solutions, while the values obtained with M_4 are in the best case the optimal solutions or upper bounds since they are solved iteratively without including all sub-tour elimination constraints within each iteration of Algorithm 1.

Table 3. Numerical results obtained with M_3 and M_4 for sparse connected graphs.

#	Density	α	M_3 (B&C)			M_3 (Benders)			M_4			
			Best	*B&Bn*	CPU (s)	Best	*B&Bn*	CPU (s)	Best	CPU (s)	Iter.	Gap
Using $n = 40$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs												
1	23.58	0.25	59.45	7,053,083	3601.21	59.45	4,436,010	3607.17	59.5	3622.52	312	0.09
2		0.5	121.96	3,277,430	3600.54	121.96	3,876,720	3600.32	122.0	3609.39	354	0.03
3		0.75	184.48	2,246,298	3600.43	184.48	2,568,663	3600.39	184.49	3604.51	456	0.0
4		1	247.0	12	2.0	247.0	22	2.06	247.0	0.65	32	0.0
Using $n = 40$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs												
5	20.12	0.25	121.64	2,310,219	3600.7	121.63	5,923,055	3601.34	121.77	3688.81	249	0.1
6		0.5	246.43	5,124,841	3601.42	246.42	4,388,909	3602.64	246.5	3675.53	283	0.02
7		0.75	371.21	3,810,306	3600.42	371.2	3,881,511	3600.46	371.24	3656.92	376	0.0
8		1	496.0	0	1.89	496.0	168	4.4	496.0	0.51	21	0.0
Using $n = 60$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs												
9	21.41	0.25	59.61	2,739,800	3679.92	59.5	1,469,457	3647.25	59.96	3740.32	73	5.92
10		0.5	122.41	1,374,975	3621.26	122.33	2,021,288	3608.48	122.64	3712.91	76	0.18
11		0.75	185.2	2,851,145	3702.12	185.18	1,550,639	3669.54	185.32	3654.23	75	0.06
12		1	248.0	0	8.56	248.0	157	7.76	248.0	1.28	32	0.0
Using $n = 60$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs												
13	21.58	0.25	116.34	2,518,469	3681.64	116.28	6,596,125	3605.45	116.62	3669.2	102	0.24
14		0.5	235.55	3,802,330	3607.9	235.52	3,095,730	3614.51	235.74	3763.58	107	0.08
15		0.75	354.77	3,260,836	3604.92	354.74	4,077,034	3601.5	354.87	3627.48	129	0.02
16		1	474.0	0	1.21	474.0	153	2.76	474.0	0.37	5	0.0

Table 4. Numerical results obtained with M_3 and M_4 for complete graphs.

#	Density	α	M_3 (B&C)			M_3 (Benders)			M_4			
			Best	*B&Bn*	CPU (s)	Best	*B&Bn*	CPU (s)	Best	CPU (s)	Iter.	Gap
Using $n = 40$, $m = 250$, and a radius of 200 m for users												
1	100	0.25	59.5	3,387,873	2498.84	59.5	6,118,703	3600.17	59.51	3632.32	254	0.02
2		0.5	122.0	1,913,208	3600.31	122.0	3,007,131	3600.39	122.0	3620.28	270	0.0
3		0.75	184.5	1,681,066	3600.42	184.49	2,235,475	3600.54	184.5	3281.32	320	0.0
4		1	247.0	0	1.4	247.0	0	0.32	247.0	0.2	2	0.0
Using $n = 40$, $m = 500$, and a radius of 200 m for users												
5	100	0.25	121.05	4,596,583	3601.4	121.05	3,106,327	3603.67	121.18	3625.98	225	0.1
6		0.5	245.37	1,743,075	3600.42	245.36	1,871,585	3600.56	245.44	3618.88	259	0.03
7		0.75	369.68	1,013,676	3600.56	369.68	4,572,259	3604.63	369.7	3616.54	374	0.0
8		1	494.0	0	0.5	494.0	0	0.17	494.0	0.39	2	0.0
Using $n = 60$, $m = 250$, and a radius of 200 m for users												
9	100	0.25	59.61	567,316	3602.4	59.59	506,317	3601.01	59.97	3621.19	56	0.61
10		0.5	122.42	308,986	3600.93	122.36	544,433	3605.35	122.64	3646.03	58	0.18
11		0.75	185.2	151,127	3601.96	185.17	269,632	3602.04	185.32	3610.28	69	0.06
12		1	248.0	0	0.4	248.0	0	0.21	248.0	0.85	2	0.0
Using $n = 60$, $m = 500$, and a radius of 200 m for users												
13	100	0.25	116.33	463,961	3605.71	116.25	329,260	3602.73	116.63	3661.73	90	0.26
14		0.5	235.55	183,034	3601.18	235.52	531,116	3603.63	235.75	3627.84	93	0.08
15		0.75	354.77	134,854	3601.84	354.73	275,613	3602.2	354.87	3620.45	114	0.02
16		1	474.0	0	0.6	474.0	0	0.28	474.0	0.5	2	0.0

