



# A 5G Hubs Location Hierarchized Problem that Balances the Connection of the Users

José-Fernando Camacho-Vallejo<sup>1</sup> · Juan-Carlos García-Vélez<sup>1</sup> · Carlos Corpus<sup>1</sup>

Accepted: 23 May 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

In this paper, a telecommunication network situation is studied. Particularly, the problem considers an upper-level decision maker that enables different types of hubs in the network and another one that connects the users to the enabled hubs. The objective of the upper-level decision maker is to balance the number of users connected to the enabled hubs. However, the connection of users is performed by another decision maker with lower hierarchy, the one associated to the lower-level problem. Hence, decisions at the upper level are given as parameters to the lower level. So that, the lower level is solved to obtain feasible bi-level solutions. In this case, the rational reaction of the decision-maker assigned to the lower level is obtained. Due to the complexity of this problem, an evolutionary algorithm is proposed to obtain a good approximation. Numerical experimentation shows the efficiency and robustness of the proposed solution scheme. Based on the results, it can be concluded that the proposed algorithm provides good alternatives for the bi-level decision-making process. Also, some managerial insights are given regarding the different types of enabled hubs and the connection of the users.

**Keywords** 5G network · Bi-level programming · Nested evolutionary algorithm · Hub location problem

## 1 Introduction

Telecommunication networks have taken an important role in the recent years. Nowadays, it is impossible to imagine the world in which computers, cell phones, automobiles, and electrical appliances are not connected in some way to those networks. Due to telecommunication networks development the creation of new technologies, such as 5G connectivity has greatly contributed to the manner in which the world is perceived at the moment [1, 2].

As a consequence of the lockdown caused by the COVID-19 pandemic, people have the necessity to be uninterruptedly connected to telecommunication networks. This led to remarkable increase in the connectivity requirements

necessary to carry out daily activities. Also, new opportunities for telecommunications services have arisen.

The development of 5G technology has facilitated communication like never before. That is why telecommunication companies placed more emphasis on adapting existing networks to fulfill the new necessities. Obviously, companies that provide a service need to meet the necessities of their customers. For example, in an office context, connections between computers are essential to perform daily tasks. To achieve it, the use of hubs plays an important role as they help to improve the distribution of the information through the network. Hubs are specialized hardware that are responsible for receiving telecommunications signals and redistributing them to their destination within the network.

Therefore, the location and installation of different hubs is important for user connectivity. Hub location theory deals with the situation described above. The seminal research is attributed to [3], where the problem for locating hubs and assigning the demand nodes to them is proposed. The objective is to minimize the cost associated to the shipment of commodities through origin-destination pairs. The main advances appeared in the 80's and 90's, [4–7], and [8] laid the foundations of the main features in terms of modeling and solution methods.

---

✉ Carlos Corpus  
carlos.corpuscd@uanl.edu.mx

José-Fernando Camacho-Vallejo  
jose.camachovl@uanl.edu.mx

Juan-Carlos García-Vélez  
jgarcia@uanl.edu.mx

<sup>1</sup> Facultad de Ciencias Físico Matemáticas, Universidad Autónoma de Nuevo León, Monterrey, Mexico

Hub Location Problems (HLPs) have been studied in diverse contexts, such as, airports [9], airlines [10], transportation [11, 12], and telecommunication systems. Additionally, these problems are classified as NP-hard [13]. The latter implies that for large-size problems, the use of heuristics and metaheuristics is a convenient alternative to solve them.

Particularly, in the context of telecommunication systems, related research is discussed next. In [14], a comparison between Iterated Local Search and Tabu Search is conducted. The main differences and their impact on the problem under study is discussed. Also, in [15], a HLP in which a network with a star topology is proposed. That problem is applied to an optical fiber context. The main objective of that problem is to locate central offices and the demand nodes to splitters, while minimizing the installation costs of optical fiber and hubs. To solve the problem, a Differential Evolution method and a Genetic Algorithm are implemented to compare their results. A hybrid algorithm to solve a case-study devoted to an optical fiber network for a hub network design is presented in [16].

New and novel applications using HLPs have arisen in many areas. In [17], a variant of HLP applied in freight transport is studied. On the other hand, a model for a multi-objective HLP under a context of a transportation case-study in Iran is proposed in [18]. In [19], a multimodal competitive HLP with loyalty and elastic demand using a postal service in Iran is analyzed. A novel consideration of transportation systems under a humanitarian logistics context is studied in [20], in which a hierarchical HLP that aims to integrate the urban and rural transport systems is formulated. Also, in [21] a multiperiod HLP is used to distribute humanitarian aids in Lebanon. As it can be seen, innovative and ingenious applications of HLPs have been addressed to study situations in different contexts. Recently, there are some references devoted to hub location in a context of COVID-19. Most of these papers are related to the location of medical centers, but in the *new normality*, 5G telecommunication networks play a key role. Therefore, this research contributes in a significant manner to the hub location field.

In this research, a problem that considers two hierarchized decision-making levels is considered. At the upper level, potential sites for enabling hubs of different types are considered. Some of these hubs are enabled to provide service to a set of users aiming to balance their workload. Under this approach, it is intended that some hubs do not operate at their maximum capacity and others are being underutilized. In the first case, a technical failure in the hub affects many network users; while in the second case, an available resource that may help to distribute the flow within the network is wasted. In addition, the entire network must be balanced disregarding the different types of hubs. In other words, a small-size hub with few users connected can be saturated, while a large-size hub with few users may be

highly underutilized. It is important to emphasize that the connection of the users is not decided at the upper level, but it is done at the lower level. Then, once the hubs are enabled, users are connected based on their connection cost.

