An Iterated Greedy Algorithm with Variable Neighborhood Descent for the Planning of Specialized Diagnostic Services in a Segmented Healthcare System[a]

Rodolfo Mendoza-Gómez
Tecnológico de Monterrey, Campus Toluca
Department of Industrial Engineering
E-mail: *rodolfomendoza@itesm.mx*

Roger Z. Ríos-Mercado[b]
Universidad Autónoma de Nuevo León (UANL)
Graduate Program in Systems Engineering
E-mail: *roger.rios@uanl.edu.mx*

Karla B. Valenzuela-Ocaña
Tecnológico de Monterrey, Campus Toluca
Department of Industrial Engineering
E-mail: *kvalenzuela@itesm.mx*

February 2018
Revised July 2018, August 2018
Accepted September 2018

[a] *Journal of Industrial and Management Optimization* (forthcoming)
[b] Corresponding author

Abstract

In this paper, a problem arising in the planning of specialized diagnostic services in a segmented public healthcare system is addressed. The problem consists of deciding which hospitals will provide the service and their capacity levels, the allocation of demand in each institution, and the reallocation of uncovered demand to other institutions or private providers, while minimizing the total equivalent annual cost of investment and operating cost required to satisfy all the demand. An associated mixed-integer linear programming model can be solved by conventional branch and bound for relatively small instances; however, for larger instances the problem becomes intractable. To effectively address larger instances, a hybrid metaheuristic framework combining iterated greedy and variable neighborhood descent components for this problem is proposed. Two greedy construction heuristics are developed, one starting with an infeasible solution and iteratively adding capacity and the other starting with a feasible, but expensive, solution and iteratively decrease capacity. The iterated greedy algorithm includes destruction and reconstruction procedures. Four different neighborhood structures are designed and tested within a VND procedure. In addition, the computation of local search components benefit from an intelligent exploitation of problem structure since, when the first-level location variables (hospital location and capacity) are fixed, the remaining subproblem can be solved efficiently as it is very close to a transshipment problem. All components and different strategies were empirically assessed both individually and within the IGA-VND framework. The resulting metaheuristic is able to obtain near optimal solutions, within 3% of optimality, when tested over a data base of 60- to 300-hospital instances.

Keywords: Healthcare planning; integer programming; hybrid metaheuristics; iterated greedy algorithm; variable neighborhood descent.

1 Introduction

One of the important problems arising in the planning of healthcare service across a network of providers is how to efficiently locate and assign to patients specialized equipment from healthcare units. The problem arises in some developing countries where the healthcare system is composed by several public institutions, each in turn composed of a network of healthcare units with its own infrastructure. The objective is to minimize the planning and operative cost for providing a specialized diagnostic service covering all system demand in a one-year planning horizon. The capacity of the service is estimated according to the amount of medical equipment and their different levels of capacity. The service is segmented into levels of patient acuity in order to identify the degree of illness severity. Demand is allocated on a monthly basis. Some examples of specialized equipment required in these type of services are: magnetic resonance imaging, computed tomography, positron emission tomography, and digital mammography.

The problem was introduced by Mendoza-Gómez et al. (2016). In that work, the authors introduced some mixed-integer programming formulations, and studied the behavior of the system under different settings. Although some previous works successfully applied optimization techniques for related healthcare-planning problems (Ayvaz and Huh, 2010; Coté et al., 2007; Mahar et al., 2011; McLafferty and Broe, 1990; Mestre et al., 2015; Ruth, 1981; Stummer et al., 2004; Syam and Coté, 2010, 2012), this proposed model is unique because it considers incorporating in one model a segmented network of healthcare units, levels of patient acuity, different levels of capacity, and multi-period demand evaluation, something not previously done before to the best of our knowledge. For a detailed discussion of those works, the reader is referred to the work of Mendoza-Gómez et al. (2016).

One of the main conclusions observed by Mendoza-Gómez et al. (2016) was the difficulty of handling medium- to large-scale instances of the problem. The results indicated that conventional branch-and-bound methods (B&B) were able to successfully handle instances of up to 60 healthcare units. However, the results with larger instances (120 to 300 healthcare units) were poor. The need of a heuristic approach for handling large-scale instances was more than evident. For a specific diagnostic service, a realistic estimate of the number of hospitals that might be involved in the decision process, ranges from 50 to 300, depending on the type of service. Thus, in this work we focus on instances with sizes of 60 to 300 healthcare units. From now on, we will refer to healthcare units just as "hospitals" for the sake of convenience because most of these types of services are provided in hospitals, although other types of healthcare units could also fit in.

This paper proposes a hybrid metaheuristic composed of iterative greedy and variable neighborhood descent mechanisms for obtaining good-quality solutions to the planning-related problem of specialized diagnostic services in a segmented healthcare system. Both methods are well-known heuristics that have been successfully applied to many combinatorial optimization problems. Our proposed method also benefits from an intelligent exploitation of the mathematical structure of the problem. Specifically, the model has a location-allocation structure that can be seen as a two-level decomposition. In the first level, we have binary decision variables deciding where to locate equipment and its corresponding capacity. Once these variables are fixed, the second-level subproblem can be seen as a transshipment problem and thus can be solved very efficiently. As a consequence of this, we develop search strategies that focus on the first-level location variables. The proposed hybrid metaheuristic integrates all these components in a clever way.

The remainder of the paper is organized as follows. A discussion of related literature is given in Section 2. Section 3 describes the problem and presents the discrete optimization model. Particular attention is paid to the definition of the first- and second-level variables that are used in the problem decomposition. This section ends by presenting and describing the decomposition subproblem that is used within the metaheuristic. The proposed hybrid metaheuristic is presented in Section 4. This includes the details of the constructive and local search components and its corresponding empirical assessment. Finally, concluding remarks and future research directions are given in Section 5.

2 Literature Review

In this section, we discuss the papers that are more relevant to our research. One of the first models applied to location of hospital services was due to Ruth (1981), who proposed a quantitative model to aid the planning of hospital inpatient services in a regional hospital network, particularly the allocation of beds to the population at risk. An evaluation of critical-care services based on two attributes, namely the geographical accessibility of services and the number of patients served by each facility was presented by McLafferty and Broe (1990). Stummer et al. (2004) propose a multi-objective combinatorial problem to determine the location of medical departments within a hospital network. Ayvaz and Huh (2010) introduce a model for the planning of hospital capacity, taking different types of patients into account.

Tlahig et al. (2013) present a mixed-integer linear program to find the optimal location and capacity in a multi-period problem. This model is aimed at assessing a centralized versus distributed sterilization service within a hospital network. The authors were able to optimally solve moderate-size instances. Mahar et al. (2011) present a non-linear programming model to locate specialized healthcare services in a hospital network. The aim of that work was to prove how hospital networks with multiple locations can leverage pooling benefits while deciding where to locate specialized services. This model takes into account not only financial considerations but also patient service levels. Based on their model, the authors can determine the number

and identification of hospitals in a network that should have the specialized capacity, the levels of capacity required, and the guidelines for locations that should serve the demand in the network.

Coté et al. (2007) introduce a location-allocation model for specialized treatment of traumatic brain injury. The model is applied in case study in the US Department of Veterans Affairs (VA). The best location for treatment units between the VA medical centers and to allocate admissions to these units while minimizing admission treatment cost, admission travel cost, and the penalty cost associated with foregone treatment revenue and excess capacity utilization is determined by this model. In a follow-up work, Syam and Coté (2010) model the same problem from the point of view of a non-profit service organization. In a follow-up paper, Syam and Coté (2012) propose an extension of the model that minimizes the total cost borne by the health system as well as its patients and incorporates admission acuity levels, service proportion requirements, and admission retention rates. Their empirical results indicate that a decentralized system is costlier than a centralized one but also serves a higher proportion of admissions. In recent years, we have seen also some stochastic models such as the work by Mestre et al. (2015) and Zarrinpoor et al. (2017) who studied stochastic location-allocation models in the strategic planning of hospital networks.

There are also a number of survey papers reviewing location, location-allocation, and capacity-planning problems in the healthcare sector. See, for instance, the recent surveys by Daskin and Dean (2004), Rais and Viana (2011), Chauhan and Singh (2016), and Ahmadi-Javid et al. (2017).

Just to lay down our contribution into perspective. In our model, we integrated some features from previous works, such as the levels of demand, the assurance of coverage, the service capacity, investment and operative-cost considerations, and the evaluation of various periods. In addition, we considered some new characteristics such as the evaluation of capacity with different types of equipment units, transfer of patients between hospitals, and incorporation of outsourcing providers. In particular, the most important incorporation is the segmentation of the system by institutions, each with its own demand and infrastructure, but with the ability to request services from other institutions or private providers if necessary. This new problem can be used, in particular, to solve related service planning problems for healthcare systems that present multiple public institutions and in general to solve location-allocation problems with regard to independent systems that share capacity. Therefore, we can see that the proposed model has unique features that have not been previously studied.

3 Problem Description

This section presents the problem for a specialized medical technology service across a segmented hospital network with a one-year planning horizon. The aim is to minimize the total annual cost required to guarantee the service coverage for a given demand. The hospital network is integrated by public institutions that can share capacity among themselves or request the service of a specific network of providers each month when they reach full capacity. The decisions are to determine the location, the capacity of the service in each institution, and the allocation of demand according to the previous decisions.

Allocated demand is associated with the number of services to be provided each period internally by an institution, or externally by other institutions or private providers. The demand must be allocated within the hospitals of each institution, or, if there is not enough capacity, reallocated to other institutions' hospitals with idle capacity or outsourced to private providers. An example of this allocation scheme is shown in Figure 1. Different types of allocation are shown with different connector arrow types. Hospitals with setup service are highlighted. For example, Hospital B1 covers demand from Hospital B2, B3, and its own demand. However, part of this demand is reallocated to Hospital A3 and to Private Services 1 and 2.

The demand is classified into levels of patient acuity, which is a measure of nursing intensity required by a patient and the degree of illness severity. The monthly evaluation of demand across the one-year planning horizon is considered, different levels of patient acuity are served by the same equipment, and different levels of equipment capacity are evaluated.

