Publié par :
Published by:
Publicación de la:

Faculté des sciences de l'administration
2325, rue de la Terrasse
Pavillon Palasis-Prince, Université Laval
Québec (Québec) Canada G1V 0A6
Tél. Ph. Tel. : (418) 656-3644
Télec. Fax : (418) 656-7047

Disponible sur Internet :
Available on Internet
Disponible por Internet :

**http://www4.fsa.ulaval.ca/la-recherche/publications/documents-de-travail/**

## DOCUMENT DE TRAVAIL 2016-012

A Simultaneous Facility Location and Vehicle Routing Problem Arising in Health Care Logistics in the Netherlands

Marjolein VEENSTRA
Kees Jan ROODBERGEN
Leandro C. COELHO
Stuart X. ZHU

Document de travail également publié par le Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, sous le numéro CIRRELT-2016-44

**Août 2016**

Dépôt legal – Bibliothèque et Archives nationales du Québec, 2016
Bibliothèque et Archives Canada, 2016

ISBN 978-2-89524-439-4 (PDF)

# A Simultaneous Facility Location and Vehicle Routing Problem Arising in Health Care Logistics in the Netherlands

**Marjolein Veenstra**[1], **Kees Jan Roodbergen**[1], **Leandro C. Coelho**[1,2,*], **Stuart X. Zhu**[1]

[1] University of Groningen, Faculty of Economics and Business, Department of Operations, Nettelbosje 2, 9747 AE Groningen, The Netherlands

[2] Interunivesrity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325, rue de la Terrasse, Université Laval, Québec, Canada, G1V 0A6

**Corresponding author: leandro.coelho@cirrelt.ca*

## ABSTRACT

This paper introduces a simultaneous facility location and vehicle routing problem that arises in health care logistics in the Netherlands. In this problem, the delivery of medication from a local pharmacy can occur via lockers, from where patients that are within the coverage distance of a locker can collect their medication, or by home delivery. The aim of the problem is to determine which lockers from a set of potential locker locations to open and to generate routes that visit the opened lockers and routes that visit the patients that are not covered by the opened lockers, while minimizing the routing costs and the opening costs of the lockers. We formally define this problem and solve it by applying a branch-and-bound algorithm to this mathematical formulation. Moreover, we propose a fast hybrid heuristic to solve the problem. Extensive computational results are given on a randomly generated instance set and an instance set inspired by practice from an industrial partner. Our results indicate that our heuristic is able to consistently outperform the exact method and that its solutions are extremely robust. We provide important business insights on several parameters of the problem.

**Keywords:** Routing, facility location, health care logistics, hybrid heuristic.

**Acknowledgments:** This project was funded by the Dutch Institute for Advanced Logistics (Dinalog) and partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764, Netherlands Organisation for Scientific Research under Grant 040.03.021, and Royal Netherlands Academy of Arts and Sciences under Grant 530-4CDI18. This support is greatly acknowledged. We thank our industrial partner for providing data.

# 1 Introduction

Health care insurance companies in the Netherlands, induced by government requirements, aim to contain costs by contracting hospitals for complete treatment plans rather than for individual activities. Currently, this type of contracting is only used for a limited number of diseases, but the system is expected to be significantly expanded in the near future. A treatment plan encompasses all activities inside the hospital, but also the medication after the patient has returned home. The hospital is paid a fixed amount for an entire treatment plan by the insurance company, and the hospital will have to purchase and distribute the medicines to the patients. Therefore it is in the interest of the hospitals to control medication costs, which causes them to prescribe medicines in a cost-effective manner and to engage in negotiations on purchase and distribution prices.

It is, however, difficult for hospitals to negotiate with all 1917 local pharmacies in the country on pricing and distribution of medicines to patients. For this reason, hospitals tend to contract a single company to supply the medicines and to arrange for the medicines to be handed to all involved patients. Our industrial partner (anonymous, abbreviated as DMS) is a company that offers such services to hospitals. DMS runs various activities, including a wholesale operation for medicines and 60 pharmacies. As DMS only has a limited number of pharmacies, many patients live too far from the nearest pharmacy and therefore cannot be expected to pick up their medicines themselves. In order to address this issue, DMS plans to install pick-up lockers from which patients can retrieve their medicines. Such lockers prevent DMS from having to open more pharmacies that may not be profitable, while at the same time no home deliveries have to be made. Of course, lockers will only be cost-effective if sufficient patients live nearby, as otherwise home delivery would be preferable. DMS came to us for support in designing their network, such that an effective trade-off can be made between home delivery of medicines and installing lockers.

Within this context, a local pharmacy acts as a depot from where the medications are distributed to patients, and where a fleet of vehicles is available. Lockers can be opened at a set of potential locations. We consider a set of patients that need repeated medication. When a locker is opened within a prespecified distance to a patient, their medication will be distributed via the locker. All other patients will be visited by a vehicle and receive their medications at home. Routes for replenishing lockers and routes for delivering medication to patients are separated. This separation of routes is due to several practical considerations. First, it must be guaranteed that lockers are replenished before a given time of the day. This is more difficult to achieve when combining patients in the same route, since the interaction time with the patients upon delivery is difficult to predict. Second, the ideal delivery period differs between lockers (before people return home from work) and home delivery (after people return home from work). Third, drivers that perform home delivery must have better interpersonal skills. Generally, the same reasons also cause a significant preference of the logistics company for delivering to lockers rather than to patients. The goal of the problem is to determine which lockers to open, and to generate the vehicle routes that visit the lockers and the vehicle routes that visit the patients. The objective is to minimize total costs, consisting of routing costs and opening costs of the lockers.

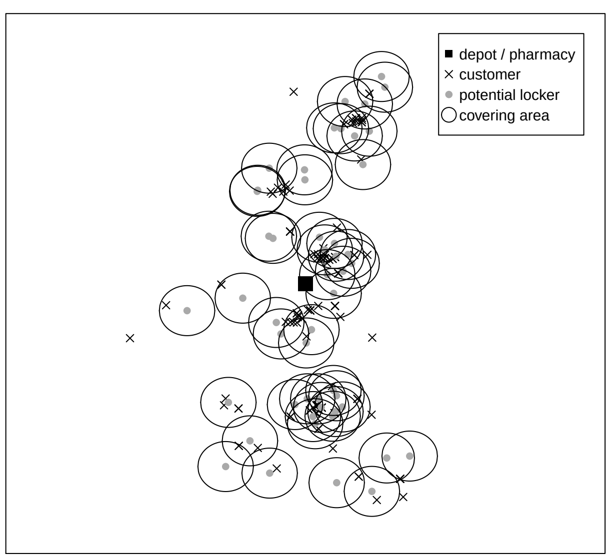


**Figure 1:** Example of an instance based on data from DMS

Figure 1 shows an example derived from data of DMS, our industrial partner (for details we refer to Section 4.2). The figure depicts a depot (marked by a square), a set of patients (marked with crosses), and a set of potential locker locations (marked by gray circles) with the area they cover (marked by the larger circles around them).