Obviously, in this network of hubs there are some characteristics that must be taken into account when enabling the hubs and connecting users. For example, an enabled hub can only be of a specific type and a pre-defined budget is considered to establish the network of hubs. Also, each specific type of hubs has a set capacity regarding the number of users connected to it. Additionally, users must be connected only to one hub, and there is a minimum number of users that must be connected to enabled hubs in order to have a functional -in practice- network.

It is evident that the connection made at the lower level determines the balance of users, which is the aim of the upper level. Due to these characteristics, the situation under study can be modeled using a bi-level programming approach. The characteristics of the problem (a non-convex lower level) prevent the reformulation of the bi-level problem into a single-level one. Therefore, we propose an evolutionary metaheuristic algorithm to find good-quality solutions of the bi-level problem in an acceptable computational time.

The contributions of this research can be summarized as: (a) proposing a new bi-level programming model for a problem of enabling different type of hubs considering the workload balance between them, but connecting the users based on a rental cost; (b) designing an evolutionary algorithm to solve the bi-level problem in an efficient way; (c) performing a sensitivity analysis on some parameters of the problem to provide some interesting managerial insights.

The remainder of this research is as follows: Section 1 presents an introduction of the problem, its importance and current context; Section 2 defines the sets, parameters and decision variables of the problem and presents its appropriate mathematical formulation; the proposed algorithm is described in detail in Section 3; the computational experimentation is shown in Section 4, this includes the validation of the proposed algorithm as a solution methodology and a sensitivity analysis. Finally, Section 5 completes the article with the conclusions obtained from this research and some future research directions derived from our study are included.

## 2 Mathematical formulation

Based on the situation under study described in the previous section, the sets, parameters and decision variables involved in the problem are going to be defined to formulate a mathematical model. Let  $I$  be the set of potential sites to enable

a hub, let  $J$  be the set of different types of hubs and let  $K$  be the set of users that intend to connect to the enabled hubs.

Consider a fixed installation cost  $f_{ij}$  to enable a hub of type  $j \in J$  in the site  $i \in I$ . Also, a rental cost  $r_{ijk}$  for user  $k \in K$  for using the type of hub  $j \in J$  in the site  $i \in I$  is associated. A limited budget  $b$  for the network design and its implementation is considered, and a capacity  $q_j$  that limits the users connectivity allowed in the hub of type  $j \in J$  is placed. Moreover, a minimum number of hubs of type

$j \in J$  must be enabled, which is denoted by  $p_j$ . In order to implement a functional network of hubs, a minimum number of users must be connected, which is denoted by  $\rho_{min}$ .

The upper-level decision variables are:

$$y_{ij} = \begin{cases} 1 & \text{if the hub of type } j \in J \text{ is enabled at site } i \in I. \\ 0 & \text{otherwise} \end{cases}$$

On the other hand, the decision variables associated to the lower level are:

$$x_{ijk} = \begin{cases} 1 & \text{if user } k \in K \text{ is connected to the hub of type } j \in J \text{ enabled at potential site } i \in I. \\ 0 & \text{otherwise} \end{cases}$$

Additionally, auxiliary continuous variables  $l_j$  and  $u_j$  denote the minimum and maximum number of users connected to hubs of type  $j \in J$ , respectively.

The resulting bi-level programming model is as follows:

$$\min_{y,x} \max_{j \in J} \{u_j - l_j\} \tag{1}$$

$$s.t. \quad \sum_{i \in I} \sum_{j \in J} f_{ij} y_{ij} \leq b \tag{2}$$

$$\sum_{i \in I} y_{ij} \geq p_j, \quad \forall j \in J \tag{3}$$

$$\sum_{j \in J} y_{ij} \leq 1, \quad \forall i \in I \tag{4}$$

$$l_j \leq \sum_{i \in I} \sum_{k \in K} x_{ijk} + |K|(1 - \sum_{i \in I} y_{ij}), \quad \forall j \in J \tag{5}$$

$$u_j \geq \sum_{i \in I} \sum_{k \in K} x_{ijk}, \quad \forall j \in J \tag{6}$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K \tag{7}$$

where for a fixed  $y$ , variable  $x$  must be the optimal solution of the following problem:

$$\min_x \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} r_{ijk} x_{ijk} \tag{8}$$

$$s.t. \quad \sum_{k \in K} x_{ijk} \leq q_j y_{ij} \quad \forall i \in I, j \in J \tag{9}$$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} x_{ijk} \geq |K| \rho_{min} \tag{10}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \leq 1 \quad \forall k \in K \tag{11}$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in I, j \in J, k \in K \tag{12}$$

The upper-level problem is defined by Eqs. 1-7 and aims to balance the number of users connected to the enabled hubs. Particularly, the balance is expected for each type of hubs, and consequently, a balanced network of hubs is obtained. That is indicated in Eq. 1, in which the minimum of the maximum unbalanced hubs is aimed. Constraint (2) imposes the available budget to design the network of hubs. Constraint (3) ensures that a minimum number of each type of hubs must be enabled and constraint (4) indicates that a single hub of any type may be enabled at each potential site. The minimum and maximum number of users connected to a hub is computed in Eqs. 5 and 6, respectively. The binary nature of upper-level decision variables is indicated in Eq. 7.