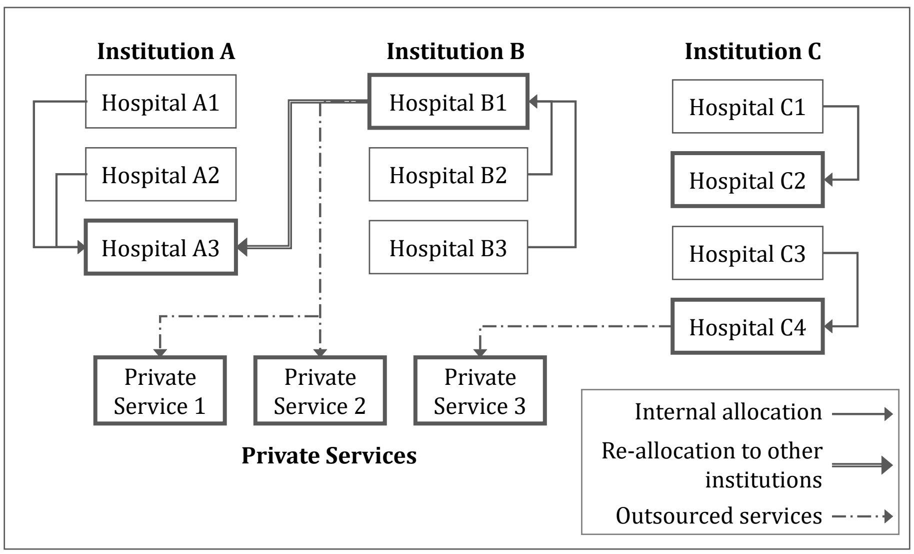

Figure 1: Example of the allocation problem.

3.1 The Integer Programming Formulation

The model studied in this paper corresponds to Model A from Mendoza-Gómez et al. (2016). For the sake of completeness and for a better understanding of the ensuing subproblem, we reproduce the formulation here.

The notation, parameters, and variables used in the problem formulation are as follows:

Indices and sets:

$k \in K$	Set of institutions.
$u \in U$	Set of levels of patient acuity.
$l \in L$	Set of types of equipment.

$n \in N$	Set of time periods (months).
$i, j \in I$	Set of hospitals in the network.
$G \subset I$	Set of public hospitals.
$G^k \subset G$	Set of public hospitals of institution k.
$G_i \subset G$	Set of public hospitals of the institution to which hospital i belongs.
$P \subset I$	Set of private service providers.
$k_j \in K$	Institution to which hospital j belongs.

Parameters:

FC_j	Fixed annual service setup cost in hospital j; $j \in G$.
VC_l	Variable annual setup cost for equipment of type l; $l \in L$.
OC^u	Operational cost for providing a service with patient acuity level u; $u \in U$.
TC_{ij}^u	Transfer cost for sending a patient with acuity level u from hospital i to hospital j; $i \in G$, $j \in I$, $u \in U$.
AC_k^u	Additional charge that institution k requests of other institutions to provide a service for a patient acuity level u ; $u \in U$, $k \in K$.
PC_j^u	Cost of provider j for a service for a patient acuity level u; $j \in P$, $u \in U$.
D_{in}^u	Demand (number of patients) with acuity level u in hospital i in period n; $i \in G$, $n \in N$, $u \in U$.
EC_l	Maximum capacity (number of services) of equipment type l in each period; $l \in L$.
CP_{jn}	Maximum capacity (number of services) of provider j in period n; $j \in P$, $n \in N$.
H_{jl}	Minimum number of required equipment type l in hospital j; $j \in G$, $l \in L$.
δ_k	Minimum percentage of annual demand to be internally covered by institution k; $k \in K$.
σ_k	Maximum demand in proportion to the capacity that each hospital of institution k can allocate; $k \in K$.
ω_k	Maximum percentage of annual demand that institution k is allowed to allocate to outsourcing; $k \in K$.
M	A very large positive value.

Decision variables:

x_{ijn}^u	Number of patients with acuity level u from hospital i allocated to hospital j in period n; $i \in G,\ j \in I,\ u \in U,\ n \in N$.
α_{jn}^u	Number of patients with acuity level u allocated to hospital j in period n unserved by any hospital of its institution; $j \in G,\ u \in U,\ n \in N$.
β_{jn}	Capacity available in hospital j in period n unused by any hospital of its institution; $j \in G,\ n \in N$.

s^u_{jn}	Service level for patient acuity level u in hospital j in period n; $j \in G,\ u \in U,\ n \in N$.
t_{jl}	Number of equipment units of type l that are allocated to hospital j; $j \in G,\ l \in L$.
y_j	Binary variable equal to 1 if any service is set up in hospital j, and 0 otherwise; $j \in G$.

$$\text{(Model A)} \quad \text{Minimize} \quad \sum_{j\in G} FC_j \cdot y_j + \sum_{j\in G}\sum_{l\in L} VC_l \cdot t_{jl} + \sum_{j\in G}\sum_{u\in U}\sum_{n\in N} OC^u \cdot s^u_{jn} + \sum_{k\in K}\sum_{u\in U} AC^u_k \cdot \sum_{n\in N}\sum_{i\in G\setminus G^k}\sum_{j\in G^k} x^u_{ijn} + \sum_{i\in G}\sum_{j\in P}\sum_{u\in U}\sum_{n\in N} PC^u_j \cdot x^u_{ijn} + \sum_{i\in G}\sum_{j\in I}\sum_{u\in U}\sum_{n\in N} TC^u_{ij} \cdot x^u_{ijn} \tag{1}$$

subject to:

$$\sum_{j\in G_i} x^u_{ijn} = D^u_{in} \qquad i \in G,\ u \in U,\ n \in N \tag{2}$$

$$\sum_{i\in G_j}\sum_{u\in U} x^u_{ijn} - \sum_{u\in U} \alpha^u_{jn} + \beta_{jn} = \sum_{l\in L} EC_l \cdot t_{jl} \qquad j \in G,\ n \in N \tag{3}$$

$$\sum_{i\in G}\sum_{u\in U} x^u_{ijn} \le M \cdot y_j \qquad j \in G,\ n \in N \tag{4}$$

$$\alpha^u_{jn} \le \sum_{i\in G_j} x^u_{ijn} \qquad j \in G,\ u \in U,\ n \in N \tag{5}$$

$$\beta_{jn} \le \sum_{l\in L} EC_l \cdot t_{jl} \qquad j \in G,\ n \in N \tag{6}$$

$$\sum_{j\in G^k}\sum_{l\in L} |N| \cdot EC_l \cdot t_{jl} \ge \delta_k \cdot \sum_{i\in G^k}\sum_{u\in U}\sum_{n\in N} D^u_{in} \qquad k \in K \tag{7}$$

$$\sum_{i\in G}\sum_{u\in U} x^u_{ijn} \le \sigma_{k_j} \cdot \sum_{l\in L} EC_l \cdot t_{jl} \qquad j \in G,\ n \in N \tag{8}$$

$$\sum_{i\in G^k}\sum_{j\in P}\sum_{u\in U}\sum_{n\in N} x^u_{ijn} \le \omega_k \cdot \sum_{i\in G^k}\sum_{u\in U}\sum_{n\in N} D^u_{in} \qquad k \in K \tag{9}$$

$$\sum_{j\in I\setminus G_i} x^u_{ijn} = \alpha^u_{in} \qquad i \in G,\ u \in U,\ n \in N \tag{10}$$

$$\sum_{i\in G\setminus G_j}\sum_{u\in U} x^u_{ijn} \le \beta_{jn} \qquad j \in G,\ n \in N \tag{11}$$

$$\sum_{i\in G} x^u_{ijn} - \alpha^u_{jn} = s^u_{jn} \qquad j \in G,\ u \in U,\ n \in N \tag{12}$$

$$\sum_{i\in G}\sum_{u\in U} x^u_{ijn} \le CP_{jn} \qquad j \in P,\ n \in N \tag{13}$$

$$t_{jl} \le M \cdot y_j \qquad j \in G,\ l \in L \tag{14}$$

$$y_j \leq \sum_{l \in L} t_{jl} \qquad j \in G \qquad (15)$$

$$t_{jl} \geq H_{jl} \qquad j \in G,\ l \in L \qquad (16)$$

$$x^u_{ijn} \in \mathbb{N} \cup \{0\} \qquad i \in G,\ j \in I,\ u \in U,\ n \in N \qquad (17)$$

$$\alpha^u_{jn},\ s^u_{jn} \in \mathbb{N} \cup \{0\} \qquad j \in G,\ u \in U,\ n \in N \qquad (18)$$

$$\beta_{jn} \in \mathbb{N} \cup \{0\} \qquad j \in G,\ n \in N \qquad (19)$$

$$t_{jl} \in \mathbb{N} \cup \{0\} \qquad j \in G,\ l \in L \qquad (20)$$

$$y_j \in \{0, 1\} \qquad j \in G \qquad (21)$$

The objective function (1) minimizes the total equivalent annual cost to provide the service for all the demand of the public hospitals in the network. The first and second terms represent fixed and variable annual investment costs, respectively. The third term represents the total operational costs of all services provided, and the fourth term represents the inter-institutional fee for all services received from other institutions. The fifth term corresponds to the total outsourcing costs, and the last term represents the total transportation cost for all types of patient acuity levels.

Constraints (2) ensure that all demand for each hospital is allocated within its own institution in each period. Constraints (3) determine the demand from the same institution allocated to a hospital according to its capacity in each period, and the idle capacity or the unallocated demand in each hospital with set-up capacity according to the case. Constraints (4) prevent allocating demand to a hospital if the service is not set up. Constraints (5) ensure that the variables for a hospital's uncovered demand only take values equal to or lower than total demand allocated to that hospital in the same period. Constraints (6) ensure that for each period, each idle capacity variable is lower than or equal to the capacity of the hospital which it belongs to. Constraints (7) ensure a minimum percentage of annual capacity according to the total annual demand of each institution, defined by $0 \leq \delta_k \leq 1$. Constraints (8) establish an upper bound for demand that can be allocated to a hospital in each period. This limit must not exceed the percentage of its capacity defined by each institution ($\sigma_k \geq 1$).

Constraints (9) set the maximum percentage of an institution's total annual demand to be reallocated to the private providers by each institution ($0 \leq \omega_k \leq 1$). Constraints (10) ensure that a hospital's uncovered demand inside its institution will be reallocated to another hospital of a different institution with idle capacity or to private providers in each period. Constraints (11) allow allocating uncovered demand of other institutions to a hospital without exceeding its idle capacity in each period. Constraints (12) are used to determine the service level of each hospital for each patient acuity level in each period. Constraints (13) limit the demand reallocated to each private provider according to its capacity in each period. Constraints (14) -(15) relate the integer variables y_j and t_{jl}. Constraints (16) enforce setting up the service with a predetermined number of equipment units of each type in a hospital that already has the service or it is mandatory to set up. Finally, the nature of the decision variables is given by (17)-(21).