The problem setting as described before also arises in the location of pick-up lockers for e-commerce operations. However, to the best of our knowledge, this configuration has not been studied before. Nevertheless, related literature exists, namely the location-routing problem (LRP), the two-echelon location-routing problem (2E-LRP), and a number of covering problems. In the classical LRP (see Nagy and Salhi [17] and Prodhon and Prins [19] for surveys), the objective is to open some depots, assign customers to the opened depots, and design vehicle routes from the opened depots to the customers such that the total costs consisting of the depot costs, the fixed costs per vehicle used and the routing costs are minimized. A special case of the LRP that incorporates subcontracting options is introduced by Stenger et al. [21]. In this problem, self-operated and subcontracted depots can be opened. The customers that are assigned to the subcontracted depots are served by an external carrier and therefore are not explicitly routed. The subcontracted depots show a similarity to the opened lockers in our problem, namely, patients assigned to them are excluded from the routing decisions. However, two main features differentiate our problem from that of Stenger et al. [21]. First, patients can only be assigned to an opened locker within the coverage distance. Second, we have to create routes visiting all opened lockers for replenishments, whereas the subcontracted depots of Stenger et al. [21] are not routed at all. The authors propose a hybrid heuristic for their problem, which is based on simulated annealing for the location decisions and variable neighborhood search for the routing decisions. The 2E-LRP (see Cuda et al. [3] for a survey), integrates the location of depots and satellites (intermediate facilities), the construction of vehicle routes from depots to satellites, and the construction of vehicle routes from the satellites to the customers.

The property of covering imposes that certain locations do not have to be visited as long as they are within a prespecified distance from another location that is visited. Covering exists in several routing problems, among these are the covering salesman problem (see Golden et al. [7] and Salari and Naji-Azimi [20] for heuristics), and the covering tour problem (see Hachicha et al. [11] and Allahyari et al. [1] for heuristics, Tricoire et al. [22] for a branch-and-cut algorithm, and Gendreau et al. [6] and Hà et al. [9] for both a heuristic and an exact algorithm). Another research on covering is that of Naji-Azimi et al. [18] in which satellite distribution centers are located using a covering tour approach. In this problem, satellite distribution centers must be located such that they cover all customers. The satellite distribution centers can be compared to the lockers in our problem. Different from our problem, all customers need to be covered in Naji-Azimi et al. [18] and consequently no routes visiting the remaining customers need to be generated. The authors provide a mathematical formulation for the problem and develop a multi-start heuristic.

Other problems combining routing and covering are the close-enough traveling salesman problem (CETSP) and the close-enough vehicle routing problem (CEVRP). In the CETSP a route needs to be designed to minimize the travel distance under the restriction that each customer is within a certain radius of the route (see Gulczynski et al. [8], Dong et al. [4], and Mennell [14] for heuristics, and Yuan et al. [23] for an exact method). The CEVRP is an extension of the CETSP that considers multiple routes (see Mennell [14] for a mathematical formulation and a heuristic). Close-enough problems also exist in the arc routing version of the problem (see Hà et al. [10], Ávila et al. [2], and Drexl [5]).

The goal of our paper is to introduce, model, and solve the simultaneous facility location and vehicle routing problem as previously described using an exact method and a heuristic approach. We mathematically model and solve this new problem arising in the health care sector in the Netherlands exactly and by means of a fast and efficient heuristic. We run several computational experiments on random instances and on instances created from data of DMS. Moreover, we also adapt instances from the classical LRP to assess the performance of our methods. We show that our hybrid heuristic is fast and finds (near-)optimal solutions.

The remainder of this paper is organized as follows. In Section 2, we formally define and formulate the problem. A hybrid heuristic is proposed in Section 3. In Section 4, extensive computational experiments are reported on several instances, and managerial insights are given. Conclusions are given in Section 5.

## 2 Problem Statement and Mathematical Formulation

The problem studied in this paper can be defined on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. Let $\mathcal{V} = \{0\} \cup \mathcal{P} \cup \mathcal{L}$, where $\{0\}$ is the depot, $\mathcal{P}$ denotes the set of patients, and $\mathcal{L}$ is the set of potential locker locations. Two sets of vehicles $\mathcal{K}$ and $\mathcal{M}$ are based at the depot. The coverage distance of a locker $j \in \mathcal{L}$ is given by $r_j$ and a fixed opening cost $F_j$ is associated with it. The cost, length and travel time of arc $(i, j) \in \mathcal{A}$ are denoted by $c_{ij}$, $d_{ij}$ and $t_{ij}$. The cost for the vehicles to the patients are penalized by a factor $\phi$ for the reasons noted in Section 1. Each patient that is within the coverage distance of an opened locker is deemed served. The remaining patients need to be visited by the vehicles from the set $\mathcal{K}$. The opened lockers need to be visited by the vehicles from the set $\mathcal{M}$. A maximum route duration $T_1$ ($T_2$) is given for the vehicles of the set $\mathcal{K}$ ($\mathcal{M}$). The route duration of a vehicle consists of the total of travel time and service time. The service time associated with node $i \in \mathcal{V}$ is given by $s_i$, where $s_i > 0$ for $i \in \mathcal{P} \cup \mathcal{L}$ and $s_0 = 0$. A vehicle route is feasible if it starts and ends at the depot and does not exceed the maximum route duration constraint.

Three types of decisions must be made, namely, 1) whether locker $j \in \mathcal{L}$ will be opened, 2) the construction of the patient routes, i.e., the vehicle routes that visit the patient nodes that are not assigned to a locker, and 3) the construction of the locker routes, i.e., the vehicle routes that visit the opened lockers. The objective of the problem is to minimize the opening costs of the lockers and routing costs.

The problem can be modeled as follows. Let binary variables $w_i$ be equal to one if and only if node $i \in \mathcal{P}$ is assigned to the depot, i.e., the patient needs to be visited by a vehicle, $z_i$ equal to one if and only if node $i \in \mathcal{P}$ is assigned to a locker, $u_{ij}$ equal to one if and only if patient $i \in \mathcal{P}$ is assigned to locker $j \in \mathcal{L}$, and $v_i$ equal to one if and only if locker $i \in \mathcal{L}$ is opened. For the patient routes, we use binary variables $x_{ij}^k$ equal to one if and only if vehicle $k \in \mathcal{K}$ travels from node $i$ to $j$, $i, j \in \mathcal{P} \cup \{0\}$, $y_i^k$ equal to one if and only if vehicle $k \in \mathcal{K}$ visits node $i \in \mathcal{P} \cup \{0\}$, and integer variables $h_{ij}^k$ reflecting the load on vehicle $k \in \mathcal{K}$ while traversing arc $(i, j), i, j \in \mathcal{P} \cup \{0\}$. Similar variables are used for the locker routes, namely binary variables $p_{ij}^k$ equal to one if and only if vehicle $k \in \mathcal{M}$ travels from node $i$ to $j$, $i, j \in \mathcal{L} \cup \{0\}$, $q_i^k$ equal to one if and only if vehicle $k \in \mathcal{M}$ visits node $i \in \mathcal{L} \cup \{0\}$, and integer variables $g_{ij}^k$ which reflect the load on vehicle $k \in \mathcal{M}$ while traversing arc $(i, j), i, j \in \mathcal{L} \cup \{0\}$. The problem can be formulated as follows:

$$\min \sum_{i \in \mathcal{P} \cup \{0\}} \sum_{j \in \mathcal{P} \cup \{0\}} \sum_{k \in \mathcal{K}} \phi c_{ij} x_{ij}^k + \sum_{i \in \mathcal{L} \cup \{0\}} \sum_{j \in \mathcal{L} \cup \{0\}} \sum_{k \in \mathcal{M}} c_{ij} p_{ij}^k + \sum_{i \in \mathcal{L}} F_i v_i \tag{1}$$