The lower level problem defined by Eqs. 8-12 is parameterized in a upper-level decision. Its objective function aims to minimize the rental cost for connecting users to the different enable hubs, and it is given by Eq. 8. Constraint (9) indicates the capacity of specific hubs, in terms of the users connected to them. A minimum number of connected users to guarantee the functionality of the network of hubs is ensured by Eq. 10. Additionally, users must be connected to a single enabled hub of any type, this is ensured by 11. Finally, the binary nature of lower-level decision variables is indicated in Eq. 12.

In order to have well-defined a bi-level programming problem, the lower level must have a unique optimal solution for an upper-level decision. In the other case, that is, when there are multiple lower-level optimal solutions, the optimistic or pessimistic approach could be assumed [22]. The former consists of assuming a cooperative approach. In other words, from all the multiple optimal solutions of the lower level, the one that suits better for the upper-level objective function is chosen. The reasoning behind this approach is that all the optimal solutions of the lower level yield to the

minimum rental cost, but the solution that results in the more balanced network is selected. The latter assumes the opposite behavior, that is, the upper-level decision maker prepares for the worst scenario due to the reaction of the lower-level one. In this paper, the optimistic approach is assumed.

### 3 A nested evolutionary algorithm

Bi-level problems are classified as NP-hard, even in their simplest form, that is, the linear-linear case. Additionally, in the problem herein proposed, the lower level is non-convex. Hence, a solution scheme based on the single-level reformulation is not possible. Based on these two main drawbacks, an evolutionary algorithm is proposed to solve the 5G hubs location bi-level problem.

Evolutionary algorithms (EAs) have been considered through the years as an efficient method for approximating solutions to hard-to-solve problems [23–25]. EAs are inspired in the biological evolution of species. Initial population of individuals (solutions) is considered, and their corresponding fitness (objective function value) is measured. Then, evolution operators, such as selection, crossover, and mutation are applied in order to accomplish an evolution in the individuals. Individuals with better fitness survive and continue the evolutionary process through a predefined number of generations (iterations).

Particularly, EAs have demonstrated to be effective for solving complex problems in different location contexts. For instance, EAs to solve facility location problems can be found in [26–28]. Also, EAs have been developed to solve bi-level problems [29–31]. Recall that in bi-level problems, the lower level must be optimally solved for a given upper-level decision. That is the so-called nested approach and it is used for obtaining bi-level feasible solutions. A general scheme of an EA is presented.

Next, a detailed explanation of each one of the components involved in the proposed Nested Evolutionary Algorithm (NEA) is presented. As it can be noticed from the pseudocode shown in Algorithm 1, the algorithm creates an initial population of individuals, and a repairing phase is applied to ensure that the upper level constraints are met. In order to achieve bi-level feasible solutions, the lower level is optimally solved for each individual. Then, the fitness evaluation of individuals is performed. A subset of the individuals with better fitness is selected to apply the evolutionary operators, such as, crossover and mutation. The latter operators create new individuals and the process is repeated. The aim is of this evolutionary scheme is to improve individuals over the generations.

#### 3.1 Initial population

As mentioned above, the initial population of individuals is created in a random manner. To achieve this, vectors of size  $|I|$ , which components are elements of the set  $J \cup \{0\}$ , are considered. Therefore, a vector contains integer numbers indicating the type of hub enable in the  $i$ -th potential site. In case that a zero appears, it implies that no hubs are enabled in that potential site.

An example of the encoding herein utilized is presented in Fig. 1, in which 8 potential sites and 3 different types of hubs are considered. The illustrated individual indicates that two hubs of type 1 are enabled in the second and eighth potential sites. Similarly, hubs of types 2 and 3 are enabled in the third and sixth potential sites, respectively. Also, note that there are four potential sites, in which no hubs are enabled.

Despite the fact that the considered encoding guarantees that a potential site may be used to enable only one type of

**Algorithm 1:** Nested Evolutionary Algorithm

---

```

1 Procedure Evolutionary Algorithm
2   Create an initial random population of individuals
3   Repairing phase
4   Lower-level solution
5   Fitness evaluation of the individuals
6   while stopping criteria is not met do
7     Selection phase
8     Crossover phase
9     Repairing phase
10    Mutation phase
11    Lower-level solution
12    Fitness evaluation of the individuals
13    Update the population in an elitist manner
14  end
15 end

```

---



Fig. 1 Illustration of the individual’s encoding

hubs, the other constraints may not be met. The enabled hubs of each type may not surpass their minimum required number or the budget may be exceeded. Therefore, a repairing phase is needed.

### 3.2 Repairing phase

Once an individual is created, its feasibility must be guaranteed. If the individual is feasible, then it is included in the population. In other case, the individual suffers some structured changes aiming to achieve feasibility. As mentioned above, the infeasibility may be caused by either the minimum number of enabled hubs is not met or the budget is exceeded.