From Table 3, for the sparse graphs, one can see that M_3 cannot solve any of the instances with a certificate of optimality with any of the algorithms since the CPU time values are larger than one hour. Again, the exception occurs for the graphs using $\alpha = 1$, which are easily solved with any of the methods. For M_4, it is not possible to ensure that the objective values correspond to the optimal solutions because they are not obtained in less than one hour. However, in the case of model M_4, recall that the objective values are upper bounds. Consequently, one can ensure that the graph instances having zero gaps report the optimal solutions. This is because these upper bounds are equal to the lower bounds obtained with one or more of the remaining models. This also ensures that the lower bounds obtained with the other models correspond to optimal solutions. For the complete graph instances in Table 4, one observes similar trends. One can see that only one graph instance is solved optimally in less than one hour. However, it is again mentioned that for model M_4 the graph instances having zero gaps report the optimal solutions. The argument is the same as for the sparse graphs presented in Table 3, i.e., the upper bounds are equal to the lower bounds.

To conclude, from the numerical results presented in Tables 1–4 one can mainly observe that model M_1 has a slightly better performance than the rest of the models when using the pure B&C approach of the CPLEX solver, at least for the reported instances.

To give more insights concerning the behavior of model M_1, in Figures 3 and 4 the impact of varying parameter $\alpha \in [0;1]$ for a sparse and a complete input graph, respectively, is plotted. For both figures, the number of covered users and active BSs is plotted while also varying the transmission radius between BSs and users from 200 ms to 300 ms. Each instance is composed of $n = 40$ BSs and $m = 300$ users. We mention that each point in these curves corresponds to an optimal solution to the problem.