3.2 Problem Decomposition

As stated before, we attempt to take advantage of the location-allocation structure of the model. For this purpose, we consider the subproblem obtained when the location decision variables y_j and t_{jl} are fixed. In fact, even for the first-level variables, we only need to know the t_{jl} variables since the y_j variables depend on the value of the t_{jl} variables.

Let $\bar{t}_{jl}$ and $\bar{y}_j$ represent the fixed and known values of the location first-level variables t_{jl} and y_j, respectively. First, as previously mentioned, knowing $\bar{t}_{jl}$ allows us to compute $\bar{y}_j$ as follows:

$$\bar{y}_j = \begin{cases} 1 & \text{if} \quad \sum_{l \in L} \bar{t}_{jl} \geqslant 1 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

In addition, let $W_j(\bar{t}_{jl})$, or simply W_j when understood from the context, represent the remaining hospital capacity for the lower-level subproblem as a function of $\bar{t}_{jl}$, which is given by:

$$W_j(\bar{t}_{jl}) = \sum_{l \in L} EC_l \cdot \bar{t}_{jl} \tag{23}$$

The subproblem is referred to Model S($\bar{y}_j$, $\bar{t}_{jl}$) to emphasize the dependency on the first-level variables, or simply as Model S($\bar{t}_{jl}$) (due to the dependency of the $\bar{y}_j$ on the $\bar{t}_{jl}$ given above). The additional notation, parameters, and variables used in the subproblem formulation are as follows:

Sets

$i, j \in O$ Set of hospitals with set-up service in the current solution ($O \subseteq G$).

$O_j \subset O$ Set of hospitals with set-up service of institution to which hospital j also belongs.

$O^k \subset O$ Set of hospitals with set-up service of institution k.

Parameters

W_j Capacity of hospital j; $j \in O$.

DC Penalty cost for unmet demand.

Variables

v_{in}^u Unserved demand for patient acuity level u in hospital i in period n; $i \in O,\ u \in U,\ n \in N$.

$$\textbf{Model S}(\bar{t}_{jl}) \quad \text{Min} \quad \sum_{j\in O}\sum_{u\in U}\sum_{n\in N} OC^u \cdot s^u_{jn} + \sum_{i\in O}\sum_{j\in P}\sum_{u\in U}\sum_{n\in N} PC^u_j \cdot x^u_{ijn} + \sum_{i\in O}\sum_{u\in U}\sum_{n\in N} DC \cdot v^u_{in}$$

$$+ \sum_{i\in G}\sum_{j\in O\cup P}\sum_{u\in U}\sum_{n\in N} TC^u_{ij} \cdot x^u_{ijn} + \sum_{k\in K}\sum_{u\in U}\sum_{n\in N} AC^u_k \sum_{i\in O\setminus G^k}\sum_{j\in O^k} x^u_{ijn} \qquad (24)$$

subject to:

$$\sum_{j\in O_i} x^u_{ijn} = D^u_{in} \qquad i\in G,\ u\in U,\ n\in N \qquad (25)$$

$$\sum_{i\in G_j}\sum_{u\in U} x^u_{ijn} + \beta_{jn} - \sum_{u\in U}\alpha^u_{jn} = W_j(\bar{t}_{jl}) \qquad j\in O,\ n\in N \qquad (26)$$

$$\alpha^u_{jn} \le \sum_{i\in G_j} x^u_{ijn} \qquad j\in O,\ u\in U,\ n\in N \qquad (27)$$

$$\beta_{jn} \le W_j(\bar{t}_{jl}) \qquad j\in O,\ n\in N \qquad (28)$$

$$\sum_{i\in O^k}\sum_{j\in P}\sum_{u\in U}\sum_{n\in N} x^u_{ijn} \le \omega_k \sum_{i\in G^k}\sum_{u\in U}\sum_{n\in N} D^u_{in} \qquad k\in K \qquad (29)$$

$$\sum_{j\in O\setminus G_i} x^u_{ijn} + v^u_{in} = \alpha^u_{in} \qquad i\in O,\ u\in U,\ n\in N \qquad (30)$$

$$\sum_{i\in O\setminus G_j} x^u_{ijn} \le \beta_{jn} \qquad j\in O,\ n\in N \qquad (31)$$

$$\sum_{i\in G}\sum_{u\in U} x^u_{ijn} \le \sigma_{k_j}\cdot W_j(\bar{t}_{jl}) \qquad j\in O,\ n\in N \qquad (32)$$

$$\sum_{i\in G_j} x^u_{ijn} + \sum_{i\in O\setminus G_j} x^u_{ijn} - \alpha^u_{jn} = s^u_{jn} \qquad j\in O,\ u\in U,\ n\in N \qquad (33)$$

$$\sum_{i\in O}\sum_{u\in U} x^u_{ijn} \le CP_{jn} \qquad j\in P,\ n\in N \qquad (34)$$

$$x^u_{ijn} \in \mathbb{N}\cup\{0\} \qquad i\in G,\ j\in O\cup P, u\in U,\ n\in N \qquad (35)$$

$$\beta_{jn} \in \mathbb{N}\cup\{0\} \qquad j\in O,\ n\in N \qquad (36)$$

$$\alpha^u_{jn}, v^u_{jn},\ s^u_{jn} \in \mathbb{N}\cup\{0\} \qquad j\in O,\ u\in U,\ n\in N \qquad (37)$$

To ensure subproblem feasibility, the variables v^u_{in} are used to allocate unmet demand when there is not enough capacity in the system to allocate all demand across hospitals. A penalty cost (DC) is added in the objective function when there is unmet demand, and this cost must be large enough so that all capacity in the system will be used first. Note that constraints (7) from Model A were not included in Model S. The resulting relaxed Model S is easier to solve this way. The heuristics described in the following section ensure these constraints are eventually met.

Thus, in essence, for given fixed values $\bar{t}_{jl}$, a corresponding subproblem solution is obtained by first computing $\bar{y}$ and W given by Equations (22) and (23), and then solving subproblem $\mathrm{S}(\bar{t}_{jl})$ for these fixed values. The output of the subproblem is denoted by solution $(\bar{x}, \bar{s}, \bar{\alpha}, \bar{v}, \bar{\beta})$, and therefore, a complete solution is given by: $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{v}, \bar{\beta})$.

Model S is a MILP that can be solved relatively quickly by branch and bound given its similarity with a transshipment problem. In fact, in preliminary work (Mendoza-Gómez et al., 2016), it was observed that solving the LP relaxations by the dual simplex algorithm instead of the primal simplex method renders the solution even faster, and it takes advantage of the warm-start feature of branch and bound, since changing hospital capacity between subproblem solution calls does not affect dual feasibility.

3.3 Parameter Description

This subsection shows the computation and update operation of some parameters that are used in the proposed heuristics.

Let us assume that an incumbent solution to the problem is given by $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{v}, \bar{\beta})$. The cost-benefit ratio of each equipment of type l, $l \in L$, is given by:

$$CB_l = VC_l / EC_l \tag{38}$$

The total uncovered demand in a solution is given by:

$$UD = \sum_{i \in O} \sum_{u \in U} \sum_{n \in N} \bar{v}^u_{in} \tag{39}$$

The total cost of service in hospital j, $j \in G$, is given by:

$$\begin{aligned} TS_j = &\sum_{u \in U} \sum_{n \in N} \left(OC^u \cdot \bar{s}^u_{jn} + \sum_{k \in K \setminus k_j} AC^u_k \sum_{i \in O \setminus G^k} \bar{x}^u_{jin} + \sum_{i \in P} PC^u_i \cdot \bar{x}^u_{jin} + \sum_{i \in O \cup P} TC^u_{ji} \cdot \bar{x}^u_{jin} \right) \\ &+ FC_j \cdot \bar{y}_j + \sum_{l \in L} VC_l \cdot \bar{t}_{jl} \end{aligned} \tag{40}$$

The unit cost for service in hospital j, $j \in G$, is given by:

$$UC_j = \begin{cases} \dfrac{TS_j}{\sum_{u \in U} \sum_{i \in G} \bar{x}^u_{ijn}} & \text{if} \quad \bar{y}_j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

The transportation cost for hospital i with no set-up service, $i \in G$, is given by:

$$ITC_i = \begin{cases} \sum_{j \in G_i} \sum_{u \in U} \sum_{n \in N} TC_{ij}^u \cdot \bar{x}_{ijn}^u & \text{if} \quad \bar{y}_j \neq 1 \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

The overall cost of hospital i to provide the service internally or externally, $i \in G$, is given by:

$$OVC_i = \sum_{u \in U} \sum_{j \in G} \sum_{n \in N} \bar{x}_{ijn}^u \cdot UC_j + ITC_i \tag{43}$$

The following binary parameters are used within the heuristic to evaluate possible feasible movements or requirements.

Candidate institutions that require additional capacity: Let CI_k be equal to 1 if institution $k \in K$ necessarily requires additional capacity to satisfy constraints (7) and (8) of model A, and 0 otherwise.

$$CI_k = \begin{cases} 1 \quad \text{if} & \max_{n \in N} \sum_{u \in U} \sum_{i \in G^k} D_{in}^u > \sum_{i \in G^k} W_i \cdot \sigma_k \text{ and } \sum_{u \in U} \sum_{i \in G^k} \sum_{n \in N} D_{in}^u \cdot \delta_k > \sum_{i \in G^k} W_i \cdot |N| \\ 0 \quad \text{otherwise} \end{cases} \tag{44}$$

Candidate equipment to be added in a hospital: Let CE_{il} be equal to 1 if an equipment of type l is feasible to be set up in hospital i without violating constraints (8) of model A, and 0 otherwise; $i \in G$ and $l \in L$.

$$CE_{il} = \begin{cases} 1 \quad \text{if} \quad \max_{n \in N} \sum_{u \in U} D_{in}^u \leqslant (W_i + EC_l) \cdot \sigma_{ki} \\ 0 \quad \text{otherwise} \end{cases} \tag{45}$$

Candidate equipment to be removed from an institution: Let EQI_{kl} be equal to 1 if an equipment of type l in institution k is a candidate for elimination without violating any constraint, and 0 otherwise; $k \in K$ and $l \in L$.

$$EQI_{kl} = \begin{cases} 1 \quad \text{if} \quad \max_{n \in N} \sum_{u \in U} \sum_{i \in G^k} D_{in}^u \leqslant \sigma_k \left(\sum_{i \in G^k} W_i - EC_l \right) \\ \quad \text{and} \sum_{u \in U} \sum_{i \in G^k} \sum_{n \in N} D_{in}^u \cdot \delta_k \leqslant |N| \left(\sum_{i \in G^k} W_i - EC_l \right) \\ 0 \quad \text{otherwise} \end{cases} \tag{46}$$

Candidate equipment to be removed from a hospital: Let EQH_{jl} be equal to 1 if an equipment of type l in hospital j is a candidate for elimination without violating any constraint, and 0 otherwise; $j \in G$ and $l \in L$.

$$EQH_{jl} = \begin{cases} 1 & \text{if} \quad \bar{t}_{jl} - H_{jl} > 0 \text{ and } EQI_{k_j l} > 0 \text{ and } \max_{n \in N} \sum_{u \in U} D^u_{jn} \leqslant (W_j - EC_l) \cdot \sigma_{k_j} \\ 0 & \text{otherwise} \end{cases} \tag{47}$$

The idle capacity in hospital $j \in G$, is given by:

$$IC_j = \begin{cases} \sum_{n \in N} \left(W_j - \sum_{u \in U} s^u_{in} \right) & \text{if} \quad \bar{y}_j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{48}$$

4 Proposed Hybrid Metaheuristic

In this section, we describe a metaheuristic framework that integrates iterated greedy algorithms (IGA) and variable neighborhood descent (VND).