Let us illustrate the proposed repairing phase with an example. Consider the individual depicted in Fig. 1, and assume that the minimum number of enable hubs of type 2 and 3 are two ( $p_2 = p_3 = 2$ ). It is evident that the individual does not meet with that specific constraint. Hence, a potential site without a hub is randomly selected to enable the specific type of hub. Selection of the type of hubs is performed lexicographically, that is, firstly the hub of type 2 is enabled, and secondly, the hub of type 3. An illustration is shown in Fig. 2

It is important to mention that during the process of enabling missing hubs is done, budget is checked. If the budget is exceeded and the minimum required hubs have not been enabled, the individual is discarded.

### 3.3 Optimal solution of the lower level

In order to evaluate the fitness of an individual, the connection of the users to the enabled hubs is needed. Recall that the connection is decided at the lower level, in which the connection costs are minimized.

In this case, once the different types of enabled hubs are known, the resulting problem at the lower level must be optimally solved. Particularly, it corresponds to a binary programming problem, which can be solved by a commercial optimization software.



Fig. 2 Illustration of a repaired individual

If the resulting lower level problem is infeasible, the associated individual is discarded. The latter may be caused by the lack of capacity to satisfy the network functionality constraint.

### 3.4 Fitness evaluation

The fitness associated to an individual is given by the objective function of the upper level. In other words, the balance of the users connected to the enabled hubs.

It is evident that the fitness evaluation can be done after the lower level is solved.

### 3.5 Selection

Selection of individuals that enter to the crossover phase is performed in an elitist manner. A subset of individuals with better fitness is selected.

### 3.6 Crossover

To generate new individuals the crossover phase is applied. Individuals in the elite subset are randomly paired with another individual of the population. For each pair of individuals (parents), a single-point crossover is performed to generate two offspring. The first offspring has the characteristics of the first parent before the crossover point and the characteristics of the second parent after the crossover point. Conversely, the second offspring is created. An example is illustrated in Fig. 3.

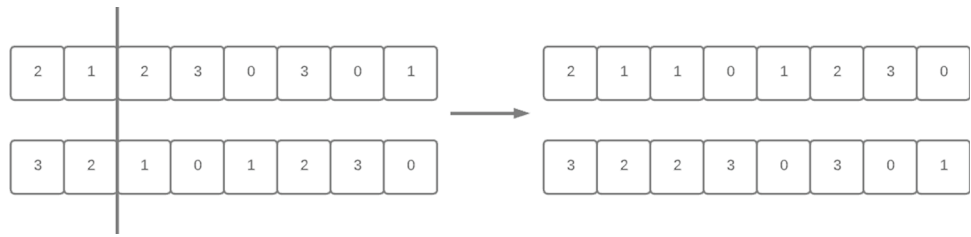
The implemented crossover may create infeasible individuals. Hence, the repairing phase previously described is applied. If offspring are feasible, then they are stored; otherwise, they are discarded.

### 3.7 Mutation

The new created individuals in the crossover phase may suffer a mutation. This phase adds diversity to the EA and promotes a wide exploration of the search space. The proposed mutation consists of random selection of two components of an individual, and interchange between them. An example is depicted in Fig. 4, in which the second component of the individual is interchanged with the penultimate one.

Note that the mutation implemented maintains feasibility regarding the minimum number enabled hubs of each type. But the budget may be exceeded. In this case, another interchange between components is explored to perform the mutation. The process stops when a successful mutation is achieved. In the case when all the possible interchanges have been explored and no feasible individual is obtained, the mutation is not applied and the individual is included in the new population.

**Fig. 3** Illustration of the single-point crossover



### 3.8 Update the population

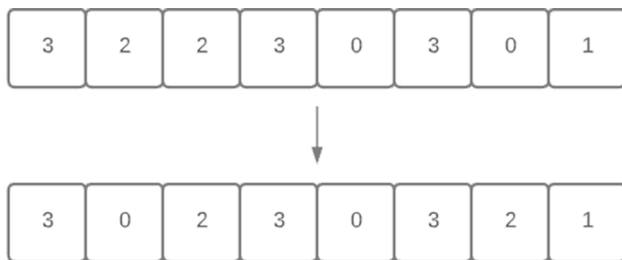
At this point, we have the original population of individuals and the population of new created ones. Both populations are merged in an elitist manner. That is, the individuals with better fitness value are included in the population that replaces the previous one.

By doing the latter, we maintain the best individuals that have been already created, and include the recently created ones to improve the fitness of the population. Despite this elitist scheme, the evolution process is not biased to local optima, since diversity is given by the mutation phase.

The evolutionary process is repeated until a predefined number of generations is reached, that is the stopping criterion considered in the proposed NEA.

## 4 Computational experimentation

Since there are no previous studies regarding the herein considered problem, no benchmark instances exist. Therefore, to evaluate the performance of the proposed NEA, a random but controlled instance generator is developed to create two main sets of different test instances. Sizes of the instances are shown in Table 1 that is, 50 potential sites and 50 users, and 75 potential sites and 100 users. Each set can be divided into two subsets, in which the minimum connectivity of the network is varied and different number of different hubs must be enabled. A total of 60 instances is created to carry out the computational experimentation.



**Fig. 4** Illustration of the mutation

### 4.1 Parameters of the NEA

In order to deal with the stochastic nature of the proposed algorithm, many executions of the NEA for the same instance are performed. Moreover, different parameters are involved in the NEA, such as, size of the population, number of tournaments in the crossover, probability of entering into crossover, and probability of entering into mutation. Additionally, the stopping criteria involve two parameters, that is, the total number of generations and the number of consecutive generations without improving the incumbent individual.