From Figures 3 and 4, one can see that the higher the transmission radius between BSs and users is, the higher the number of covered users. Additionally, the lower the number of BSs required to connect the backbone network. In particular, in Figure 4, the number of active BSs is less than in Figure 3 since the backbone is fully connected. Finally, it is observed that the objective values increase with parameter α. In Tables 5 and 6, the numerical results obtained with Algorithm 2 for the sparse and complete graph instances, respectively, solved in the previous tables are reported. In particular, in Tables 5 and 6, the same column information is reported. More precisely, in columns 1–3, the instance number, the density of the graph, and the value of parameter α are reported. Next, in columns 4–10, the best objective value obtained in the previous tables, the best objective found with Algorithm 1, the CPU time in seconds required by the algorithm, its number of iterations, the number of attended or covered users in the network instance, and the number of BSs required to connect the backbone in the output solution of the problem are reported. Finally, the gaps in percentage that we compute by $\left[\frac{\text{the value of column 4-the value of column 5}}{\text{the value of column 4}}\right] * 100$ are reported. From these tables, one can observe that the solutions obtained by the proposed local search heuristic are near-optimal, with a worst gap obtained of 0.88% in the case of the sparse graphs and a worst gap of 3.03% for the complete graphs. It is further noted that the worst CPU time is lower than one minute for all the tested instances. One can also see that the larger the value of parameter α, the larger the number of BSs required to construct the backbone. Finally, it is observed that most of the users are covered for all tested instances. To further investigate the behavior of Algorithm 2, in Tables 7 and 8 numerical results obtained with M_1 versus Algorithm 2 for a large sparse connected and a fully connected graph instance, respectively, are compared. In these tables, the legend is the same. In columns 1–3, the instance number, the density of the graph, and the value of the parameter α are reported. Next, in columns 4–6, the best objective function value of model M_1, the number of branch and bound nodes, and its CPU time in seconds are presented. Subsequently, the best objective value obtained with Algorithm 2, its CPU time in seconds, the number of iterations of the algorithm, the number of covered users, the number of active BSs, and the gaps in percentage that we compute by $\left[\frac{\text{the value of column 4-the value of column 7}}{\text{the value of column 4}}\right] * 100$ are shown. From Tables 7 and 8, it is observed that M_1 requires more than one hour to obtain the optimal solution except for the instances using $\alpha = 1$. In this case, the problem becomes much more simple to solve optimally. Next, one can observe that the Algorithm 2 has an even better performance with a worst gap percentage of 0.21%. Finally, one can also see that most of the users are covered without using a large number of active BSs forming the backbone.

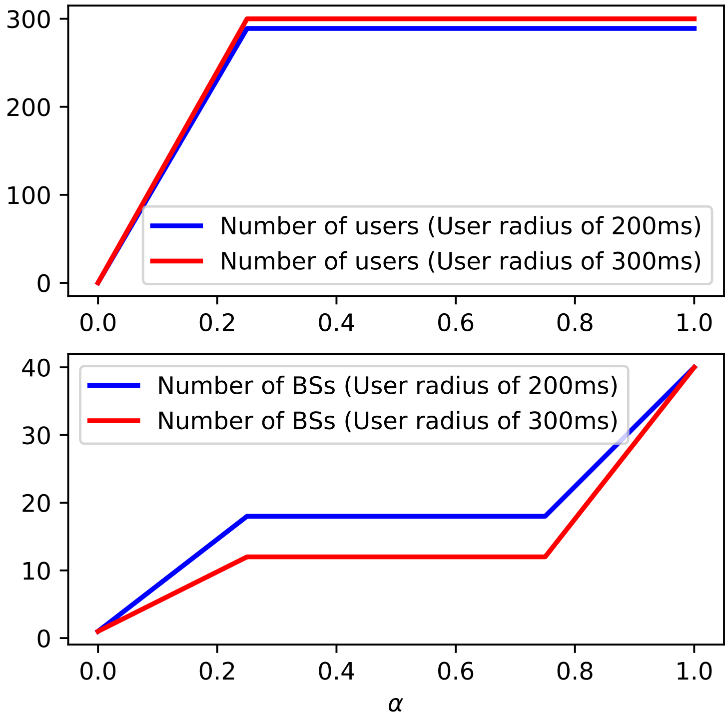

Figure 3. Impact of varying parameter $\alpha \in [0;1]$ for a sparse input graph with transmission radius between BSs of 300 m, and transmission radii between BSs and users of 200 m and 300 m. The instance is composed of $n = 40$ BSs and 300 users.