The IGA, also referred to as iterated local search (ILS) (Lourenço et al., 2003), is a stochastic local search method (Hoos and Stützle, 2004) that generates a sequence of solutions by iterating over a greedy construction heuristic using destruction and reconstruction phases. The first works were proposed by Ruiz and Stützle (2006) for the permutation flow-shop scheduling problem and by Ruiz and Stützle (2008) for the sequence-dependent set-up time flow-shop problem with makespan and weighted tardiness objectives. In Quevedo-Orozco and Ríos-Mercado (2015), the authors present an iterated greedy local search with VND to solve the capacitated vertex p-center problem. Some important applications of IGA have been also proposed by Ruiz and Stützle (2008), Yuan et al. (2008), Ribas et al. (2011), and more recently by Pan and Ruiz (2014).

Variable neighborhood search (VNS) is a metaheuristic proposed by Mladenović and Hansen (1997) based on the simple principle of systematically changing the structure of neighborhoods inside the search. Variable neighborhood descent is a special case of VNS when the choice of neighborhood is made in a deterministic way. VNS/VND have been successfully applied to many discrete location problems (Crainic et al., 2004; García-López et al., 2002; Hansen and Mladenović, 1997, 2001; Ljubić, 2007; Mladenović and Hansen, 2003; Quevedo-Orozco and Ríos-Mercado, 2015).

Some VNS approaches have been developed for location problems. For instance, Hansen and Mladenović (1997) proposed a VNS for the p-median problem. Later, Hansen and Mladenović (2001) applied the reduced VNS and variable neighborhood decomposition search (VNDS) to solve larger instances of the problem. Two parallel VNS versions were proposed to solve this problem by Crainic et al. (2004) and García-López et al. (2002). The p-center problem, which consists in locating p facilities and assigning clients to them in order to minimize the maximum distance between a client and the facility to which it is allocated, was addressed by Mladenović and Hansen (2003), who proposed a basic VNS and Tabu search. For simple plant location problems, a VNDS was developed by Hansen and Mladenović (2001). A double VNS heuristic was proposed by Diakova and Kochetov (2012) for the facility location problem and pricing problem in which the facilities can charge different prices; the objective is to maximize the overall revenue. A hybrid VNS for the connected facility location problem was presented by Ljubić (2007). In particular, VNS/VND methods have also been applied in healthcare problems. An application of VNS in healthcare facility location was presented by Marić et al. (2013), who developed a hybrid metaheuristic based on combining the evolutionary approach with modified VNS for determining the location for long-term healthcare facilities. Rego and de Sousa (2009) presented a hybrid Tabu search/VNS metaheuristic for the design of alternative configurations in a hospital supply chain.

In the remainder of this section we provide the building blocks for the proposed metaheuristic. We first describe two constructive heuristics. These are integrated later in an IGA, which is fully described. Then, the VND is described in detail. The last section presents the overall IGA-VND framework. Each component is empirically assessed in each subsection.

4.1 Constructive Heuristics

Two heuristic strategies are proposed in this section. The first one constructs a solution from scratch, adding capacity at each iteration, and the second one is a two-phase heuristic that first constructs a solution with excess of capacity and then refines and decreases this excess capacity in a second phase.

Both heuristics require an initial presolution, that is, finding initial values for parameter $\bar{t}$. The requirement is to fix at least one equipment unit in a hospital for each institution. To construct this presolution, parameters H_{jl} are first copied to $\bar{t}_{jl}$, since these are the mandatory equipment units required for each hospital. Then, if an institution does not have any equipment units set up at a hospital, it is required to set one up. In this case, it is suggested to select the hospital with the highest demand and the equipment type with the lowest cost-benefit ratio according to Equation (38).

Constructive Method 1

The core of this constructive method (CM1) is to add capacity to the system in each iteration until feasibility is achieved. An equipment unit is added to the current solution in each iteration to increase the capacity of the system. A greedy function is used to select the hospital that will have its capacity increased.

Then, the current solution is updated, the subproblem (Model S) is solved, and the procedure is repeated until all constraints of Model A are satisfied. The procedure is shown in Pseudocode 1.

Pseudocode 1 Constructive Method 1

1: **procedure** CONSTRUCTIVE_METHOD_1($\bar{t}$)
2:     Solve_Subproblem($\bar{t}$) (relaxing (32)) ;
3:     Compute CI_k and uncovered demand UD according to (44) and (39) ;
4:     **while** ($\sum_{k \in K} CI_k > 0$ OR $UD > 0$) **do**
5:         **if** ($\sum_{k \in K} CI_k = 0$) **then**
6:             $CL \leftarrow \{G\}$;
7:         **else**
8:             **for** (any $k \in K | CI_k > 0$) **do**
9:                 $CL \leftarrow CL \cup \{G_k\}$;
10:             **end for**
11:         **end if**
12:         Update CE_{il} and OVC_i in each hospital according to (45) and (43);
13:         $i^* \leftarrow \arg\max_{i \in CL}\{OVC_i \mid \sum_{l \in L} CE_{il} > 0\}$;
14:         $l^* \leftarrow \arg\min_{l \in L}\{CB_l \mid CE_{i^*l} > 0\}$;
15:         $\bar{t}_{i^*l^*} \leftarrow \bar{t}_{i^*l^*} + 1$;
16:         Solve_Subproblem($\bar{t}$) (relaxing (32));
17:         Update CI_k and uncovered demand UD according to (44) and (39) ;
18:     **end while**
19:     Solve_Subproblem($\bar{t}$) ;
20: **return** $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{\beta})$
21: **end procedure**

At first, in Step 2, we solve the subproblem (Model S) but relax the constraints (32), since they could generate an infeasible solution. At the end of the procedure, these constraints are no longer relaxed (Step 19).

The initial solution may be infeasible for Model A because the constraints (7) and (8) were relaxed. The feasibility is evaluated with Equations (44) and (39). If any of these values is strictly positive (i.e., it is infeasible), then it is required to add an additional equipment unit to the system to increase its capacity. A candidate list of hospitals (CL) is created to select the hospital to which an equipment unit will be added. If all CI_k are equal to zero, the candidate list is formed by all sets of public hospitals (G), but if some $k \in K$, $CI_k = 1$, only the hospitals that belong to that institution (G_k) will be added to CL. A greedy function (OVC_i) is used to select a hospital from CL, and this function evaluates the overall cost that each hospital requires to satisfy its demand. To determine this value, the total cost of each hospital with installed service is divided proportionally among all hospitals of its own institution that allocate demand to it. The equations used to calculate these values are (40), (41), (42), and (43); these equations use the values of the current solution of Model S. The hospital from CL with higher OVC_i and the type of equipment with lower cost-benefit (evaluated with Equation (38) in Step 14) is selected. The chosen hospital

i^*, to which equipment of type l^* will be added, must also guarantee feasibility of constraints (8) of Model A; this is evaluated with Equation (45). Once $\bar{t}$ is updated, it is required to resolve Model S to update the new solution. The feasibility of the new solution is reevaluated, and the previous steps are repeated until a complete feasible solution is found. Then, Model S is solved without relaxing constraints (32) to get the final feasible solution of Model A.

Constructive Method 2

This is a two-phase heuristic (depicted in Pseudocode 2) with the following idea. In phase one, an initial feasible solution that satisfies all constraints of Model A is obtained by a greedy constructive procedure (Step 2). Then, the second phase (Step 3) attempts to improve the objective function value without losing feasibility. Both procedures are described next.

Pseudocode 2 Constructive Method 2

1: **procedure** CONSTRUCTIVE_METHOD_2($\bar{t}$)
2: $\quad\bar{t} \leftarrow$ Greedy_Construction($\bar{t}$);
3: $\quad(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{\beta}) \leftarrow$ Improvement($\bar{t}$);
4: **return** $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{\beta})$
5: **end procedure**

The idea behind the Greedy_Construction($\bar{t}$) procedure, depicted in Pseudocode 3, is to solve the location problem for each institution without considering the interaction among institutions or the interaction with private providers. The rest of the process determines, for each institution, the hospitals and the number of equipment units of each type that will be set up.

Initially, in Step 2, we determine the initial capacity and the hospital set-up capacity with Equations (23) and (22), given the initial presolution. For each institution all demand must be allocated at the end of the procedure. The demand parameters (D) are copied to $\bar{D}$ to identify unallocated demand. To identify unallocated capacity, W is copied to w_n for each period. The demand will be allocated to hospitals with idle capacity ($w_n > 0$) in each period. If the capacity is full but there is still unallocated demand, a new equipment unit must be added to the system. The algorithm will stop when this requirement and constraints (7)-(8) of Model A, determined by Equation (44), are satisfied.

The demand that belongs to the acuity level with the highest cost and cheapest transport cost is allocated first. Then, the allocated demand is removed from $\bar{D}$ and the available capacity is updated from w_n; this procedure is repeated until there is no available capacity.