Although there are studies that indicate that evolutionary algorithms converge despite the calibration of the parameters [32], we decided to investigate the NEA performance under different parameters configurations.

Extensive preliminary tests are performed to identify the most-suitable parameters of the NEA when solving the problem herein studied. Hence, we fix the size of the population to 100 individuals. The selection of the individuals is performed after 5 tournaments have been completed. The half of the best-ranked individuals enter into the crossover with a probability of 0.65. Then, offspring enter into the mutation phase with a probability of 0.2. The maximum number of generations is set to 100. But, if the incumbent fitness does not improve in 10 consecutive generations, the NEA stops.

**Table 1** Characteristics of the created instances

Potential sites	Users	Types of hubs	$p_{min}$	$P_j$
50	50	3	0.5	[5, 3, 2]
				[2, 3, 5]
				[2, 6, 2]
50	50	3	0.75	[5, 3, 2]
				[2, 3, 5]
				[2, 6, 2]
75	100	3	0.5	[5, 5, 5]
				[7, 5, 3]
				[4, 4, 7]
			0.75	[5, 5, 5]
				[7, 5, 3]
				[4, 4, 7]

Preliminary results indicated that the value of the parameters involved in the NEA play a key role in the efficiency of the proposed algorithm. It is worthy to mention that the selected parameters are in the range indicated in the conclusions found in [32], which indicates that they may work adequately.

### 4.2 Numerical results

NEA is implemented in the programming language of FICO Xpress Mosel. Also, the lower-level problem is solved by using the FICO Xpress Solver 8.9. All the computational experimentation is carried out on a workstation using an Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz with 16 GB of RAM under Windows 11 Professional operating system. As mentioned before, due to the stochasticity involved in the NEA, 10 executions are performed for each instance.

The obtained results from the computational experimentation are shown in Tables 2 and 3. The label

$ww \times xxx.yy.zz$  of each instance indicates the number of potential sites ( $ww$ ), the number of users ( $xxx$ ), the set of hubs of each type that must be enabled ( $yy$ ), and the number of instance with these specific characteristics ( $zz$ ). In the column labeled  $F_{Best}$ , the best value obtained for the upper-level objective function is reported. To measure the dispersion of the obtained values, the following are measured: the average ( $F_{Avg}$ ) and the worst ( $F_{Worst}$ ) objective function. The number of runs that the NEA reached the best objective value is included in # Best. Then, the required average time (in seconds) is included in  $Time_{avg}$  column. Moreover, the number of additionally enabled hubs than the minimum required is shown in the next three columns. For example, in Extra\_min is indicated the greater number of extra hubs enabled in that particular instance. Correspondingly, Extra\_avg and Extra\_max contain the average and minimum number of extra hubs enabled. Finally, the total number of users connected is shown in column Connected.

**Table 2** Results for the instances of size 50x50

Instance	$F_{best}$	$F_{avg}$	$F_{worst}$	# Best	$Time_{avg}$	Extra <sub>min</sub>	Extra <sub>avg</sub>	Extra <sub>max</sub>	Connected
50x50.B1.01	0	0.2	1	8	41.989	0	0.1	1	25
50x50.B1.02	0	0.1	1	9	41.904	0	0.2	1	25
50x50.B1.03	0	0.3	1	7	48.398	0	0.5	1	25
50x50.B1.04	0	0	0	10	43.643	0	0	0	25
50x50.B1.05	0	0.2	1	8	32.085	0	0	0	25
50x50.B1.06	0	0.3	1	7	41.264	0	0.3	1	38
50x50.B1.07	0	0.1	1	9	45.195	0	0.1	1	38
50x50.B1.08	0	0.2	2	9	52.362	0	0.3	1	38
50x50.B1.09	1	1	1	10	44.962	0	0.2	1	38
50x50.B1.10	1	1.2	2	8	55.133	0	0	0	38
50x50.B2.01	1	1.1	2	9	46.855	0	0.4	1	25
50x50.B2.02	1	1.3	2	7	70.895	0	0.1	1	25
50x50.B2.03	1	1.1	2	9	58.995	0	0.5	1	25
50x50.B2.04	1	1.2	2	8	45.21	0	0.1	1	25
50x50.B2.05	1	1.4	2	6	48.117	0	0	0	25
50x50.B2.06	1	1.6	2	4	43.755	0	0.1	1	38
50x50.B2.07	0	0.3	2	8	64.05	0	0.2	1	38
50x50.B2.08	1	1.3	3	8	64.119	0	0.4	1	38
50x50.B2.09	1	1.4	3	7	52.125	0	0	0	38
50x50.B2.10	1	1.3	3	8	43.506	0	0.1	1	38
50x50.B3.01	1	1	1	10	50.11	0	0.5	1	25
50x50.B3.02	1	1	1	10	40.364	0	0	0	25
50x50.B3.03	0	0.3	2	8	47.084	0	0.6	1	25
50x50.B3.04	1	1.2	2	8	41.369	0	0.1	1	25
50x50.B3.05	0	0	0	10	47.321	0	0.1	1	25
50x50.B3.06	1	1.4	2	6	52.724	0	0.2	1	38
50x50.B3.07	0	0.4	2	7	49.168	0	0	0	38
50x50.B3.08	1	1.1	2	9	51.313	0	0.2	1	38
50x50.B3.09	1	1.3	3	8	44.56	0	0.1	1	38
50x50.B3.10	1	1.5	3	7	64.153	0	0	0	38