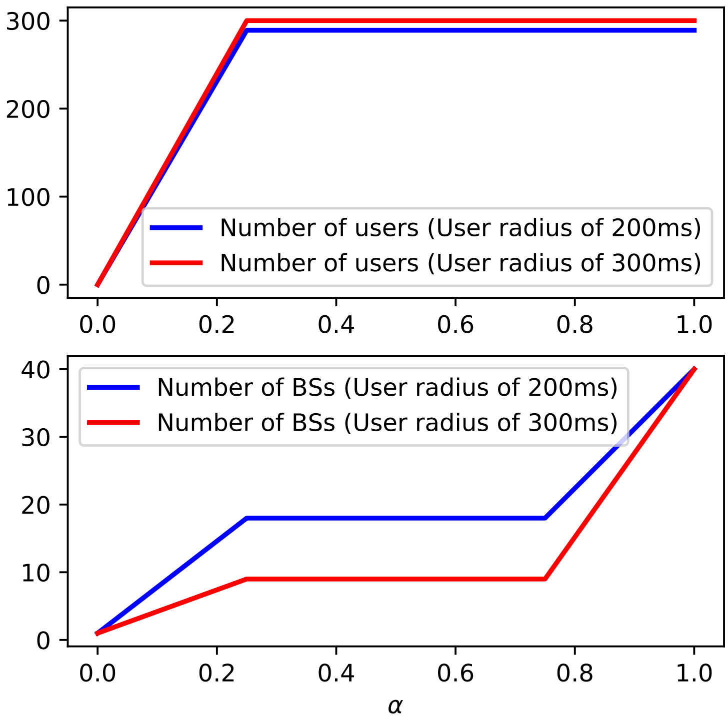

Figure 4. Impact of varying parameter $\alpha \in [0;1]$ for a complete input graph using transmission radii between BSs and users of 200 m and 300 m. The instance is composed of $n = 40$ BSs and 300 users.

Table 5. Numerical results obtained with Algorithm 2 for sparse connected graphs.

#	Density	α	Algorithm 2						
			Best	Best Alg.	CPU (s) Alg.	# Iter.	# of Users	# of BSs	Gap (%)
Using $n = 40$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs									
1	23.58	0.25	59.45	59.19	3.89	3258	247	23	0.42
2		0.5	121.96	121.79	4.03	3494	247	24	0.13
3		0.75	184.48	184.37	8.55	7059	247	24	0.05
4		1	247.0	246.0	8.29	2700	246	39	0.4
Using $n = 40$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs									
5	20.12	0.25	121.64	120.56	8.54	5271	493	28	0.88
6		0.5	246.43	245.24	12.13	7172	494	29	0.47
7		0.75	371.21	368.89	16.09	11,781	493	27	0.62
8		1	496.0	492.0	9.35	2672	492	40	0.8
Using $n = 60$, $m = 250$, a radius of 200 m for users, and a radius of 300 m for BSs									
9	21.41	0.25	59.63	59.38	5.51	5411	248	24	0.4
10		0.5	122.42	122.21	8.21	6970	248	26	0.16
11		0.75	185.21	185.13	10.26	8294	248	24	0.04
12		1	248.0	248.0	15.32	2501	248	60	0.0
Using $n = 60$, $m = 500$, a radius of 200 m for users, and a radius of 300 m for BSs									
13	21.58	0.25	116.32	115.91	25.16	12,553	473	24	0.35
14		0.5	235.55	235.44	23.24	11,981	474	24	0.04
15		0.75	354.77	353.22	20.04	7988	472	23	0.43
16		1	474.0	471.0	18.57	2519	471	58	0.63

Table 6. Numerical results obtained with Algorithm 2 for complete graphs.