subject to

$$\sum_{j \in \mathcal{L}} u_{ij} = z_i \qquad i \in \mathcal{P} \tag{2}$$

$$w_i + z_i = 1 \qquad i \in \mathcal{P} \tag{3}$$

$$u_{ij} \leq v_j \qquad i \in \mathcal{P},\ j \in \mathcal{L} \tag{4}$$

$$u_{ij} d_{ij} \leq r_j v_j \qquad i \in \mathcal{P},\ j \in \mathcal{L} \tag{5}$$

$$r_j v_j \leq d_{ij} + z_i M \qquad i \in \mathcal{P},\ j \in \mathcal{L} \tag{6}$$

$$\sum_{j \in \mathcal{P} \cup \{0\}} x_{ij}^k + \sum_{j \in \mathcal{P} \cup \{0\}} x_{ji}^k = 2 y_i^k \qquad i \in \mathcal{P},\ k \in \mathcal{K} \tag{7}$$

$$\sum_{j \in \mathcal{P}} x_{0j}^k + \sum_{j \in \mathcal{P}} x_{j0}^k = 2 y_0^k \qquad k \in \mathcal{K} \tag{8}$$

$$\sum_{k \in \mathcal{K}} y_i^k = w_i \qquad i \in \mathcal{P} \tag{9}$$

$$\sum_{j\in\mathcal{P}\cup\{0\}} x_{ij}^k = \sum_{j\in\mathcal{P}\cup\{0\}} x_{ji}^k \qquad i\in\mathcal{P},\ k\in\mathcal{K} \tag{10}$$

$$\sum_{j\in\mathcal{P}}\sum_{k\in\mathcal{K}} h_{0j}^k = \sum_{i\in\mathcal{P}} w_i \tag{11}$$

$$\sum_{i\in\mathcal{P}\cup\{0\}} h_{ij}^k - \sum_{i\in\mathcal{P}\cup\{0\}} h_{ji}^k = y_j^k \qquad j\in\mathcal{P},\ k\in\mathcal{K} \tag{12}$$

$$h_{ij}^k \le (|P|-1)x_{ij}^k \qquad i\in P,\ j\in P\cup\{0\},\ k\in K \tag{13}$$

$$h_{0j}^k \le |P|x_{0j}^k \qquad j\in P\cup\{0\},\ k\in K \tag{14}$$

$$\sum_{i\in\mathcal{P}\cup\{0\}}\sum_{j\in\mathcal{P}\cup\{0\}} t_{ij}x_{ij}^k + \sum_{i\in\mathcal{P}\cup\{0\}}\sum_{j\in\mathcal{P}\cup\{0\}} s_j x_{ij}^k \le T_1 \qquad k\in\mathcal{K} \tag{15}$$

$$\sum_{j\in\mathcal{L}\cup\{0\}} p_{ij}^k + \sum_{j\in\mathcal{L}\cup\{0\}} p_{ji}^k = 2q_i^k \qquad i\in\mathcal{L},\ k\in\mathcal{M} \tag{16}$$

$$\sum_{j\in\mathcal{L}} p_{0j}^k + \sum_{j\in\mathcal{L}} p_{j0}^k = 2q_0^k \qquad k\in\mathcal{M} \tag{17}$$

$$\sum_{k\in\mathcal{M}} q_i^k = v_i \qquad i\in\mathcal{L} \tag{18}$$

$$\sum_{j\in\mathcal{L}\cup\{0\}} p_{ij}^k = \sum_{j\in\mathcal{L}\cup\{0\}} p_{ji}^k \qquad i\in\mathcal{L},\ k\in\mathcal{M} \tag{19}$$

$$\sum_{j\in\mathcal{L}}\sum_{k\in\mathcal{M}} g_{0j}^k = \sum_{i\in\mathcal{L}} v_i \tag{20}$$

$$\sum_{i\in\mathcal{L}\cup\{0\}} g_{ij}^k - \sum_{i\in\mathcal{L}\cup\{0\}} g_{ji}^k = q_j^k \qquad j\in\mathcal{L},\ k\in\mathcal{M} \tag{21}$$

$$g_{ij}^k \le (|L|-1)p_{ij}^k \qquad i\in L,\ j\in L\cup\{0\},\ k\in M \tag{22}$$

$$g_{0j}^k \le |L|p_{0j}^k \qquad j\in L\cup\{0\},\ k\in M \tag{23}$$

$$\sum_{i\in\mathcal{L}\cup\{0\}}\sum_{j\in\mathcal{L}\cup\{0\}} t_{ij}p_{ij}^k + \sum_{i\in\mathcal{L}\cup\{0\}}\sum_{j\in\mathcal{L}\cup\{0\}} s_j p_{ij}^k \le T_2 \qquad k\in\mathcal{M} \tag{24}$$

$$z_i,\ w_i \in \{0,1\} \qquad i\in\mathcal{P} \tag{25}$$

$$u_{ij} \in \{0,1\} \qquad i\in\mathcal{P},\ j\in\mathcal{L} \tag{26}$$

$$v_i \in \{0,1\} \qquad i\in\mathcal{L} \tag{27}$$

$$x_{ij}^k \in \{0,1\} \qquad i,j\in\mathcal{P}\cup\{0\},\ k\in\mathcal{K} \tag{28}$$

$$y_i^k \in \{0,1\} \qquad i\in\mathcal{P}\cup\{0\},\ k\in\mathcal{K} \tag{29}$$

$$h_{ij}^k \geq 0 \qquad i, j \in \mathcal{P} \cup \{0\},\ k \in \mathcal{K} \tag{30}$$

$$p_{ij}^k \in \{0, 1\} \qquad i, j \in \mathcal{L} \cup \{0\},\ k \in \mathcal{M} \tag{31}$$

$$q_i^k \in \{0, 1\} \qquad i \in \mathcal{L} \cup \{0\},\ k \in \mathcal{M} \tag{32}$$

$$g_{ij}^k \geq 0 \qquad i, j \in \mathcal{L} \cup \{0\},\ k \in \mathcal{M}. \tag{33}$$

The objective function (1) minimizes the total of the travel costs for the patient and the locker routes, and the costs for the opened lockers. Constraints (2)–(6) correspond to the opening of the lockers and the assignment of patients to the lockers or the depot. Constraints (2) ensure that if a patient is assigned to a locker, it is assigned to only one locker. Constraints (3) ensure that a patient is either assigned to the depot or to one of the lockers. Due to constraints (4) a patient can only be assigned to a locker if it is opened. Constraints (5) ensure that a patient can only be assigned to a locker within its coverage distance. Due to constraints (6), a patient is forced to be assigned to a locker if it is within the coverage distance of at least one opened locker. Constraints (7)–(15) correspond to the patient routes. Constraints (7) ensure that if a vehicle visits a patient, it enters and leaves its node exactly once, whereas, if the vehicle does not visit a patient, its node is not visited. Constraints (8) ensure that only vehicles that are used can visit patients. Due to constraints (9), if a patient is assigned to the depot then exactly one vehicle visits its node, otherwise no vehicle visits it. Constraints (10) are the flow conservation constraints. Constraints (11)–(14) ensure that the solution is connected. More specifically, constraint (11) ensures that the total load leaving the depot is equal to the number of patients assigned to it. Constraints (12) ensure that if a patient is assigned to a given vehicle, the difference between the load of the vehicle arriving at and leaving from its node is exactly one. Constraints (15) ensure that the route duration is respected. Constraints (16)–(24) are similar to (7)–(15) and apply for the locker routes. Constraints (25)–(33) define the nature and the range of the variables. Parameter $M$ in constraints (6) can be set as small as possible as:

$$M = \begin{cases} \max_{j \in \mathcal{L}} r_j & \text{if } \exists i \in \mathcal{P},\ j \in \mathcal{L} \text{ s.t. } d_{ij} < r_j, \\ \max_{j \in \mathcal{L}} r_j - \min_{i \in \mathcal{P}, j \in \mathcal{L} | d_{ij} < r_j} d_{ij} & \text{otherwise.} \end{cases}$$