**Table 3** Results for the instances of size 75×100

Instance	$F_{best}$	$F_{avg}$	$F_{worst}$	# Best	Time <sub>avg</sub>	Extra <sub>min</sub>	Extra <sub>avg</sub>	Extra <sub>max</sub>	Connected
75×100.B1.01	2	2.2	3	8	121.443	0	0.7	1	50
75×100.B1.02	1	1.3	3	8	97.68	0	0.4	1	50
75×100.B1.03	1	1.2	3	9	93.981	0	0.6	2	50
75×100.B1.04	2	2.3	3	7	106.294	0	0.6	1	50
75×100.B1.05	2	2.1	3	9	101.687	0	0.3	1	50
75×100.B1.06	2	2.2	3	8	93.785	0	0.6	2	75
75×100.B1.07	0	0.6	3	7	84.878	0	0.1	1	75
75×100.B1.08	2	2.1	3	9	98.925	0	0.3	1	75
75×100.B1.09	2	2.4	4	7	98.58	0	0.4	2	75
75×100.B1.10	2	2.2	3	8	127.042	0	0.7	2	75
75×100.B2.01	2	2.1	3	9	106.787	0	0.6	2	50
75×100.B2.02	1	1.4	3	7	87.581	0	0	0	50
75×100.B2.03	2	2	2	10	88.137	0	0.4	1	50
75×100.B2.04	1	1.3	3	8	103.928	0	1.2	2	50
75×100.B2.05	2	2.1	3	9	114.68	0	0.6	1	50
75×100.B2.06	1	1.3	3	8	96.151	0	0.3	1	75
75×100.B2.07	1	1.2	3	9	71.526	0	0	0	75
75×100.B2.08	2	2.3	3	7	86.256	0	0.3	1	75
75×100.B2.09	2	2	2	10	97.411	0	0.5	1	75
75×100.B2.10	1	1.4	3	8	88.809	0	0.4	1	75
75×100.B3.01	2	2.2	3	8	105.271	0	0.3	1	50
75×100.B3.02	2	2.1	3	9	89.339	0	0.4	1	50
75×100.B3.03	2	2	2	10	106.922	0	0.1	1	50
75×100.B3.04	2	2.2	3	8	96.9	0	0.5	2	50
75×100.B3.05	2	2.1	3	9	98.896	0	0.2	1	50
75×100.B3.06	2	2.4	3	6	109.415	0	0.5	1	75
75×100.B3.07	2	2.3	3	7	83.265	0	0.2	1	75
75×100.B3.08	2	2.2	3	8	87.392	0	0.4	1	75
75×100.B3.09	2	2.7	5	7	122.905	0	1	2	75
75×100.B3.10	2	2.1	4	9	100.25	0	0.4	1	75

In Table 2 are shown the results for the first set of instances, that is, the set containing 50 potential hubs and 50 users. It can be noticed that only in 12 instances a perfect users balance is achieved. However, in the rest of the instances, a value of 1 is obtained, which indicates that the largest difference of users connected to hubs corresponds to one user. The worst balancing is 3, and it is obtained in instances 08, 09 and 10 of type B2, and 09 and 10 of type B3. Coincidentally, all these instances consider a  $p_{min}=0.75$ , which implies that at least 38 of the 50 users must be connected.

For the instances in which the perfect balancing is obtained, the number of runs that achieved that value is above 5. For example, in instances 07 of type B1 and instances 05 and 07 of type B3, the highest number of perfect balance is reported (with 9, 6 and 6, respectively).

Another convenient aspect of the numerical experimentation is that for instances 09 of type B1, and instances 01

and 02 of type B3, the best balancing is obtained in all the ten runs.

Regarding the required computational time for each instance, it is less than 1 minute for almost all of them. Only instances 02, 07 and 08 of type B2, and instance 10 of type B3 presented average execution times greater than one minute, that is, they reported times between 64.05 and 70.895 seconds.

It is important to emphasize that for all the 30 instances, the minimum number of hubs are enabled. However, in the worst case for 22 out of the 30 instances, an additional hub is enabled besides the one indicated in vector  $p_j$ . Similarly, the number of connected users is the minimum required, that is, 25 and 38 for  $p_{min}=0.5$  and  $p_{min}=0.75$ , respectively. Therefore, it would be interesting to perform an analysis to evaluate the convenience of enabling more hubs or connecting more users.

The obtained results for large-size instances are displayed in Table 3. In general, the perfect balancing is achieved in



more than half of the 30 instances. However, since the size of the instance increased, the *unbalance* also increased up to two. As expected, the required computational time increased. Now, it take between 1 and 2 minutes, in average, for solving each instance. In this case, instances 75×100.B2.07 and 75×100.B1.10 are the ones with minimum and maximum time, that is, 71.5 and 127.0 seconds, respectively. Similarly than for the medium-size instances, in most of the runs the minimum number of hubs are enabled. In average, only in 20 out of the 30 instances an extra hub is needed. Also, the minimum number of users is always connected, that is, no benefit for the upper-level decision maker is obtained for connecting an additional user. This is an interesting finding that may be analyzed in the upper-level problem.