#	Density	α	Algorithm 2						
			Best	Best Alg.	CPU (s) Alg.	# Iter.	# of Users	# of BSs	Gap (%)
Using $n = 40$, $m = 250$, and a radius of 200 m for users									
1	100	0.25	59.5	59.2	27.05	10,151	247	24	0.49
2		0.5	122.0	121.79	23.73	8555	247	26	0.16
3		0.75	184.5	183.64	20.24	7353	246	26	0.46
4		1	247.0	246.0	19.89	4246	246	37	0.4
Using $n = 40$, $m = 500$, and a radius of 200 m for users									
5	100	0.25	121.06	118.22	26.24	6389	484	28	2.33
6		0.5	245.37	241.15	53.37	12,623	486	28	1.71
7		0.75	369.68	363.59	41.86	9742	486	27	1.64
8		1	494.0	479.0	31.12	5317	479	39	3.03
Using $n = 60$, $m = 250$, and a radius of 200 m for users									
9	100	0.25	59.63	59.34	19.64	6439	248	26	0.48
10		0.5	122.4	122.22	22.21	8635	248	25	0.14
11		0.75	185.2	185.09	14.28	4935	248	26	0.05
12		1	248.0	248.0	26.08	2501	248	60	0.0
Using $n = 60$, $m = 500$, and a radius of 200 m for users									
13	100	0.25	116.32	115.92	32.09	8958	473	25	0.34
14		0.5	235.55	234.92	22.89	5601	473	27	0.26
15		0.75	354.77	353.99	36.44	9848	473	25	0.21
16		1	474.0	471.0	32.09	2692	471	60	0.63

Table 7. Numerical results obtained with M_1 vs. Algorithm 2 for a large sparse connected graph instance.

#	Density	α	M_1			Algorithm 2				
			Best	*B&Bn*	CPU (s)	Best Alg.	CPU (s) Alg.	# Iter.	# of Users	# of BSs	Gap (%)
Using $n = 100$, $m = 1000$, a radius of 200 ms between users and BSs, and a radius of 300 ms between BSs											
1	22.82	0.25	247.62	133,580	3603.95	247.12	65.84	11,070	1000	37	0.19
2		0.5	498.45	140,350	3603.35	498.09	55.57	9978	1000	36	0.07
3		0.75	749.2	129,126	3600.81	749.03	38.1	7327	1000	38	0.02
4		1	1000.0	0	1.28	999.0	48.47	2504	999	97	0.1

Table 8. Numerical results obtained with M_1 vs. Algorithm 2 for a complete graph instance.

#	Den.	α	M_1			Algorithm 2				
			Best	*B&Bn*	CPU (s)	Best Alg.	CPU (s) Alg.	# Iter.	# of Users	# of BSs	Gap (%)
Using $n = 100$, $m = 1000$, a radius of 200 ms between users and BSs, and a radius of 300 ms between BSs											
1	100	0.25	247.58	13,697	3600.23	247.05	52.84	6338	1000	36	0.21
2		0.5	498.38	8297	3600.26	498.15	77.99	10,610	1000	32	0.04
3		0.75	749.21	9792	3600.18	749.06	92.74	11,863	1000	35	0.01
4		1	1000.0	0	3.82	999.0	81.89	2507	999	97	0.1

To provide further insights concerning the behavior of Algorithm 2, in Figures 5 and 6 feasible solutions in terms of objective values for a sparsely and fully connected input network obtained with M_1 while varying α in the interval [0.25;0.9] are presented. In both figures, each network is composed of $n = 100$ nodes for the backbone and $m = 500$ users. In particular, in Figure 5 a radial transmission distance of 300 ms for the BSs is used and 200 ms for the distances between BSs and users. In Figure 6, the distances between BSs and users are set to 200 ms as well. Finally, we mention that both figures report objective values and velocities obtained with M_1 and Algorithm 2 when finding feasible solutions. The numerical results obtained in Figures 5 and 6 show that CPLEX finds very poor solutions at the beginning. More precisely, CPLEX takes at least 10 s to reach good quality solutions, whereas Algorithm 2 can find very good objective values in fractions of a second. After this time, it is observed that CPLEX and Algorithm 2 are both competitive in the sense that they reach similar solutions. Another observation is that one can ensure that the objective values are near-optimal ones. This can be ensured because the Mipgaps obtained with CPLEX are lower than 1% [7]. Finally, it is observed that varying the parameter α does not seem to have a significant effect on the performance of either M_1 or the Algorithm 2. However, note that the higher the value of the parameter α, the larger the number of users that can be covered.