Pseudocode 3 Constructive Method 2 Phase 1

1: **procedure** GREEDY_CONSTRUCTION($\bar{t}$)
2: Set $W = (W_j(\bar{t}))$ and $\bar{y} = (\bar{y}_j(\bar{t}))$;
3: $\bar{D} \leftarrow D$;
4: $w_n \leftarrow W, n \in N$;
5: Determine candidate institutions CI according to (44);
6: **for** ($k \in K$) **do**
7: **while** ($\sum_{u \in U} \sum_{i \in G^k} \sum_{n \in N} \overline{D}^u_{in} > 0$ or $CI_k > 0$) **do**
8: **for** ($j \in G^k \mid \sum_{n \in N} w_{jn} > 0$) **do**
9: **for** ($n \in N$) **do**
10: **while** ($w_{jn} > 0$ and $\sum_{i \in G^k} \sum_{u \in U} \overline{D}^u_{in} > 0$) **do**
11: $u^* \leftarrow \arg\max_{u \in U}\{OC^u\}$;
12: $i^* \leftarrow \arg\min_{i \in G^k}\{TC^{u^*}_{ij}\}$;
13: Adjust $\overline{D}^{u^*}_{i^*n} \leftarrow \overline{D}^{u^*}_{i^*n} - \min\{\overline{D}^{u^*}_{i^*n}, w_{jn}\}$;
14: Adjust $w_{jn} \leftarrow w_{jn} - \min\{\overline{D}^{u^*}_{i^*n}, w_{jn}\}$;
15: **end while**
16: **end for**
17: **end for**
18: **if** ($\sum_{u \in u} \sum_{i \in G^k} \sum_{n \in N} \overline{D}^u_{in} > 0$ or $CI_k > 0$) **then**
19: Update candidate equipments CE according to (45) ;
20: $\lambda^k_{jl} \leftarrow FC_j/(W_j + C_l) + (VC_l + \sum_{i \in G_j} \sum_{u \in U} \sum_{n \in N} TC^u_{ij} \cdot \overline{D}^u_{in})/C_l \quad \forall j \in G^k, l \in L$;
21: $(j^*, l^*) \leftarrow \arg\min_{j \in G^k, l \in L}\{\lambda^k_{jl} \mid CE_{jl} > 0\}$;
22: $\bar{t}_{j^*l^*} \leftarrow \bar{t}_{j^*l^*} + 1$;
23: Update W and $\overline{y}$;
24: $w_{j^*n} \leftarrow w_{j^*n} + C_{l^*} \quad \forall n \in N$;
25: **end if**
26: **end while**
27: **end for**
28: **return** $\bar{t}$
29: **end procedure**

When capacity is not available, capacity must be increased to cover the remaining unallocated demand. To select the hospital that will increase its capacity and the type of equipment to set up, a greedy function is used (Step 20). This function evaluates for each hospital the cost-benefit of adding capacity. The function considers the initial fixed cost divided by the total potential capacity, plus the variable initial cost of the potential new equipment units and the total transportation cost, supposing that all unallocated demand is allocated to this hospital. This sum is divided by the capacity of the new potential equipment.

We must ensure that the selection of this hospital and the type of equipment selected do not violate constraints (8). This is evaluated with Equation (45). Then, the new available capacity is updated and unallocated demand is reallocated. This procedure is repeated until no unallocated demand exists and constraints (7) and (8) are satisfied.

Pseudocode 4 Constructive Method 2 Phase 2

procedure IMPROVEMENT($\bar{t}$)
  Solve_Suproblem($\bar{t}$);
  $t^{\text{best}} \leftarrow \bar{t}$;
  Compute EQH and UD according to (46) and (39);
  Improve $\leftarrow$ 1;
  while ($\sum_{j \in G} \sum_{l \in L} EQH_{jl} > 0$, $UD = 0$ and Improve $= 1$) **do**
    Update IC according to (48);
    $j^* \leftarrow \arg\max_{j \in G}\{IC_j | \sum_{l \in L} EQH_{jl} > 0\}$;
    $l^* \leftarrow \arg\max_{l \in L}\{CB_l | EQH_{j^*l} > 0\}$;
    $\bar{t}_{j^*l^*} \leftarrow \bar{t}_{j^*l^*} - 1$;
    Solve_Subproblem($\bar{t}$);
    if ($Z(\bar{t}) < Z(t^{\text{best}})$) **then**
      $t^{\text{best}} \leftarrow \bar{t}$;
      Update UD and EQH according to (46) and (39);
    else
      Improve $\leftarrow$ 0;
    end if
  end while
return $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{\beta})$
end procedure

The objective of improvement procedure (shown in Pseudocode 4) is to reduce the number of equipment units in the system (implying a reduction in costs). This can certainly be achieved due to the integration of both inter-institutional allocation and private service. This procedure requires solving Model S at each iteration. The algorithm stops when no improvement in the objective function can be made or when infeasibility is found. Parameter t^{best} represents the best solution found, and the initial value of $\bar{t}$ is iteratively modified within the main loop. Let $Z(\bar{t})$ denote the corresponding objective function value associated with the complete solution obtained from $\bar{t}$. To evaluate the feasibility of Model A, Equations (46) and (39) are used. To determine from which hospital some capacity is to be removed, Equation (48) is used. In this regard, the hospital with the highest idle capacity such that feasibility is maintained is chosen to have its capacity reduced. The type of equipment with the highest cost-benefit is selected. The selected hospital and type of equipment removed must satisfy feasibility, this is evaluated with Equation (47). When an equipment unit is removed from the system, the capacity is updated, and then Model S is solved again. The new objective function value is evaluated, and if an improvement exists the best solution is updated. The algorithm stops when no improvement is found or when no feasible reduction of capacity can be made.

Evaluation of the Constructive Heuristics

Computer specification and testing environment: All procedures were coded in C++ and compiled with Visual Studio compiler 2012 and run on a PC operating system with 2.70 GHz Intel Core i7-2620M processor and 16 GB 1067 MHz DDR3 of RAM. For the calls to the branch-and-bound method, the CPLEX Studio 12.6.2 callable library from IBM was used.

To evaluate and compare the constructive heuristics, a database of instances taken from Mendoza-Gómez et al. (2016) is used. The network size is given by the number of hospitals in the network (i.e., $|I| = ns$). For each $ns \in \{60, 120, 180, 240, 360\}$, 30 instances were used. These were randomly generated based on real-world data. For each instance, the number of patient acuity levels was randomly selected from 1 to 3 to represent demand that requires emergency, ordinary, or outpatient services. The number of periods was set to 12 to evaluate the monthly demand behavior. The number of equipment types was also defined randomly with values from 1 to 3 for each instance. To evaluate a wide variety of scenarios, the equipment capacity was randomly determined from specific values that range from 90 to 720 services per period. This represents a service rate of 1 service per hour to 3 services per day in a 24-hour service scheme. All costs related to the service were also randomly selected from a wide variety of possible values. To simulate demand behavior, the MRI service was used as an example. The demand for each hospital in each period was randomly generated according to a Weibull probability distribution obtained from Mexico National Health Information System data for MRI services in public hospitals in 2012. For further details, the reader can refer to Mendoza-Gómez et al. (2016).

To assess the quality of solutions found by either heuristic, the relative optimality gap is used. For instances with no optimal solution found with B&B and a three-hour limit of computing time, the best-known lower bound of all the remaining open nodes in the branch-and-node tree is used. The results are presented in Table 1.

As can be seen from the table, there are significant improvements on the average relative gaps and computing times observed by the heuristics when compared to B&B. Relative gaps were very similar between the two constructive methods. The average relative gap for CM1 was 4.60%, and for CM2 was 4.12%. Using a non-parametric test (Mood Median Test), no significant differences were found. Nevertheless, significant differences were found for the computing time. A lower running time was achieved with CM2 with an average time of 6.34 seconds against the 10.84 seconds observed by CM1. Clearly, both heuristics are more efficient than B&B, especially for larger instances. For example, for the 300-node instances, the average relative gap was 43.96% within three hours of computing time for B&B, while CM1 and CM2 produced 5.49% and 5.22%, respectively. These relative gaps were obtained with a significantly lower running time compared to B&B and with average computing times of 27.2 and 16.0 seconds, respectively.

Table 1: Comparison of constructive methods and B&B.

Method	n	Average gap (%)	Minimum gap (%)	Maximum gap (%)	Average time (s)	Minimum time (s)	Maximum time (s)
B&B	60	0.32	0.00	2.31	6,755	18.5	10,800
	120	2.24	0.00	10.62	10,566	3,568	10,800
	180	6.98	0.81	23.91	10,654	8,244	10,800
	240	20.94	3.11	59.43	10,693	7,557	10,800
	300	43.96	6.45	76.65	10,762	9,367	10,800
CM1	60	2.11	0.26	7.91	0.7	0.2	1.7
	120	3.06	0.66	16.87	3.2	0.9	8.1
	180	3.58	1.09	8.86	8.4	2.1	24.7
	240	4.59	0.98	12.12	14.7	3.7	36.3
	300	5.49	1.75	10.52	27.2	5.8	59.0
CM2	60	2.04	0.0	6.67	0.5	0.2	1.1
	120	2.46	0.81	6.50	1.8	0.7	3.2
	180	3.38	0.56	10.07	5.0	1.3	10.9
	240	3.36	0.70	9.43	8.5	2.6	18.5
	300	5.22	1.08	10.03	16.0	4.7	38.9

4.2 Iterated Greedy Algorithm

The iterated greedy algorithm (IGA) is a simple stochastic local search method that generates a sequence of solutions by iterating over a greedy construction heuristic using destruction and reconstruction mechanisms. Pseudocode 5 shows our implementation of the IGA. The method takes the maximum number of iterations (iteration_limit) and the destruction parameter ρ as input. The destruction phase (Step 7) removes some elements from the incumbent solution according to ρ. In the reconstruction phase (Step 8), starting from a partial solution, a new candidate solution is created by reconstructing a complete solution using a greedy constructive heuristic. Any of the two construction methods outlined before can be applied here, starting the construction from a partial solution. Once the candidate solution has been completed (Step 9), an acceptance criterion is applied to decide whether the constructed solution should replace the incumbent solution (Step 11). The best-known solution is updated if necessary (Steps 12-14). The process iterates between these two phases until a stopping criterion is met. In this case we use a limit on the number of iterations.

The parameter ρ is used to indicate the percentage of elements in the solution to be destroyed. In this problem, ρ indicates the percentage of the total number of equipment units in the system that are "removed" or unassigned. The destruction method is displayed in Pseudocode 6. The number of equipment units that are necessarily required (indicated by parameter H) are not considered as candidate elements to be removed (Step 2). A candidate list of pairs (j, l), CL, with positive removable capacity F_{jl}, is formed (Step 4). Then, each pair (j, l) is stored in CL such that every possible equipment unit has equal chance of being removed. At each iteration an equipment unit is removed randomly from the solution and the candidate list (CL) is updated.