### 4.3 Analyzing the performance of the NEA

To show the performance of the proposed algorithm, three random instances are selected and the best upper-level objective function is recorded at each iteration. Therefore, they are plotted in Fig. 5. The aim is to show the convergence of the NEA.

An important issue that arises from that figure is the small range of values for the upper-level objective function. It seems that from the beginning, the balance is relatively small. Recall that 0 represents the perfect balance among users and hubs of different types. Also, in two out of the three selected instances, the perfect balance is not achieved. However, the graph shows that the algorithm tends to perform well, even at large-size instances. In other words, it converges to a good solution.

### 4.4 Managerial insights

During the computational experimentation different values of  $p_j$  are considered. In other words, the impact of having more small-type hubs than large-type hubs is explored. This can be observed from Table 1, which three different configurations for each size of the instances are displayed.

For instance, enabling more small-type hubs does not guarantee that the perfect balance is obtained. It can be observed from Tables 2 and 3 that enabling medium and large-type hubs yield to offer more connection options to the users; resulting in a more balanced network. The latter prevails despite the fact that medium and large-type hubs are more expensive than small-type hubs. Particularly, see instances 50×50.B1.07 and 50×50.B3.07, both of them achieved perfect users balance, even though different sizes of hubs are enabled. Another remarkable case corresponds to instances that achieved an almost perfect balance. For example, instances 75×100.B1.02 and 75×100.B2.02. That is the reason to state that the use of different types of hubs leads not only to perfect or almost perfect balance, but also it takes consideration in the final user connections.

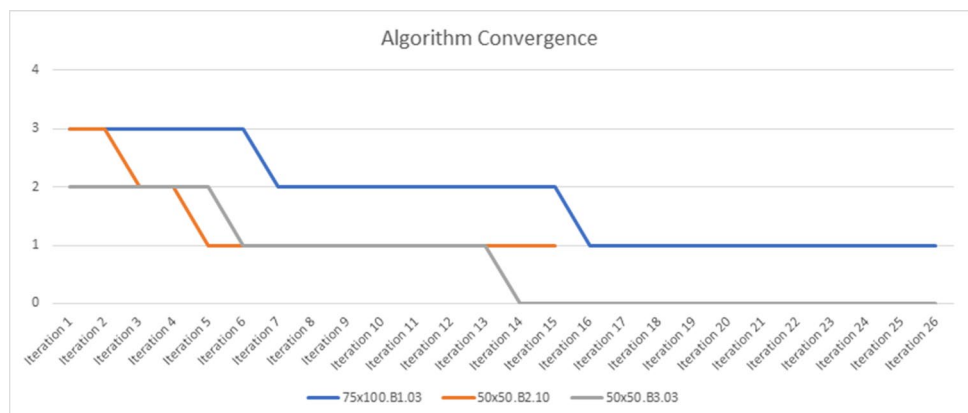
Moreover, in many instances, the perfect balance is achieved when additional hubs are enabled, that is, more than the minimum required hubs are used in the network. This may be caused by the number of alternatives offered to the users, who choose the best option for them.

In other words, if additionally to a perfect balance, the minimization of the network cost, or the maximization of the profit obtained from the connected users is pursued; then, a different structure of the solutions is obtained. The latter objective seeks an increase in the number of connected users.

## 5 Conclusions

In this paper, a bi-level problem of hub location in the context of a 5G network is presented. Nowadays, the use of 5G technology in the *new normality* that we are experiencing has become a pillar of the manner that the world needs to be connected. It is hard to conceive the idea of having electronic devices that are not connected to the Internet or work performance without the necessity of being connected to a telecommunication network. To accomplish the latter, companies that provide telephone and internet services must

Fig. 5 Convergence of the NEA



satisfy the needs of the users. Also, users select the service provider according to their needs. Given this situation in which there is an evident hierarchy among decision makers, a bi-level programming approach is applied. As a result of this, a novel bi-level model is proposed to study this interesting situation.

Based on the characteristics of the problem and due to the lack of specialized bi-level programming software, a nested evolutionary algorithm is proposed to solve the problem. This algorithm maintains a balance between diversification and intensification, since it involves elitist and stochastic components during the evolution. To evaluate the efficiency of the proposed algorithm, a large set of different instances is created.

Obtained results show the convenience for using the proposed algorithm to approximate good quality solutions. The algorithm solves the problem in a relatively short time, even for large instances. The deviations obtained are small, which implies that the proposed evolutionary algorithm is steady and robust. Interesting managerial insights are given to emphasize the impact of having different types of hubs in the instance. There is no evidence to indicate that enabling smaller-type of hubs is better than having large-type of hubs. Additionally, other objectives must be considered simultaneously in the problem for promoting the increase in the number of connected users in the network. Currently, the structure of the problem promotes that only the minimum number of required users are connected.

Nowadays, hub location problems in a telecommunications context are in vogue due to the social distancing forced by the pandemic of COVID-19. The problem herein proposed is the first effort to model 5G networks. Consequently, classic variants or the consideration of additional aspects can be studied to handle situations under this context.

**Data Availability** All data generated or analysed during this study are included in this published article or are available for interest readers upon request.

**Declarations** The authors did not receive support from any organization for the submitted work.

**Conflicts of Interest** The authors have no competing interests to declare that are relevant to the content of this article.