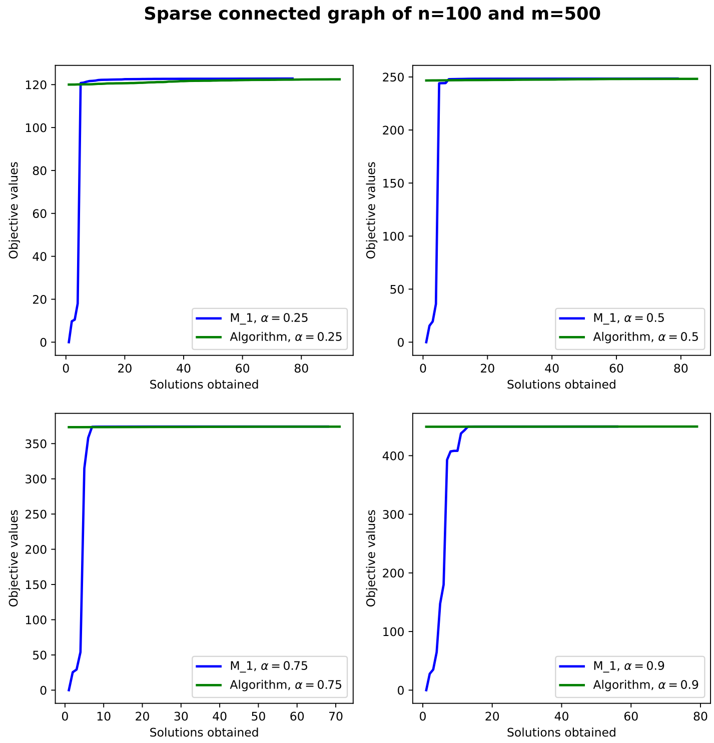

Figure 5. Feasible solutions obtained with M_1 and Algorithm 2 for a sparse connected input network while varying α in the interval $[0.25; 0.9]$.

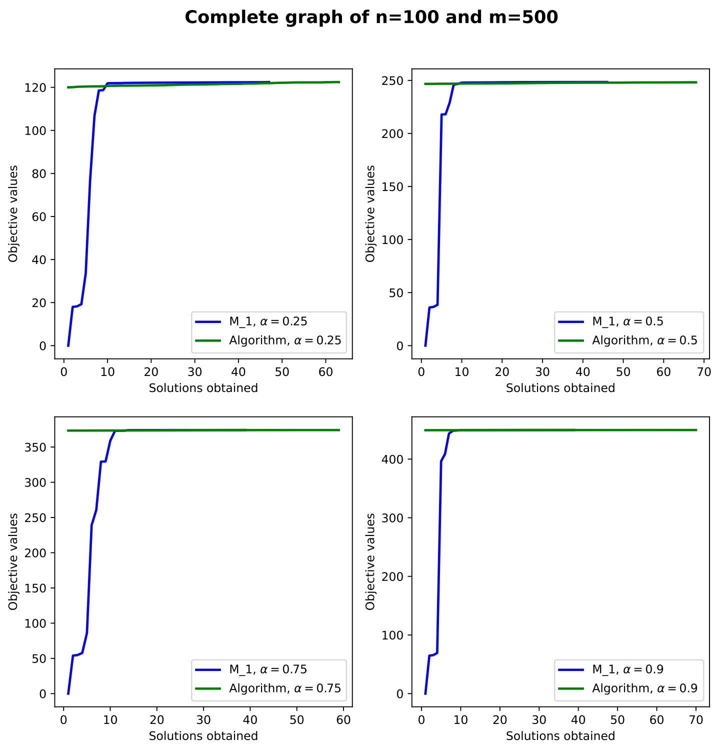

Figure 6. Feasible solutions obtained with M_1 and Algorithm 2 for a fully connected input network while varying α in the interval $[0.25; 0.9]$.

6. Conclusions

This article considers the problem of how to connect nodes in a network while covering as many users as possible at a minimum connectivity cost simultaneously for 5G networks.