Pseudocode 5 Iterated Greedy Algorithm

procedure ITERATED_GREEDY(iteration_limit, ρ)
  $t_0 \leftarrow$ Constructive_Method( ) ;
  Solve_Subproblem(t_0);
  $t^{\text{best}} \leftarrow t_0$;
  for ($i = 1, \ldots,$ iteration_limit) **do**
    $\bar{t} \leftarrow t_0$;
    $\bar{t} \leftarrow$ Destruction_Method($\bar{t}$, ρ);
    $\bar{t} \leftarrow$ Constructive_Method($\bar{t}$);
    Solve_Subproblem($\bar{t}$);
    if ($Z(\bar{t}) < Z(t_0)$) **then**
      $t_0 \leftarrow \bar{t}$;
      if ($Z(\bar{t}) < Z(t^{\text{best}})$) **then**
        $t^{\text{best}} \leftarrow \bar{t}$;
      end if
    end if
  end for
return (t^{best})
end procedure

Pseudocode 6 Destruction method

procedure DESTRUCTION_METHOD($\bar{t}$, ρ)
  $F_{jl} \leftarrow \bar{t}_{jl} - H_{jl}$;
  $\eta \leftarrow \lceil \rho \cdot \sum_{j \in G} \sum_{l \in L} F_{jl} \rceil$;
  CL $\leftarrow$ all pairs (j, l) with $F_{jl} > 0$;
  for ($i = 1, \ldots, \eta$) **do**
    Choose $(j^*,\ l^*)$ randomly from CL;
    $\bar{t}_{j^*l^*} \leftarrow \bar{t}_{j^*l^*} - 1$;
    CL $\leftarrow$ CL $\setminus \{(j^*,\ l^*)\}$;
  end for
return $(\bar{t})$
end procedure

Calibration of IGA

The IGA requires two parameters, one to define a level of solution destruction (ρ) and the number of main iterations. To identify an adequate value for each parameter, an experiment was conducted with the same samples. Different values of ρ were tested in order to identify the best value that generates the best improvement for the solutions. Since it is not yet known the best number of iterations required, in this experiment 100 iterations were considered for all instances. To compare both constructive methods, the same initial solution generated from CM1 was used. These initial methods generated initial solutions with an average relative gap of 3.8% in 13.1 seconds.

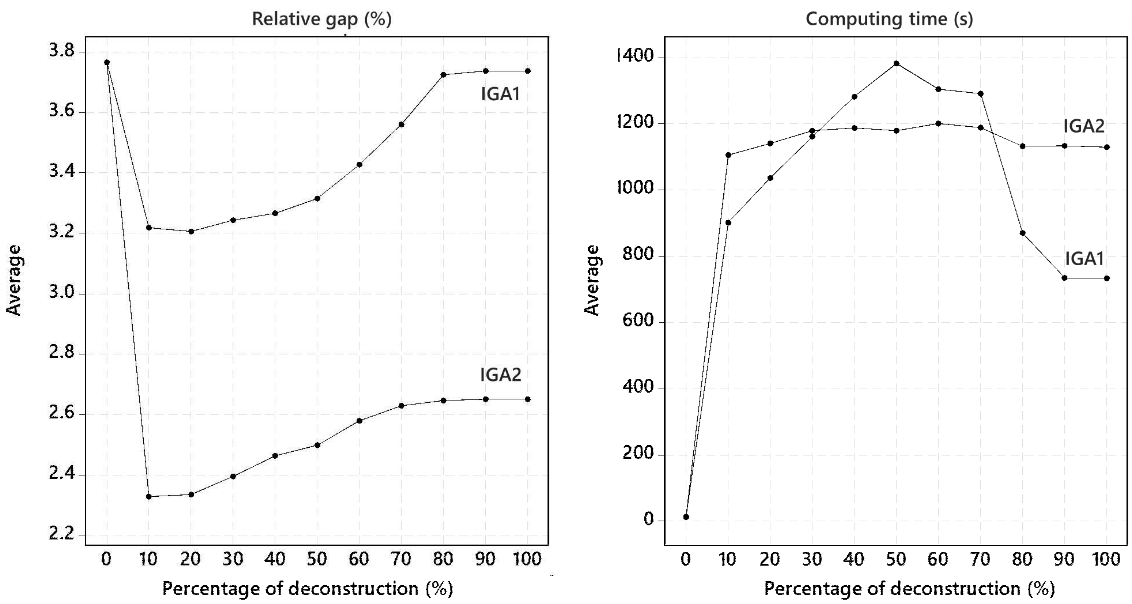

Figure 2: Calibration of ρ in the IGA.

We name IGA1 and IGA2 the iterated greedy algorithm using CM1 and CM2, respectively. The results are shown in Figure 2. In the left-hand-side plot, the average relative gaps are compared for each value of ρ. The best relative gap was achieved with a ρ equal to 10% and using IGA2 with an average relative gap of 2.33% in an average running time of 1,106 seconds. However, using a ρ equal to 20% generated an increment of only 0.0067% in the average relative gap but increased the average running time by 85 seconds. The best results using IGA1 were achieved with a ρ equal to 20% and with an average relative gap of 3.21% in an average of 1,036 seconds. We conclude that IGA2 provides significantly better results than IGA1.

In addition, in the right-hand-side plot, we observe that increasing the value of ρ generates an increased running time. IGA2 presented more stable computing times, while IGA1 reached its highest peak with a ρ equal to 50%, and after that it decreased. If we compare the computing time for the best results of each strategy, the average computing time of IGA1 is only 20 seconds lower than IGA2. With these experiments, we conclude that the best strategy is to use IGA2 with a value of ρ between 0.10 and 0.20.

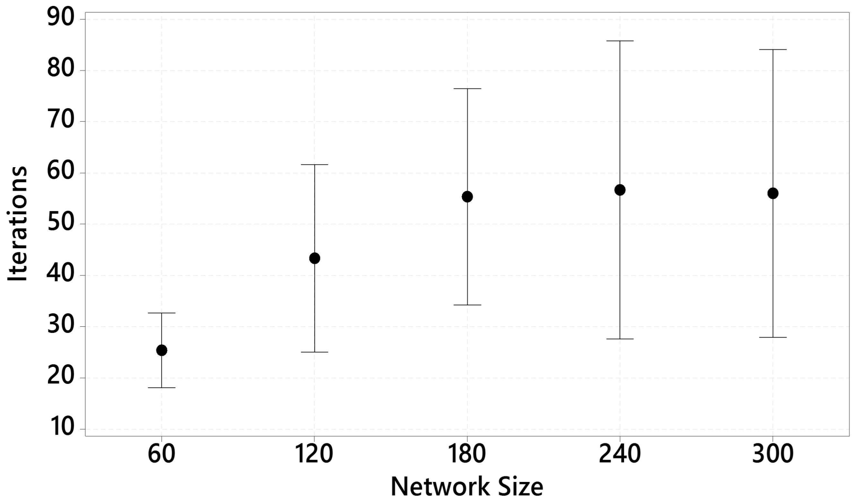

Figure 3: Interval plot with a confidence interval of 95% for the number of suggested iterations.

To identify the number of iterations required by the IGA2, all samples were solved using up to 200 iterations with $\rho = 0.20$. In a first analysis, the total number of iterations required to find the best solution for each instance was recorded. In general, most of the instances (82%) required less than 100 iteration to find the best solution. In an additional experiment, the appropriate number of iterations required for each instance was intuitively identified, taking into account the improvement achieved compared with the running time. An interval was identified for the number of iterations required for each instance size (30 samples) with a confidence level of 95%. The interval plot is presented in Figure 3. Considering both previous analyses, we considered that an adequate number of iterations for instances up to 300 nodes could be around 100 iterations.

4.3 Variable Neighborhood Descent

VNS is a metaheuristic based on a systematic change of neighborhood within the search. The VND method is obtained if the change of neighborhoods is performed in a deterministic way. Basically, several different neighborhoods are explored in order, typically from the smallest and fastest, to evaluate the slowest and largest one. The process iterates over each neighborhood while improvements are found, doing local search until meeting local optima at each neighborhood. Only strictly better solutions are accepted after each neighborhood search. Pseudocode 7 depicts the VND, where $\bar{t}$ again denotes the compact version of a complete feasible solution given by $(\bar{t}, \bar{y}, \bar{x}, \bar{s}, \bar{\alpha}, \bar{v}, \bar{\beta})$.

Pseudocode 7 Variable Neighborhood Descent

procedure VND($\bar{t}$)
    $t^{\text{best}} \leftarrow \bar{t}$; $k \leftarrow 1$;
    while ($k \leq k^{\max}$) **do**
        $\bar{t} \leftarrow$ BestNeighbor($\mathcal{N}_k(t^{\text{best}})$);
        if ($Z(\bar{t}) < Z(t^{\text{best}})$) **then**
            $t^{\text{best}} \leftarrow \bar{t}$;
            $k \leftarrow 1$;
        else
            $k \leftarrow k + 1$;
        end if
    end while
return ($\bar{t}$)
end procedure

Proposed Neighborhoods

Four neighborhoods are proposed for this problem. These neighborhoods are defined by movements of equipment units in the system. They are described below:

1. *Equipment type swap*: The move $move1(i, l, l^*)$ is defined as replacing one equipment unit of type l for another unit of different type l^* in the same hospital i. Then $\mathcal{N}_1(\bar{t})$ is the set of neighbors reachable from $\bar{t}$ by performing all possible moves $move1(i, l, l^*)$. The following are some important computational considerations:

    - A hospital $i \in G$ is a candidate for this move if

    $$\sum_{l \in L} (\bar{t}_{il} - H_{il}) > 0. \tag{49}$$

    - An equipment type l in a given hospital i is feasible for this move if $\bar{t}_{il} - H_{il} > 0$.
    - For a selected hospital i with an equipment of type l, equipment type l^* is feasible for this move if all of these conditions are met:

$$\max_{n \in N} \sum_{u \in U} D^u_{in} \leqslant (W_i - EC_l + EC_{l^*}) \cdot \sigma_{k_i}$$

$$\max_{n \in N} \sum_{u \in U} \sum_{j \in G_i} D^u_{jn} \leqslant \left(\sum_{j \in G_i} W_j - EC_l + EC_{l^*} \right) \cdot \sigma_{k_i} \tag{50}$$

$$\sum_{u \in U} \sum_{j \in G_i} \sum_{n \in N} D^u_{jn} \cdot \delta_k \leqslant \left(\sum_{j \in G_i} W_j - EC_l + EC_{l^*} \right) \cdot |N| \tag{51}$$

- This move is meaningful only for instances with $|L| > 1$.

2. *Inter-institutional hospital total capacity exchange*: The move $move2(i, j)$ consists of transferring all current allocation from hospital i to another hospital j of the same institution with nonzero allocation. Then $\mathcal{N}_2(\bar{t})$ is the set of neighbors reachable from $\bar{t}$ by performing all possible moves $move2(i, j)$. The following are some important computational considerations:

   - A hospital $i \in G$ is a "donor" candidate for this move if $\bar{y}_i = 1$ and $\sum_{l \in L} H_{il} = 0$.
   - A hospital $j \in G_i$ is a "recipient" candidate for this move if $\bar{y}_j = 1$.