The following valid inequalities can be imposed to strengthen the formulation:

$$h_{i0}^k = 0 \qquad i \in \mathcal{P},\ k \in \mathcal{K} \tag{34}$$

$$h_{ii}^k = 0 \qquad i \in \mathcal{P} \cup \{0\},\ k \in \mathcal{K} \tag{35}$$

$$\sum_{k \in \mathcal{K}} x_{i0}^k \leq \sum_{k \in \mathcal{K}} y_i^k \qquad i \in \mathcal{P} \tag{36}$$

$$\sum_{k \in \mathcal{K}} x_{0i}^k \leq \sum_{k \in \mathcal{K}} y_i^k \qquad i \in \mathcal{P} \tag{37}$$

$$x_{ij}^k + x_{ji}^k \leq 1 \qquad i, j \in \mathcal{P},\ k \in \mathcal{K} \tag{38}$$

$$g_{i0}^k = 0 \qquad i \in \mathcal{L},\ k \in \mathcal{M} \tag{39}$$

$$g_{ii}^k = 0 \qquad i \in \mathcal{L} \cup \{0\},\ k \in \mathcal{M} \tag{40}$$

$$\sum_{k \in \mathcal{M}} p_{i0}^k \leq \sum_{k \in \mathcal{M}} q_i^k \qquad i \in \mathcal{L} \tag{41}$$

$$\sum_{k \in \mathcal{M}} p_{0i}^k \leq \sum_{k \in \mathcal{M}} q_i^k \qquad i \in \mathcal{L} \tag{42}$$

$$p_{ij}^k + p_{ji}^k \leq 1 \qquad i, j \in \mathcal{L},\ k \in \mathcal{M}. \tag{43}$$

Constraints (34)–(38) apply for the patient routes. Constraints (34) enforce the load on a vehicle to be zero if it enters the depot, while constraints (35) enforce the load on a vehicle to equal zero for each arc $(i, i)$, $i \in P \cup \{0\}$. Constraints (36) and (37) impose that if a patient is not visited by a vehicle, the arc from the patient towards the depot and the arc from the depot towards the patient, respectively, are not traversed by a vehicle. Constraints (38) exclude an arc between two patients to be traversed by a vehicle if its opposite arc is traversed by that vehicle. Constraints (39)–(43) are similar to (34)–(38) and apply for the locker routes.

# 3 Hybrid Heuristic

In this section, we describe the hybrid heuristic we propose to simultaneously solve the facility location and vehicle routing problem as formalized in Section 2. The general idea behind this heuristic is that it iteratively modifies a solution by changing the set of opened lockers, updates the routes accordingly and applies a variable neighborhood search (VNS) procedure to improve the routes. The structure of the heuristic is given in Algorithm 1. The heuristic starts with an initial solution (line 1). While the stopping criterion is not met, the following steps are executed in each iteration. The set of opened lockers is changed (line 4). Then, the patient and locker routes are updated accordingly (line 5). A VNS procedure is applied to the patient and locker routes (line 6). The acceptance of the solution is based on simulated annealing (line 7). When the stopping criterion is met, the heuristic outputs the best found solution.

1 Construct initial solution $s$;<br>
2 **while** *stopping criterion is not met* **do**<br>
3     $s' \leftarrow s$;<br>
4     Change the set of opened lockers in $s'$;<br>
5     Update the routes in $s'$;<br>
6     Apply the VNS Procedure to the routes in $s'$;<br>
7     **if** *Accept*$(s', s)$ **then**<br>
8         $s \leftarrow s'$;<br>
9     **end**<br>
10 **end**<br>
**Result**: $s_{BEST}$;

**Algorithm 1:** Structure of the Hybrid Heuristic

In Section 3.1 we describe the construction of the initial solution. Section 3.2 details the step in which the set of opened lockers is changed. The phase in which the routes are updated is described in Section 3.3. The VNS procedure is presented in Section 3.4. The acceptance and stopping criteria are given in Section 3.5. Section 3.6 discusses feasibility issues.

## 3.1 Initial Solution

The initial solution is constructed as follows. It starts with $\mathcal{M}$ empty locker routes and $\mathcal{K}$ empty patient routes. Each locker $i \in \mathcal{L}$ is opened with probability $\rho$, $0 < \rho < 1$. The opened lockers are in a random sequence iteratively inserted. Each locker is inserted at its best position in one of the locker routes, i.e., in the position that yields the smallest increase in objective value. Let $\mathcal{P}$ be the set of patients that are not within the covering range of any of the opened lockers. Each patient $p \in \mathcal{P}$ is inserted in one of the patient routes at its best position, where the sequence of patients is randomly determined. The VNS procedure (described in Section 3.4) is applied to the locker and patient routes.

## 3.2 Changing the Set of Opened Lockers

In this step the set of opened lockers is changed based on one of the following three procedures: 1) opening a locker, 2) closing a locker, or 3) opening and closing a locker. The three different procedures will be explained in detail in the next paragraphs. The idea of opening, closing, and opening and closing a locker is derived from Nagy and Salhi [16]. The first procedure is applied if all lockers in the current solution are closed, whereas the second procedure is applied if all lockers in the current solution are opened. If the former two situations do not apply, either procedure 1, 2, or 3 is invoked with a probability equal to $\delta_1$, $\delta_2$, and $\delta_3$, respectively, where $0 < \delta_1, \delta_2, \delta_3 < 1$ and $\delta_1 + \delta_2 + \delta_3 = 1$.

### 3.2.1 Opening a Locker

In this procedure one locker in the set $\mathcal{C}$ of closed lockers is opened. Opening a locker is based on one of the following three methods: 1) random, 2) based on cost reduction, or 3) based on the number of covered patients. The probability of using methods 1, 2, and 3 is $\eta_1$, $\eta_2$, and $\eta_3$, respectively, where $0 < \eta_1, \eta_2, \eta_3 < 1$ and $\eta_1 + \eta_2 + \eta_3 = 1$. Method 1 randomly selects one locker from the set $\mathcal{C}$. Method 2 selects a locker with a bias towards lockers with high cost reductions as follows. The cost reduction of opening currently closed locker $c \in \mathcal{C}$ is calculated as $r_c = R_s - R_c - F_c$, where $R_c$ is the total routing costs with locker $c$ inserted at its best position on a locker route and removing from the patient routes those that are within its coverage range; $R_s$ is the routing costs of the current solution; and $F_c$ is the opening cost of locker $c$. Let $L$ be the array of closed lockers sorted by non-increasing cost reduction. Locker $c = L[\lceil y^{p_1}|\mathcal{C}|\rceil]$ is selected to open, where $p_1$ is a user-defined parameter and $y$ is randomly drawn from the interval $[0, 1)$. Method 3 computes, for each closed locker, the number of patients it covers that are not covered by any of the remaining opened lockers. Then, the locker is selected similarly to Method 2, where $L$ is now the array of closed lockers sorted by non-increasing number of affected patients. The method uses parameter $p_2$ instead of $p_1$.