It is well known that 5G technology is fast, but it is a bit tricky because it does not cover long distances well when using millimeter-wave technology. Consequently, we proposed new mathematical formulations to solve this problem from a management point of view. These models help us decide which nodes in the network should be connected to cover most of the users. Our models are solved with the branch and cut and Benders decomposition algorithms of the CPLEX solver to find the optimal solutions. We also proposed an exact iterative approach to solve the exponential model and proposed an approximation meta-heuristic algorithm to find good solutions faster. In our tests, we found that solving the problem perfectly is hard. We could only do it for cases with up to 40 nodes and 500 users. One of our models worked better in terms of time. On the other hand, our meta-heuristic showed that we could obtain tight solutions with less effort for both small and large graph instances using symmetric radial transmission distances. Finally, we mention that since our proposed exponential model is solved iteratively with our exact approach, we could obtain tight upper bounds. On the other hand, since the rest of the models obtain in the best case the optimal or feasible lower bounds, we provided a tight interval while ensuring that the optimal solutions lie inside that interval. Ultimately, since there are no models proposed in the literature covering both aspects at the same time, there are no benchmark instances to compare with yet in the literature. Thus, we believe that these models will enrich the literature to develop further modeling aspects of the problem and will allow for benchmark comparisons.

In future research, new avenues to be considered for the network planning problem are suggested as follows. First, we mention that the proposed models can include nonlinear formulas such as capacity and interference which can potentially lead to more difficult formulations to deal with. Furthermore, if the arising nonlinearities are nonconvex, there would be no certification of optimality. Thus, this constitutes a challenging task to perform in future research. Finally, another research direction is to consider exploring urban areas, even though this would require realistic data from particular cities while also considering non-line-of-sight aspects when using symmetric and asymmetric radial transmission distance costs as well as new mathematical models and algorithms to solve this relevant network planning problem from a management point of view.

Author Contributions: Conceptualization, P.A., A.D.F. and S.C.; methodology, P.A., A.D.F. and S.C.; software, P.A., A.D.F. and S.C.; validation, P.A. and A.D.F.; formal analysis, P.A., A.D.F. and S.C.; investigation, P.A., A.D.F. and S.C.; resources, P.A. and A.D.F.; data curation, P.A., A.D.F. and S.C.; writing—original draft preparation, P.A., A.D.F. and S.C.; writing—review and editing, P.A. and A.D.F.; visualization, P.A., A.D.F. and S.C.; supervision, P.A.; project administration, P.A. and A.D.F.; funding acquisition, P.A. and A.D.F. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the financial support from Projects Dicyt 062313AS, ANID/FONDECYT Iniciación No. 11230129, and the Competition for Research Regular Projects, year 2021, code LPR21-02; Universidad Tecnológica Metropolitana.

Data Availability Statement: Dataset available on request from the authors.: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors acknowledge the support of the Vicerrectoría de Investigación, Innovación y Creación (VRIIC) of the Universidad de Santiago de Chile.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. IEEE 5G and Beyond Technology Road Map, 2020. Available online: https://futurenetworks.ieee.org/images/files/pdf/ieee-5g-roadmap-white-paper.pdf (accessed on 30 May 2023).
2. Palizban, N. Millimeter Wave Small Cell Network Planning for Outdoor Line-of-Sight Coverage Millimeter Wave Small Cell Network Planning for Outdoor. Ph.D. Thesis, Carleton University, Ottawa, ON, Canada, 2017.
3. Rappaport, T.S.; Sun, S.; Mayzus, R.; Zhao, H.; Azar, Y.; Wang, K.; Wong, G.N.; Schulz, J.K.; Samimi, M.; Gutierrez, F. Millimeter wave mobile communications for 5G cellular: It will work! *IEEE Access* **2013**, *1*, 335–349. [CrossRef]