3. *Inter-institutional hospital unit capacity balancing*:

   The move $move3(i, j, l)$ consists of transferring one unit of capacity of equipment type l from hospital i to another hospital j of the same institution with a lower number of units assigned. Then $\mathcal{N}_3(\bar{t})$ is the set of neighbors reachable from $\bar{t}$ by performing all possible moves $move3(i, j, l)$. This move is not considered when all hospitals have the same number of equipment units. The following are some important computational considerations:

   - For any $k \in K$, a hospital $i \in G$ is a *donor* candidate for this move if $W_i = \max_{j \in G^k} \{W_j\}$ and constraints (49) are met.
   - For a given donor hospital i, a hospital $j \in G_i$ is a valid *recipient'* candidate for this move if $\bar{y}_j = 1$.
   - For chosen donor and recipient hospitals i and j, an equipment of type $l \in L$ is a candidate for this move if (50), (51), and the following conditions are met:

$$\max_{n \in N} \sum_{u \in U} D_{in}^u \leqslant (W_i - EC_l) \cdot \sigma_{k_i} \quad \text{if} \quad W_i - EC_l > 0 \tag{52}$$

4. *Hospital capacity exchange*: The move $move4(i, j, l)$ consists of transferring one unit of capacity of equipment type l from a hospital i to a hospital j from a different institution. Then $\mathcal{N}_4(\bar{t})$ is the set of neighbors reachable from $\bar{t}$ by performing all possible moves $move4(i, j, l)$. The following are some important computational considerations:

   - For any $k \in K$, a hospital $i \in G^k$ is a valid "donor" candidate for this move if (49) is met.
   - For a given donor hospital i, a hospital j is a valid *recipient* candidate if $G_j \neq G_i$ and $\bar{y}_j = 1$.
   - For a chosen donor hospital i, an equipment of type $l \in L$ is a valid candidate for this

move if (52) and the following two conditions are met:

$$\max_{n \in N} \sum_{u \in U} \sum_{j \in G_i} D_{jn}^u \leqslant \left(\sum_{j \in G_i} W_j - EC_l \right) \cdot \sigma_{k_i}$$

$$\sum_{u \in U} \sum_{j \in G_i} \sum_{n \in N} D_{jn}^u \cdot \delta_k \leqslant \left(\sum_{j \in G_i} W_j - EC_l \right) \cdot |N|$$

Neighborhood Assessment

The proposed neighborhoods are empirically assessed by individually running each of them within a hill-climbing local search (LS) strategy. The 150 instances previously tested are used in this experiment. The CM1 method was used to get initial solutions.

The results are presented in Table 2, where the first column LSi represents neighborhood $\mathcal{N}_i$. The final (after LS) average relative gaps are presented in column 2. The third column displays the average of the relative improvements of all instances (IMP) for each local search procedure. The fourth and fifth columns indicate the average and maximum running time, respectively. The final average relative gap for each network size is presented in the last five columns.

Table 2: Individual neighborhood evaluation, initial relative gap = 3.77%.

LS	Final gap (%)	IMP (%)	Ave. time (s)	Max. time (s)	Average final gap (%)				
					60	120	180	240	300
LS1	3.30	12.28	0.9	15.0	1.93	2.62	3.12	3.92	4.93
LS2	3.61	4.17	0.4	6.0	1.70	2.96	3.49	4.48	5.42
LS3	2.85	24.40	18.1	198.0	1.61	2.32	2.69	3.43	4.18
LS4	2.47	34.29	184.1	1,578.4	1.43	1.99	2.40	2.88	3.66

As can be seen from Table 2, LS4 presented the best average improvement but also the highest running time. LS3 produced the second best average improvement but with a lower running time compared to LS4. LS1 and LS2 generated moderate improvements but had the lowest running times. In general, large instances required more computing time in all local search schemes and were more difficult to solve.

Assessment of VND

The next step is to integrate a sequence of these neighborhoods in the VND method. This aims to intensify the search of solutions in a specific local region obtained from the IGA knowing that a good solution is found in this region. Some experiments were performed to select the neighborhoods to be integrated in the VND. The results of five different strategies are shown in Table 3. The second column shows the neighborhood ordered sequence in the VND. For instance for VND1, the three neighborhoods used were $\mathcal{N}_1$, $\mathcal{N}_2$, and $\mathcal{N}_3$. The remaining columns are similar to those in Table 2.

Since the LS4 presented the highest running times, VND2 and VND4 are the ones with the

Table 3: VND evaluation, initial relative gap = 3.77%.

VND	$\mathcal{N}$ Order	Final gap (%)	IMP (%)	Ave. time (s)	Max. time (s)	Average final gap (%) 60	120	180	240	300
VND1	(1, 2, 3)	2.30	38.90	21.8	218.8	1.04	1.86	2.23	2.71	3.68
VND2	(1, 2, 3, 4)	2.04	48.87	263.5	2,660.5	0.91	1.62	1.89	2.44	3.34
VND3	(2, 1, 3)	2.31	38.74	53.1	420.5	1.05	1.88	2.21	2.72	3.68
VND4	(2, 1, 3, 4)	2.05	45.70	257.4	2,139.6	0.91	1.60	1.88	2.45	3.39
VND5	(2, 3)	2.45	35.07	17.1	178.8	1.45	1.97	2.29	2.81	3.73

highest running times. However, they are the ones with the highest improvements. The objective is to implement a VND at each iteration of the IGA. Therefore, strategies VND1 and VND5 are good candidates because they offer a good compromise between quality and running time, but VND1 was selected for implementation in the IGA-VND metaheuristics because it has a better improvement than VND5 and a slight increase in running time.

4.4 Iterated Greedy Algorithm with Variable Neighborhood Descent

Pseudocode 8 displays the proposed IGA-VND metaheuristic. It is essentialy the IGA method enhanced with a VND local search phase everytime a new solution is found after the destruction and reconstruction procedures. The IGA component may be viewed as a diversification mechanism, and the VND component as an intensification method. As an additional strategy of improvement, a LS4 is applied when the best solution is improved (Step 15).

Table 4 presents some experiments of the metaheuristic with different parameter values and components. The first column represents the name of the experiment. The second column describes the completed method. For instance, for E3 the CM1 is used as the initial solution for IGA2; at each iteration of IGA2 a VND1 is applied, and at each improvement a LS4 is performed. The third column corresponds to the deconstruction parameter. The fourth column shows the number of iterations for the IGA. The following six columns show the average relative gaps for each network size and the global average. In the last column the average running time is shown.

We can observe in Table 2 that LS4 produced the best improvement with respect to all other neighborhoods but had the longest running time. The inclusion of LS4 in the VND significantly increases the computing time of the IGA, according to Table 3. However, if this strategy is used only when the best solution is updated, the computing time is not be considerably affected. This is shown in the experiments E1 and E2 of Table 4; in average, better relative gaps are achieved with E2, with an increase in the average running time of only 3 seconds.

An additional improvement in the IGA is to provide an alternative initial solution method. The use of CM1 to generate an initial solution provides better improvements when IGA2 is performed, as shown in experiments E2 vs. E3 of Table 4. A decrease of 0.29% in the average relative gap and a decrease of 164 seconds in the average running time are achieved. In experiment E3 vs. E4, it is observed that only 50 iterations are required for a significant improvement, and from 50 to

Pseudocode 8 Iterated Greedy Algorithm with VND

```
procedure ITERATED_GREEDY_VND( iteration_limit, ρ)
    t_0 ← Constructive_Metod( ) ;
    t_0 ← VND( t_0 ) ;
    Solve_Subproblem( t_0 );
    t^best ← t_0;
    for ( i = 1, . . . , iteration_limit ) do
        t̄ ← t_0;
        t̄ ← Destruction_Method( t̄, ρ );
        t̄ ← Constructive_Method( t̄ );
        t̄ ← VND( t̄ );
        Solve_Subproblem( t̄ );
        if ( Z(t̄) < Z(t_0) ) then
            t_0 ← t̄;
            if ( Z(t̄) < Z(t^best) ) then
                t̄ ← LS( t̄ );
                t_0 ← t̄;
                t^best ← t̄;
            end if
        end if
    end for
return ( t^best )
end procedure
```

Table 4: Overall assessment of IGA-VND strategies.

Exp.	Method	ρ	Iter	Average relative gap for NS (%)						Average time (s)
				60	120	180	240	300	Global	
E1	IGA2_VND1	0.20	50	0.87	1.44	1.64	2.26	3.63	1.97	1,902
E2	IGA2_VND1_LS4	0.20	50	0.83	1.38	1.67	2.21	3.61	1.94	1,907
E3	**CM1_IGA2_VND1_LS4**	**0.20**	**50**	**0.68**	**0.97**	**1.49**	**2.03**	**3.06**	**1.65**	**1,743**
E4	CM1_IGA2_VND1_LS4	0.20	100	0.67	0.97	1.45	2.02	3.06	1.64	2,495
E5	CM1_IGA2_VND1_LS4	0.10	50	0.67	1.05	1.58	2.04	3.11	1.69	1,550

100 iterations, only a 0.01% of relative decrease is made with an additional running time of 752 seconds. In Figure 2, it was observed that relative gaps for values of ρ between 0.10 and 0.20 were similar. In the comparison E3 vs. E5, both values are tested, where the best results were achieved with a ρ equal to 0.20.

With these experiments, we conclude that the best performance of the metaheuristic is achieved with CM1_IGA2_VND1_LS4, with 50 iterations and a destruction parameter percentage of 20%.

In order to assess the contribution of each of the components of the IGA-VND metaheuristic, we run the heuristic under different algorithmic conditions, that is, omitting one component at a time. The results are shown in Table 5, where each "different heuristic" is shown in each row. The second column displays the component being omitted. For instance, the first row is the full metaheuristic, the second row represents the same heuristic but with the LS4 component omitted, and so on.

Table 5: Assessment of individual components.