### 3.2.2 Closing a Locker

In this procedure, one locker in the set $\mathcal{O}$ of opened lockers is closed. Closing a locker is based on one of the following three methods: 1) random, 2) based on cost reduction, or 3) based on the number of covered patients. These methods are similar to opening a locker, where in Method 2 the cost reduction of closing a currently opened locker $o \in \mathcal{O}$ is calculated as $r_o = R_s - R_o + F_o$, where $R_o$ is the routing costs when removing locker $o$ and inserting the patients that are within the coverage range of locker $o$ and not within the coverage range of any of the remaining opened lockers, where the patients are iteratively inserted at their best position, based on a random sequence. In Method 3 the array of closed lockers is sorted by non-decreasing number of patients only covered by the closed locker. The probability of using methods 1, 2, and 3 is $\gamma_1$, $\gamma_2$, and $\gamma_3$, respectively, where $0 < \gamma_1, \gamma_2, \gamma_3 < 1$ and $\gamma_1 + \gamma_2 + \gamma_3 = 1$.

### 3.2.3 Opening and Closing a Locker

Let $\mathcal{O}$ and $\mathcal{C}$ be the set of opened and closed lockers, respectively. With a probability equal to $\theta$, a locker $o \in \mathcal{O}$ to close is selected first, after which a locker $c \in \mathcal{C}$ to open is randomly selected with equal probability such that locker $o$ is within the coverage radius of locker $c$. With probability $1 - \theta$, the procedure first selects a locker to open and then selects a locker to close.

## 3.3 Updating the Routes

After the set of opened lockers is changed, the routes are updated accordingly. If a locker is closed in the current iteration it is removed from its route and the patients that are within its coverage radius and not within the coverage radius of any of the remaining opened lockers are inserted in the patient routes at their best positions, which is done iteratively in a random sequence. If a locker is opened in the current iteration it is inserted at its best position in one of the locker routes, and the patients within its coverage radius are removed from their routes. If a locker is opened and another one is closed in the current iteration, all removal operations occur before the insertion operations begin.

## 3.4 VNS Procedure

The basic idea of VNS is proposed by Mladenović and Hansen [15]. In the VNS procedure different local search operators are applied to the vehicle routes. The structure of our VNS procedure is given in Algorithm 2. The procedure starts with a set of routes, i.e., all locker routes or all patient routes. The VNS procedure applies a local search operator until no improvement is found and then continues with the next operator. If all operators are applied and an improvement is found the procedure restarts, otherwise it is stopped. The six local search operators are used in the following order: *1-0-Exchange*, *1-1-Exchange*, *Intra-2-Opt*, *Inter-2-Opt*, *Intra-3-Opt*, *Inter-3-Opt'*. In Algorithm 2, these operators are referred to as $Operator_k$, $k \in \{1, \ldots, 6\}$. The *1-0-Exchange*, *1-1-Exchange* and *k-Opt* operators are commonly applied in the literature (see, e.g., Zachariadis and Kiranoudis [24], and Kindervater and Savelsbergh [12]).

```
R: Set of routes;
improveoverall ← 1;
while improveoverall = 1 do
    improveoverall ← 0 ;
    for k ∈ {1, ..., 6} do
        improve ← 1;
        while improve = 1 do
            improve ← 0; R′ ← R;
            R′ ← Operator_k(R′);
            if cost(R) > cost(R′) then
                improve ← 1; improveoverall ← 1; R ← R′;
            end
        end
    end
end
```

**Algorithm 2:** Structure of the VNS procedure

The *1-0-Exchange* operator iteratively removes a location and reinserts it at its best position, where the sequence of locations chosen at each iteration is randomly determined.

For the *1-1-Exchange* operator, the sequence in which the locations are examined is random. The position of a current location $i$ is interchanged with the position of the location

$j$ that yields the largest decrease in objective value (if any).

The *Intra-2-Opt* operator is applied to each route separately. For each combination of two arcs in the same route, the operator computes the decrease in objective value when removing the arcs and reconnecting the resulting paths. If there is at least one combination of arcs that results in a decrease in objective value, the two arcs that yield the largest decrease in objective value are removed and the resulting paths are reconnected.

The *Inter-2-Opt* operator is similar to the *Intra-2-Opt* operator, but here the two arcs come from different routes.

The *Intra-3-Opt* operator computes the decrease in objective value when removing three arcs from the same route and reconnecting the resulting paths in a different way such that the largest decrease in objective value is obtained. If there is at least one combination of arcs that results in a decrease in objective value, the three arcs that yield the largest decrease in objective value are removed and the resulting paths are reconnected.

The *Inter-3-Opt'* operator is derived from the *Inter-3-Opt* operator and computes the decrease in objective value when removing a partial path and inserting it in the best position of another route, where the size of the partial path is larger than one. The partial path that leads to the largest decrease in objective value (if any) is inserted at the best position of another route.

## 3.5 Acceptance and Stopping Criteria

In each iteration, a new solution $s'$ is accepted if its objective value $f(s')$ is smaller than the objective value $f(s)$ of the current solution $s$. If $f(s') \geq f(s)$, the acceptance of the new solution $s'$ is based on a simulated annealing acceptance criterion (see Kirkpatrick et al. [13]), where the probability of acceptance is equal to $e^{-(f(s')-f(s))/T}$, where $T > 0$ is the temperature at the given iteration. At the first iteration, the temperature is set to $T = T_{start}$. In the subsequent iterations, the temperature decreases according to the formula $T = \kappa \cdot T$, where $0 < \kappa < 1$ is the cooling rate. The algorithm stops when it has run for $N_{iter}$ iterations.

### 3.6 Feasibility

We allow infeasible solutions that violate the maximum route duration constraints to be accepted during the search of our heuristic. In order to steer the solution towards feasibility, a penalty is incurred and added to the objective value if one of the maximum route duration constraints is violated. For each route $R$, the penalty $p(R)$ is computed as $p(R) = p_{time} \cdot \max\{0, T_R - \overline{T}_R\}$, where $p_{time}$ is a constant penalty for each time unit exceeding the maximum route duration, $T_R$ is the total time of route $R$ and $\overline{T}_R$ is the maximum route duration of the vehicle corresponding to route $R$. At the end of the heuristic, the solution is checked for feasibility.

## 4 Computational Experiments

The mathematical model in Section 2 was solved using CPLEX 12.6.1, and the hybrid heuristic was coded in Java. All computations were executed on machines equipped with two Intel Westmere EP X5650 six-core processors running at 2.667 GHz, and with up to 48 GB of RAM installed per node running the Scientific Linux 6.3. In Section 4.1 we describe the parameter setting of the hybrid heuristic. In Section 4.2 we describe the two data sets used to test the solution methods. Computational results of the branch-and-bound algorithm applied to the integer formulation given in Section 2 and results of the hybrid heuristic are discussed in detail in Section 4.3.

### 4.1 Parameter Setting

We created a set of 240 random instances to tune the parameters of the hybrid heuristic. The instances in this set contain up to 100 patient nodes and 50 potential locker locations.