4. Rakuten , Reimagining the End-to-End Mobile Network in the 5G Era. 2019. Available online: https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/elastic-services-controller-esc/reimagining-mobile-network-white-paper.pdf (accessed on 30 May 2023).
5. Niu, Y.; Li, Y.; Jin, D.; Su, L.; Vasilakos, A.V. A survey of millimeter wave communications (mmWave) for 5G: Opportunities and challenges. *Wirel. Netw.* **2015**, *21*, 2657–2676. [CrossRef]
6. Adasme, P. p-Median based formulations with backbone facility locations. *Appl. Soft Comput.* **2018**, *67*, 261–275. [CrossRef]
7. IBM ILOG, CPLEX High-Performance Mathematical Programming Engine, V20.1.0, (2020). Available online: http://www.ibm.com/software/integration/optimization/cplex/ (accessed on 10 June 2023).
8. Rahmaniani, R.; Crainic, T.G.; Gendreau, M.; Rei, W. The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* **2017**, *259*, 801–817. [CrossRef]
9. Adasme, P.; Cordero, S.; Soto, I.; Firoozabadi, A.D. New Formulations for Optimal User Coverage using 5G Millimeter-Wave-based Networks. In Proceedings of the 2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2022; pp. 732–737.
10. Rezaei, A.; Mascheroni, A.; Stevens, M.C.; Argha, R.; Papandrea, M.; Puiatti, A.; Lovell, N.H. Unobtrusive Human Fall Detection System Using mmWave Radar and Data Driven Methods. *IEEE Sensors J.* **2023**, *23*, 7968–7976. [CrossRef]
11. Anjinappa, C.K.; Erden, F.; Güvenç, I. Base Station and Passive Reflectors Placement for Urban mmWave Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 3525–3539. [CrossRef]
12. Gu, Z.; Lu, H.; Zhang, M.; Sun, H.; Chen, C.W. Association and Caching in Relay-Assisted mmWave Networks: A Stochastic Geometry Perspective. *IEEE Trans. Wirel. Commun.* **2021**, *12*, 8316–8332. [CrossRef]
13. Szyszkowicz, S.S.; Lou, A.; Yanikomeroglu, H. Automated Placement of Individual Millimeter-Wave Wall-Mounted Base Stations for Line-of-Sight Coverage of Outdoor Urban Areas. *IEEE Wirel. Commun. Lett.* **2016**, *5*, 316–319. [CrossRef]
14. Sandoval, C.; Adasme, P.; Juan, E.S.; Soto, I.; Firoozabadi, A.D. Novel p-Median-based Formulations for Maximizing Coverage in 5G/6G Wireless Networks. In Proceedings of the 2021 Third South American Colloquium on Visible Light Communications (SACVLC), Toledo, Brazil, 11–12 November 2021; pp. 1–5. [CrossRef]
15. Koutsaftis, A.; Kumar, R.; Liu, P.; Panwar, S.S. Fast Inter-Base Station Ring (FIBR): A New Millimeter Wave Cellular Network Architecture. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2699–2714. [CrossRef]
16. Sabzehali, J.; Shah, V.K.; Dhillon, H.S.; Reed, J.H. 3D Placement and Orientation of mmWave-Based UAVs for Guaranteed LoS Coverage. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1662–1666. [CrossRef]
17. Fortet, R. Applications de lálgebre de boole en recherche operationelle. *Rev. Fr. Rech. Oper.* **1960**, *4*, 17–26.
18. Letchford, A.N.; Nasiri, S.D. Dirk Oliver Theis: Compact formulations of the Steiner Traveling Salesman Problem and related problems. *Eur. J. Oper. Res.* **2013**, *228*, 83–92. [CrossRef]
19. Adasme, P.; Andrade, R.; Letournel, M.; Lisser, A. Stochastic maximum weight forest problem. *Networks* **2015**, *65*, 289–305. [CrossRef]
20. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*; MIT Press and McGraw-Hill: Cumberland, RI, USA, 2009.