## References

- Al-Falahy N, Alani OY (2017) Technologies for 5g networks: Challenges and opportunities. *IT Prof* 19(1):12–20
- Akpakwu GA, Silva BJ, Hancke GP, Abu-Mahfouz AM (2017) A survey on 5g networks for the internet of things: Communication technologies and challenges. *IEEE Access* 6:3619–3647
- Hakimi SL (1964) Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper Res* 12(3):450–459
- O’Kelly M (1986) The location of interacting hub facilities. *Transp Sci* 20:92–106
- O’Kelly M (1987) A quadratic integer program for the location of interacting hub facilities. *Eur J Oper Res* 32:393–404
- Klincewicz JG (1996) A dual algorithm for the uncapacitated hub location problem. *Locat Sci* 4(3):173–184
- Klincewicz JG (1991) Heuristics for the p-hub location problem. *Eur J Oper Res* 53(1):25–37
- Campbell J (1994) Integer programming formulations of discrete hub location problem. *Eur J Oper Res* 72:387–405
- Martin JC, Román C (2004) Analyzing competition for hub location in intercontinental aviation markets. *Transport Res E Log* 40(2):135–150
- Carlos MJ, Román C (2003) Hub location in the south-atlantic airline market: A spatial competition game. *Transp Res A Policy Pract* 37(10):865–888
- Gelareh S, Nickel S (2011) Hub location problems in transportation networks. *Transport Res E Log* 47(6):1092–1111
- Campbell JF (2009) Hub location for time definite transportation. *Comput Oper Res* 36(12):3107–3116
- Contreras I, O’Kelly M (2019) Hub location problems. In: *Location science*, pp 327–363. Springer
- Carello G, DellaCroce F, Ghirardi M, Tadei R (2004) Solving the hub location problem in telecommunication network design: A local search approach. *Netw Int J* 44(2):94–105
- Rabbani M, Ravanbakhsh M, Farrokhi-Asl H, Taheri M (2017) Using metaheuristic algorithms for solving a hub location problem: application in passive optical network planning. *Int J Supply Oper Manag* 4(1):15–32
- Ebrahimi SB, Sharifi A, Nezhad-Afrasiabi M, Mirkhani SM (2019) A hybrid solution method for hub-and-spoke network design under uncertainty a case study to design optical fiber network in iran. *AUT J Model Simul* 51(2):179–190
- Osorio-Mora A, Núñez-Cerda F, Gatica G, Linfati R (2020) Multi-modal capacitated hub location problems with multi-commodities: An application in freight transport. *J Adv Transp* 2020:Article ID 2431763
- Mokhtarzadeh M, Tavakkoli-Moghaddam R, Triki C, Rahimi Y (2021) A hybrid of clustering and meta-heuristic algorithms to solve a p-mobile hub location-allocation problem with the depreciation cost of hub facilities. *Eng Appl Artif Intell* 98:104121
- Mahmoodjanloo M, Tavakkoli-Moghaddam R, Baboli A, Jamiri A (2020) A multi-modal competitive hub location pricing problem with customer loyalty and elastic demand. *Comput Oper Res* 123:105048
- Shang X, Yang K, Jia B, Gao Z (2021) Distributionally robust cluster-based hierarchical hub location problem for the integration of urban and rural public transport system. *Comput Ind Eng* 155:107181
- Monemi RN, Gelareh S, Nagih A, Maculan N, Danach K (2021) Multi-period hub location problem with serial demands: A case study of humanitarian aids distribution in lebanon. *Transport Res E Log* 149:102201
- Dempe S (2003) Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints. *A Journal of Mathematical Programming and Operations Research* 52:333–359
- Yu X, Gen M (2010) *Introduction to evolutionary algorithms*. Springer Science & Business Media
- Simon D (2013) *Evolutionary optimization algorithms*. John Wiley & Sons
- Ahn CW (2006) *Advances in evolutionary algorithms*. Springer

26. Ghanbari R, Mahdavi-Amiri N (2011) Solving bus terminal location problems using evolutionary algorithms. *Appl Soft Comput* 11(1):991–999
27. Villegas JG, Palacios F, Medaglia AL (2006) Solution methods for the bi-objective (cost-coverage) unconstrained facility location problem with an illustrative example. *Ann Oper Res* 147(1):109–141
28. Medaglia AL, Villegas JG, Rodríguez-Coca DM (2009) Hybrid biobjective evolutionary algorithms for the design of a hospital waste management network. *J Heuristics* 15(2):153–176
29. Sinha A, Malo P, Deb K (2013) Efficient evolutionary algorithm for single-objective bilevel optimization. [arXiv:1303.3901](https://arxiv.org/abs/1303.3901)
30. Wang Y, Jiao Y-C, Li H (2005) An evolutionary algorithm for solving nonlinear bilevel programming based on a new constraint-handling scheme. *IEEE Trans Syst Man Cybern Part C Appl Rev* 35(2):221–232
31. Sinha A, Malo P, Deb K (2014) An improved bilevel evolutionary algorithm based on quadratic approximations. In: 2014 IEEE congress on evolutionary computation (CEC), pp 1870–1877. IEEE
32. Sipper M, Fu W, Ahuja K, Moore JH (2018) Investigating the parameter space of evolutionary algorithms. *BioData Mining* 11(1):1–14

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.