We started the parameter tuning by setting the parameters to values that seemed reasonable based on our experience. We iteratively tested a range of values for each parameter and updated the value of the parameter if it led to an improvement. We continued until we could not find a significant improvement for any of the parameters, evaluating the trade-off between solution quality and computational time. The final values for the parameters in the heuristic are as follows. The number of iterations $N_{iter}$ is equal to 50000, the start temperature $T_{start}$ is such that a solution 5% worse than the initial solution is accepted in the first iteration with probability 0.2. The cooling rate $\kappa$ is set to 0.999875716 which implies that the temperature at the last iteration is equal to $0.002 \cdot T_{start}$. The probability of opening a locker in the initial solution is $\rho = 0.1$. The probability $\delta_1$ of opening a locker, $\delta_2$ of closing a locker, and $\delta_3$ of opening and closing a locker are set to 0.45, 0.5 and 0.05, respectively. The probabilities for the different methods for opening or closing a locker are given by $(\eta_1, \eta_2, \eta_3, \gamma_1, \gamma_2, \gamma_3) = (0.01, 0.94, 0.05, 0.01, 0.94, 0.05)$. The performance of the heuristic strongly depends on the choice of the values for the parameters in the set $(\delta_2, \delta_2, \delta_3, \eta_1, \eta_2, \eta_3, \gamma_1, \gamma_2, \gamma_3)$. The parameters $p_1$ and $p_2$ within the methods for opening or closing a locker are set to 10 and 3, respectively. The parameter $\theta$ in the procedure of opening and closing a locker is set to 0.7. The parameter $p_{time}$ is instance-dependent and equal to $50 \cdot (x_{max} - x_{min} + y_{max} - y_{min})$, where $x_{max}$ and $x_{min}$ are the values of the largest and smallest x-coordinate, respectively, and $y_{max}$ and $y_{min}$ are the values of the largest and smallest y-coordinate, respectively.

## 4.2 Description of the Data Sets

We created three sets of instances to test the branch-and-bound and hybrid heuristic algorithms. The first set of instances is randomly generated, the second set of instances is based on data from DMS, and the third set of instances are modified from a set of benchmarks for the LRP (see Appendix).

The first set of instances is randomly generated and consists of 117 instances. The number of potential locker locations is up to 50 and the number of patients is up to 150. For each instance the number of vehicles in the sets $\mathcal{K}$ and $\mathcal{M}$ are equal to 3 and 2, respectively. The feasibility and difficulty heavily depends on the choice of the parameters. We have chosen the parameters such that the instances are feasible and that the instances rightly reflect the structure of the problem faced by our industrial partner. The x-coordinates and y-coordinates of the patients and lockers are randomly generated in the range $[0, 100]$, whereas those of the depot are random in the range $[25, 75]$. The distances are computed based on the Euclidean measure and rounded to the nearest integer. We set the travel time and the travel cost equal to the travel distance. The penalty factor $\phi$ is set to 10. The service time for the patients is a random integer in the set $\{1, \ldots, 5\}$, while the service time for a locker is five times the service time for a patient. Let $N_p$ be the nearest-neighbor heuristic cost for visiting all patients starting from and ending at the depot (without considering the maximum route duration), and let $N_l$ be the corresponding cost for visiting all locker locations. Then, we set $T_1$, the maximum route duration of the vehicles in the set $\mathcal{K}$, equal to the nearest integer of $2/3 \cdot N_p + s_p \cdot |\mathcal{P}|$, where $s_p$ denotes the service time for the patients, and similarly, we set $T_2$, the maximum route duration of the vehicles in the set $\mathcal{M}$, equal to the nearest integer of $2/3 \cdot N_l + s_l \cdot |\mathcal{L}|$, where $s_l$ denotes the service time for the lockers. The opening cost of a locker $F_i$ differs between the lockers and is equal to the nearest integer of $\pi \cdot N_p/|\mathcal{L}|$, where $\pi$ is a random value in the range $[1, 4]$. The coverage distance $r_j$ is the same for all potential lockers and is a random integer in the set $\{10, 11, \ldots, 20\}$.

The second set of instances is derived from location data of DMS. The original data set consists of one depot location (a pharmacy) and the locations of the patients that this pharmacy has served over a period of time. We created a set of 117 instances, which we refer to as the DMS instances. For these instances, we focused on an area of $20000 \times 20000$ meters, with the depot located in the center. Within this area, we randomly selected a set of patients from the original data set, not selecting the ones within the coverage distance of the depot since these patients are naturally served by the pharmacy. We created instances with up to 150 patients and 50 potential locker locations. The lockers are randomly located in the aforementioned area, where we ensured that at least one patient is within the coverage distance of each potential locker location. For each instance, the number of vehicles in the sets $\mathcal{K}$ and $\mathcal{M}$ are equal to 3 and 2, respectively, and the maximum route duration for each vehicle is equal to 4 hours. The service time of a patient equals 180 seconds and the service time of a locker equals 360 seconds. The travel cost is equal to the travel distance in meters and the travel time is computed based on an average speed of 50 km/h. The penalty factor $\phi$ is set to 10. The opening cost of a locker is set to 20000 and the coverage distance of a locker is set to 1000 meters.

## 4.3 Comparative Results

The branch-and-bound algorithm applied to the integer program given in Section 2 was run with a time limit of 7200 seconds. The hybrid heuristic was run ten times for each instance. In Tables 1 and 2, the computational results are given for the instances containing up to 100 patients in the set of random instances and the set of DMS instances, respectively. The tables contain for each instance $|\mathcal{P}|$, the number of patients and $|\mathcal{L}|$, the number of potential locker locations. For the branch-and-bound algorithm we report for each instance $\overline{z}$, the value of the best integer solution found, $\underline{z}$ the best lower bound, and *Gap* (%), the percentage gap between $\overline{z}$ and $\underline{z}$, computed as $(\overline{z} - \underline{z})/\underline{z} \cdot 100$, and $Time$ $(s)$, the total computation time in seconds. For the hybrid heuristic, we report for each instance *Minimum*, the minimum objective value over ten runs, *Gap (%)* the percentage gap between *Minimum* and $\overline{z}$, computed as $(Minimum - \overline{z})/\overline{z} \cdot 100$, *Average*, the average objective value over ten runs, *Gap (%)* the percentage gap between *Average* and $\overline{z}$, computed as $(Average - \overline{z})/\overline{z} \cdot 100$, and $Time$ $(s)$, the average running time (in seconds) over the 10 runs.