Method	Omitted component	Average relative gap for NS (%)							Average time (s)	
		60	120	180	240	300	Global	Shift	Global	Shift
CM1_IGA2_VND1_LS4	Neither	0.68	0.97	1.49	2.03	3.06	1.65		1,743	
CM1_IGA2_VND1	LS4	0.73	1.06	1.54	2.08	3.12	1.71	+0.06	1,781	+ 38
IGA2_VND1_LS4	CM1	0.83	1.38	1.67	2.21	3.61	1.94	+0.29	1,907	+164
CM1_IGA2_LS4	VND1	1.13	1.42	1.98	2.67	3.66	2.17	+0.52	603	−1,140
CM1_VND1_LS4	IGA2	0.96	1.72	2.10	2.62	3.54	2.19	+0.54	46	−1,697

The first thing to notice is that the best results are obtained when no component is removed; that is, each component of the algorithm brings some benefit to the table. Now, to identify which components are more critical, as can be seen from the table, the best contribution in the solution quality is provided by the iterated greedy algorithm, which reduces the average relative gap by 0.54% but also requires an increase of 1,697 seconds in the average running time. The second best improvement is achieved by the VND1 and with a decrease in the average relative gap of 0.52% with an increment of 1,140 seconds in the average running time. The use of CM1 as the initial solution provides an improvement of 0.29% in the average relative gap and an increase of 164 seconds. Finally, the LS4 generated a moderate of 0.06% in the relative gap with an increase of only 38 seconds in the running time. When the IGA component is omitted, the resulting algorithm is essentially a VND heuristic. As seen and mention before, the solution quality may not be as good as the complete metaheuristic; however, it is very fast compared to the other choices. As a conclusion, this can be seen as a double contribution, that is, an IG-based metaheuristic delivering the highest quality results, and an alternate VND-based heuristic delivering solutions very quickly.

Finally, Table 6 displayes the results of the metaheuristic compared with B&B with a time limit of 3 hours. It is important to note that to the best of our knowledge, no other heuristic or exact methods have been developed for this problem. The methods developed for similar problems are not applicable at all for this problem given its very unique features. It is clearly observed that for instances of 120 hospitals and more, the metaheuristic is significantly better with respect to both solution quality and computing time, as expected. Furthermore, the proposed metaheuristic finds near-optimal solutions, obtaining average relative optimality gaps of less than 3.06% for the largest set and less than 2.03% for the rest.

Table 6: Comparison between IGA-VND and B&B.

Method	Average relative gap for NS (%)					Average run-time for NS (s)				
	60	120	180	240	300	60	120	180	240	300
B&B	0.32	2.24	6.98	20.94	43.96	6,755	10,566	10,654	10,693	10,761
CM1_IGA2_VND1_LS4	0.68	0.97	1.49	2.03	3.06	105	562	1,720	3,529	6,542

5 Conclusions

This paper proposes a hybrid IGA-VND algorithm to solve medium- to large-scale instances of a location-allocation problem for specialized medical service across public hospital networks. The method is based on an intelligent exploitation of problem structure as we showed how to decompose this structure by fixing first-level location variables, leaving a transshipment-like problem in the allocation of the second-level subproblem. This property was taken advantage of in the constructive and local search components. The contribution of our work included the design and development of specific components tailored-made for this specific application, such as greedy construction heuristics, several local search schemes, a Variable Neighborhood Search, all cast into an Iterated Greedy Algorithm metaheuristic.

Our computational study revealed the effectiveness of the proposed approach and each of its components. When assessing each component individually, we found that the IGA and VND methods provide the largest benefit to the overall methodology. Since no other heuristic methods exist for this problem, and other methods developed for similar problems are not quite applicable to this problem given its unique features, a comparison was made to a branch-and-bound method (implemented in CPLEX). Except for the 60-node instances, the proposed metaheuristic clearly found solutions of significantly better quality than those found by the exact method. In fact, using the best lower bound found by branch and bound as a measure for computing relative optimality gaps, our method was empirically shown to find near-optimal solutions, obtaining relative optimally gaps of less than 3.06% for the largest instances.

Some additional opportunities for extending this research are to study new characteristics evaluated in recent studies, such as the uncertainty in demand or supply of services, the use of a hierarchical hospital structure, multiple-services evaluation, and evaluation of lost demand or patient dissatisfaction. Furthermore, the formulated problem focuses on helping decision makers be strategic about infrastructure planning, but a second problem needing to be solved is associated with the operative decisions. In this sense, it is worthwhile to investigate a methodology to evaluate the best strategies for the programming of services that require a coordination between departments, staff, and institutions. To this end, several of the ideas, such as the problem decomposition, or components developed in this research, may prove useful to some of these problems.

Acknowledgments: The first author was financially supported by a scholarship for doctoral studies by the Mexican National Council for Science and Technology (CONACYT). The research of the second author is supported by CONACYT (grant CB2011-1-166397) and by UANL through its Scientific and Technological Research Suport Program (grant UANL-PACYT CE331-15).

References

Ahmadi-Javid, A., Seyedi, P., and Syam, S. S. (2017). A survey of healthcare facility location. *Computers & Operations Research*, 79:223–226.

Ayvaz, N. and Huh, W. T. (2010). Allocation of hospital capacity to multiple types of patients. *Journal of Revenue and Pricing Management*, 9(5):386–398.

Chauhan, A. and Singh, A. (2016). Healthcare waste management: A state-of-the-art literature review. *International Journal of Environment and Waste Management*, 18(2):120–144.

Coté, M. J., Syam, S. S., Vogel, W. B., and Cowper, D. C. (2007). A mixed integer programming model to locate traumatic brain injury treatment units in the department of veterans affairs: A case study. *Health Care Management Science*, 10(3):253–267.

Crainic, T. G., Gendreau, M., Hansen, P., Hoeb, N., and Mladenović, N. (2004). Cooperative parallel variable neighborhood search for the p-median. *Journal of Heuristics*, 10(3):293–314.

Daskin, M. S. and Dean, L. K. (2004). Location of health care facilities. In Brandeau, M. L., Sainfort, F., and Pierskalla, W. P., editors, *Operations Research and Health Care: A Handbook of Methods and Applications*, chapter 3, pages 43–76. Springer, New York.

Diakova, Z. and Kochetov, Y. (2012). A double VNS heuristic for the facility location and pricing problem. *Electronic Notes in Discrete Mathematics*, 39:29–34.

García-López, F., Melián-Batista, B., Moreno-Pérez, J. A., and Moreno-Vega, J. M. (2002). The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, 8(3):375–388.

Hansen, P. and Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350.

Hoos, H. H. and Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications.* Morgan Kaufmann, San Francisco.

Ljubić, I. (2007). A hybrid VNS for connected facility location. In Bartz-Beielstein, T., Blesa Aguilera, M. J., Blum, C., Naujoks, B., Roli, A., Rudolph, G., and Sampels, M., editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer-Verlag, Berlin, Germany.

Lourenço, H. R., Martin, O. C., and Stützle, T. (2003). Iterated local search. In Glover, F. and Kochenberger, G. A., editors, *Handbook of Metaheuristics*, chapter 11, pages 320–353. Springer, Boston.

Mahar, S., Bretthauer, K. M., and Salzarulo, P. A. (2011). Locating specialized service capacity in a multi-hospital network. *European Journal of Operational Research*, 212(3):596–605.

Marić, M., Stanimirović, Z., and Božović, S. (2013). Hybrid metaheuristic method for determining locations for long-term health care facilities. *Annals of Operations Research*, 227(1):3–23.

McLafferty, S. and Broe, D. (1990). Patient outcomes and regional planning of coronary care services: A location-allocation approach. *Social Science and Medicine*, 30(3):297–305.

Mendoza-Gómez, R., Ríos-Mercado, R. Z., and Valenzuela, K. B. (2016). Efficient planning of specialized diagnostic services in a segmented healthcare system. Technical report PISIS-2016-01, Graduate Program in Systems Engineering, Universidad Autónoma de Nuevo León.

Mestre, A. M., Oliveira, M. D., and Barbosa-Póvoa, A. P. (2015). Location-allocation approaches for hospital network planning under uncertainty. *European Journal of Operational Research*, 240(3):791–806.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers and Operations Research*, 24(11):1097–1100.

Mladenović, N. and Hansen, P. (2003). Solving the p-center problem by tabu search and variable neighborhood search. *Networks*, 42(1):48–64.

Pan, Q. K. and Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, 44(1):41–50.

Quevedo-Orozco, D. R. and Ríos-Mercado, R. Z. (2015). Improving the quality of heuristic solutions for the capacitated vertex p-center problem through iterated greedy local search and variable neighborhood descent. *Computers and Operations Research*, 62:133–144.

Rais, A. and Viana, A. (2011). Operations research in healthcare: A survey. *International Transactions in Operational Research*, 18(1):1–31.

Rego, N. and de Sousa, J. P. (2009). Supply chain coordination in hospitals. In Camarinha-Matos, L. M., Paraskakis, I., and Afsarmanesh, H., editors, *Leveraging Knowledge for Innovation in Collaborative Networks*, volume 307 of *IFIP Advances in Information and Communication Technology (IFIPAICT)*, pages 117–127. Springer, Berlin, Germany.

Ribas, I., Companys, R., and Tort-Martorell, X. (2011). An iterated greedy algorithm for the flowshop scheduling problem with blocking. *Omega*, 39(3):293–301.

Ruiz, R. and Stützle, T. (2006). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033–2049.

Ruiz, R. and Stützle, T. (2008). An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143–1159.

Ruth, R. J. (1981). A mixed integer programming model for regional planning of a hospital inpatient service. *Management Science*, 27(5):521–533.

Stummer, C., Doerner, K., Focke, A., and Heidenberger, K. (2004). Determining location and size of medical departments in a hospital network: A multiobjective decision support approach. *Health Care Management Science*, 7(1):63–71.

Syam, S. S. and Coté, M. J. (2010). A location-allocation model for service providers with application to not-for-profit health care organizations. *Omega*, 38(34):157–166.

Syam, S. S. and Coté, M. J. (2012). A comprehensive location-allocation method for specialized healthcare services. *Operations Research for Health Care*, 1(4):73–83.

Tlahig, H., Bouchriha, A. J. H., and Ladet, P. (2013). Centralized versus distributed sterilization service: A locationallocation decision model. *Operations Research for Health Care*, 2(4):75–85.

Yuan, Z., Fügenschuh, A., Homfeld, H., Balaprakash, P., Stützle, T., and Schoch, M. (2008). Iterated greedy algorithms for a real-world cyclic train scheduling problem. In Blesa, M. J., Blum, C., Cotta, C., Fernández, A. J., Gallardo, J. E., Roli, A., and Sampels, M., editors, *Hybrid Metaheuristics*, volume 5296 of *Lecture Notes in Computer Science*, pages 102–116. Springer.

Zarrinpoor, N., Fallahnezhad, M. S., and Pishvaee, M. S. (2017). Design of a reliable hierarchical location-allocation model under disruptions for health service networks: A two-stage robust approach. *Computers & Industrial Engineering*, 109:130–150.