**Table 1:** Computational Results for the Random Instances

| Instance | $\|\mathcal{P}\|$ | $\|\mathcal{L}\|$ | Branch-and-Bound $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Hybrid heuristic Minimum | Gap (%) | Average | Gap (%) | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R1 | 30 | 10 | 3,884 | 3,884.0 | 0.00 | 44.5 | 3,884 | 0.00 | 3,884.0 | 0.00 | 10.4 |
| R2 | 30 | 15 | 2,897 | 2,897.0 | 0.00 | 59.8 | 2,897 | 0.00 | 2,897.0 | 0.00 | 11.1 |
| R3 | 30 | 20 | 2,050 | 2,050.0 | 0.00 | 12.6 | 2,050 | 0.00 | 2,050.0 | 0.00 | 11.3 |
| R4 | 30 | 25 | 3,496 | 3,496.0 | 0.00 | 100.4 | 3,496 | 0.00 | 3,496.0 | 0.00 | 12.0 |
| R5 | 30 | 30 | 3,841 | 3,841.0 | 0.00 | 189.8 | 3,841 | 0.00 | 3,841.0 | 0.00 | 15.0 |
| R6 | 30 | 35 | 2,767 | 2,767.0 | 0.00 | 248.7 | 2,767 | 0.00 | 2,767.0 | 0.00 | 17.0 |
| R7 | 30 | 40 | 2,129 | 2,129.0 | 0.00 | 471.3 | 2,129 | 0.00 | 2,129.0 | 0.00 | 15.9 |
| R8 | 30 | 45 | 1,012 | 1,012.0 | 0.00 | 162.3 | 1,012 | 0.00 | 1,012.4 | 0.04 | 17.2 |
| R9 | 30 | 50 | 455 | 455.0 | 0.00 | 251.1 | 455 | 0.00 | 455.0 | 0.00 | 18.2 |
| R10 | 40 | 10 | 4,388 | 4,388.0 | 0.00 | 70.5 | 4,388 | 0.00 | 4,388.0 | 0.00 | 19.7 |
| R11 | 40 | 15 | 3,874 | 3,874.0 | 0.00 | 206.7 | 3,874 | 0.00 | 3,874.0 | 0.00 | 19.4 |
| R12 | 40 | 20 | 3,531 | 3,531.0 | 0.00 | 102.3 | 3,531 | 0.00 | 3,531.0 | 0.00 | 16.7 |
| R13 | 40 | 25 | 4,495 | 4,495.0 | 0.00 | 117.8 | 4,495 | 0.00 | 4,495.0 | 0.00 | 21.5 |
| R14 | 40 | 30 | 4,399 | 4,399.0 | 0.00 | 302.1 | 4,399 | 0.00 | 4,399.0 | 0.00 | 25.7 |
| R15 | 40 | 35 | 3,964 | 3,964.0 | 0.00 | 337.4 | 3,964 | 0.00 | 3,964.0 | 0.00 | 27.0 |
| R16 | 40 | 40 | 2,789 | 2,789.0 | 0.00 | 466.5 | 2,789 | 0.00 | 2,789.0 | 0.00 | 27.5 |
| R17 | 40 | 45 | 2,641 | 2,641.0 | 0.00 | 498.9 | 2,641 | 0.00 | 2,641.0 | 0.00 | 24.9 |
| R18 | 40 | 50 | 3,390 | 3,390.0 | 0.00 | 649.9 | 3,390 | 0.00 | 3,390.0 | 0.00 | 29.8 |
| R19 | 50 | 10 | 5,259 | 5,259.0 | 0.00 | 386.9 | 5,259 | 0.00 | 5,259.0 | 0.00 | 31.2 |
| R20 | 50 | 15 | 4,768 | 4,768.0 | 0.00 | 265.0 | 4,768 | 0.00 | 4,768.0 | 0.00 | 34.0 |
| R21 | 50 | 20 | 5,058 | 5,058.0 | 0.00 | 549.3 | 5,058 | 0.00 | 5,058.0 | 0.00 | 32.3 |
| R22 | 50 | 25 | 2,351 | 2,351.0 | 0.00 | 25.9 | 2,351 | 0.00 | 2,351.0 | 0.00 | 30.5 |
| R23 | 50 | 30 | 4,040 | 4,040.0 | 0.00 | 582.7 | 4,040 | 0.00 | 4,040.0 | 0.00 | 38.2 |
| R24 | 50 | 35 | 3,959 | 3,959.0 | 0.00 | 566.2 | 3,959 | 0.00 | 3,959.0 | 0.00 | 38.3 |
| R25 | 50 | 40 | 3,697 | 3,697.0 | 0.00 | 1,045.5 | 3,697 | 0.00 | 3,697.0 | 0.00 | 35.7 |
| R26 | 50 | 45 | 4,120 | 4,120.0 | 0.00 | 821.9 | 4,120 | 0.00 | 4,120.0 | 0.00 | 42.7 |
| R27 | 50 | 50 | 2,715 | 2,715.0 | 0.00 | 4,286.4 | 2,715 | 0.00 | 2,715.2 | 0.01 | 38.1 |
| R28 | 60 | 10 | 6,170 | 6,170.0 | 0.00 | 2,022.6 | 6,245 | 1.22 | 6,314.5 | 2.34 | 58.9 |
| R29 | 60 | 15 | 4,289 | 4,289.0 | 0.00 | 891.9 | 4,289 | 0.00 | 4,289.0 | 0.00 | 42.9 |
| R30 | 60 | 20 | 5,245 | 5,245.0 | 0.00 | 1,227.9 | 5,245 | 0.00 | 5,245.0 | 0.00 | 54.1 |
| R31 | 60 | 25 | 2,540 | 2,540.0 | 0.00 | 653.2 | 2,540 | 0.00 | 2,540.0 | 0.00 | 41.0 |
| R32 | 60 | 30 | 3,854 | 3,854.0 | 0.00 | 1,969.3 | 3,854 | 0.00 | 3,854.0 | 0.00 | 49.0 |
| R33 | 60 | 35 | 2,960 | 2,960.0 | 0.00 | 1,226.3 | 2,960 | 0.00 | 2,960.0 | 0.00 | 45.4 |
| R34 | 60 | 40 | 4,830 | 4,830.0 | 0.00 | 1,839.2 | 4,830 | 0.00 | 4,830.0 | 0.00 | 71.5 |
| R35 | 60 | 45 | 3,580 | 3,290.3 | 8.80 | 7,200.0 | **3,562** | **–0.50** | **3,562.0** | **–0.50** | 55.4 |
| R36 | 60 | 50 | 2,677 | 2,677.0 | 0.00 | 5,198.4 | 2,677 | 0.00 | 2,677.0 | 0.00 | 48.1 |
| R37 | 70 | 10 | 5,605 | 5,605.0 | 0.00 | 3,144.9 | 5,605 | 0.00 | 5,605.0 | 0.00 | 81.8 |
| R38 | 70 | 15 | 5,157 | 5,157.0 | 0.00 | 4,954.3 | 5,157 | 0.00 | 5,157.0 | 0.00 | 73.4 |
| R39 | 70 | 20 | 5,447 | 5,447.0 | 0.00 | 2,466.0 | 5,447 | 0.00 | 5,447.0 | 0.00 | 74.9 |
| R40 | 70 | 25 | 4,738 | 4,738.0 | 0.00 | 5,254.2 | 4,738 | 0.00 | 4,738.0 | 0.00 | 56.6 |
| R41 | 70 | 30 | 5,083 | 5,083.0 | 0.00 | 2,159.2 | 5,083 | 0.00 | 5,083.0 | 0.00 | 73.6 |
| R42 | 70 | 35 | 1,732 | 1,732.0 | 0.00 | 36.9 | 1,732 | 0.00 | 1,732.0 | 0.00 | 63.3 |
| R43 | 70 | 40 | 5,250 | 3,170.0 | 65.61 | 7,200.0 | **4,053** | **–22.80** | **4,053.0** | **–22.80** | 74.6 |
| R44 | 70 | 45 | 3,767 | 1,940.4 | 94.14 | 7,200.0 | **3,365** | **–10.67** | **3,365.0** | **–10.67** | 73.2 |
| R45 | 70 | 50 | 2,298 | 2,264.9 | 1.46 | 7,200.0 | **2,286** | **–0.52** | 2,287.0 | –0.48 | 76.1 |
| R46 | 80 | 10 | 6,244 | 6,244.0 | 0.00 | 4,068.2 | 6,244 | 0.00 | 6,244.0 | 0.00 | 118.2 |
| R47 | 80 | 15 | 5,472 | 4,846.1 | 12.92 | 7,200.0 | **5,162** | **–5.67** | **5,162.0** | **–5.67** | 92.6 |

*Continued on next page*

Table 1 – *Continued from previous page*

| Instance | $\|\mathcal{P}\|$ | $\|\mathcal{L}\|$ | Branch-and-Bound $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Hybrid heuristic Minimum | Gap (%) | Average | Gap (%) | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| R48 | 80 | 20 | 6,086 | 5,742.3 | 5.99 | 7,200.0 | **5,997** | **–1.46** | **5,997.0** | **–1.46** | 104.2 |
| R49 | 80 | 25 | 4,323 | 4,250.5 | 1.70 | 7,200.0 | 4,323 | 0.00 | 4,323.0 | 0.00 | 96.8 |
| R50 | 80 | 30 | 4,145 | 4,145.0 | 0.00 | 6,707.3 | 4,155 | 0.24 | 4,155.0 | 0.24 | 90.7 |
| R51 | 80 | 35 | 6,226 | 6,042.1 | 3.04 | 7,200.0 | **6,181** | **–0.72** | **6,181.0** | **–0.72** | 123.3 |
| R52 | 80 | 40 | 4,065 | 3,963.9 | 2.55 | 7,200.0 | 4,065 | 0.00 | 4,065.0 | 0.00 | 108.4 |
| R53 | 80 | 45 | 2,424 | 2,424.0 | 0.00 | 2,794.1 | 2,424 | 0.00 | 2,424.0 | 0.00 | 99.3 |
| R54 | 80 | 50 | 3,623 | 3,419.7 | 5.94 | 7,200.0 | 3,623 | 0.00 | 3,623.0 | 0.00 | 118.6 |
| R55 | 90 | 10 | 8,550 | 7,253.3 | 17.88 | 7,200.0 | **7,453** | **–12.83** | **7,453.0** | **–12.83** | 172.5 |
| R56 | 90 | 15 | 6,962 | 5,738.9 | 21.31 | 7,200.0 | **5,919** | **–14.98** | **5,919.0** | **–14.98** | 139.0 |
| R57 | 90 | 20 | 5,075 | 4,347.7 | 16.73 | 7,200.0 | **4,668** | **–8.02** | **4,668.0** | **–8.02** | 111.2 |
| R58 | 90 | 25 | 7,301 | 5,775.9 | 26.40 | 7,200.0 | **6,003** | **–17.78** | **6,003.0** | **–17.78** | 136.7 |
| R59 | 90 | 30 | 5,260 | 3,922.9 | 34.08 | 7,200.0 | **4,638** | **–11.83** | **4,638.0** | **–11.83** | 130.5 |
| R60 | 90 | 35 | 6,093 | 4,032.9 | 51.08 | 7,200.0 | **4,867** | **–20.12** | **4,867.0** | **–20.12** | 160.0 |
| R61 | 90 | 40 | 6,153 | 2,239.9 | 174.70 | 7,200.0 | **3,681** | **–40.18** | **3,681.0** | **–40.18** | 110.2 |
| R62 | 90 | 45 | 2,593 | 1,677.0 | 54.62 | 7,200.0 | **2,553** | **–1.54** | **2,553.0** | **–1.54** | 133.9 |
| R63 | 90 | 50 | 1,586 | 1,586.0 | 0.00 | 3,387.5 | 1,586 | 0.00 | 1,586.0 | 0.00 | 124.8 |
| R64 | 100 | 10 | 10,888 | 6,852.8 | 58.88 | 7,200.0 | **7,034** | **–35.40** | **7,034.0** | **–35.40** | 218.8 |
| R65 | 100 | 15 | 17,775 | 6,712.0 | 164.83 | 7,200.0 | **7,389** | **–58.43** | **7,389.0** | **–58.43** | 216.9 |
| R66 | 100 | 20 | 5,871 | 5,631.2 | 4.26 | 7,200.0 | **5,790** | **–1.38** | **5,790.0** | **–1.38** | 186.1 |
| R67 | 100 | 25 | 7,510 | 5,097.0 | 47.34 | 7,200.0 | **5,720** | **–23.83** | **5,720.0** | **–23.83** | 178.3 |
| R68 | 100 | 30 | 4,000 | 3,554.1 | 12.55 | 7,200.0 | **3,798** | **–5.05** | **3,798.0** | **–5.05** | 139.1 |
| R69 | 100 | 35 | 3,477 | 3,371.3 | 3.14 | 7,200.0 | 3,477 | 0.00 | 3,481.6 | 0.13 | 163.6 |
| R70 | 100 | 40 | 3,633 | 2,744.9 | 32.35 | 7,200.0 | **3,621** | **–0.33** | **3,621.0** | **–0.33** | 149.1 |
| R71 | 100 | 45 | 5,031 | 3,847.7 | 30.75 | 7,200.0 | **4,621** | **–8.15** | **4,621.0** | **–8.15** | 211.4 |
| R72 | 100 | 50 | 873 | 873.0 | 0.00 | 308.1 | 873 | 0.00 | 873.0 | 0.00 | 146.9 |
| Average | | | | | 13.24 | 3,476.8 | | **–4.18** | | **–4.16** | 77.2 |

**Table 2:** Computational Results for the DMS Instances

| Instance | $\|\mathcal{P}\|$ | $\|\mathcal{L}\|$ | Branch-and-Bound $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Hybrid heuristic Minimum | Gap (%) | Average | Gap (%) | Time (s) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| DMS1 | 30 | 10 | 402,805 | 402,805.0 | 0.00 | 123.5 | 402,805 | 0.00 | 402,805.0 | 0.00 | 9.7 |
| DMS2 | 30 | 15 | 320,809 | 320,809.0 | 0.00 | 43.0 | 320,809 | 0.00 | 320,809.0 | 0.00 | 10.4 |
| DMS3 | 30 | 20 | 376,827 | 376,827.0 | 0.00 | 57.6 | 376,827 | 0.00 | 376,827.0 | 0.00 | 10.6 |
| DMS4 | 30 | 25 | 263,178 | 263,178.0 | 0.00 | 10.6 | 263,178 | 0.00 | 263,178.0 | 0.00 | 13.6 |
| DMS5 | 30 | 30 | 280,986 | 280,986.0 | 0.00 | 69.4 | 280,986 | 0.00 | 280,986.0 | 0.00 | 14.3 |
| DMS6 | 30 | 35 | 217,404 | 217,404.0 | 0.00 | 7.9 | 217,404 | 0.00 | 217,404.0 | 0.00 | 14.5 |
| DMS7 | 30 | 40 | 259,006 | 259,006.0 | 0.00 | 22.5 | 259,006 | 0.00 | 259,006.0 | 0.00 | 16.2 |
| DMS8 | 30 | 45 | 289,541 | 289,541.0 | 0.00 | 192.2 | 289,541 | 0.00 | 289,541.0 | 0.00 | 18.9 |
| DMS9 | 30 | 50 | 256,455 | 256,455.0 | 0.00 | 296.2 | 256,455 | 0.00 | 256,455.0 | 0.00 | 18.1 |
| DMS10 | 40 | 10 | 443,380 | 443,380.0 | 0.00 | 265.7 | 443,380 | 0.00 | 443,380.0 | 0.00 | 16.4 |
| DMS11 | 40 | 15 | 420,794 | 420,794.0 | 0.00 | 203.6 | 420,794 | 0.00 | 420,794.0 | 0.00 | 16.6 |
| DMS12 | 40 | 20 | 398,636 | 398,636.0 | 0.00 | 147.0 | 398,636 | 0.00 | 398,636.0 | 0.00 | 13.8 |
| DMS13 | 40 | 25 | 332,717 | 332,717.0 | 0.00 | 166.5 | 332,717 | 0.00 | 332,717.0 | 0.00 | 20.5 |
| DMS14 | 40 | 30 | 295,215 | 295,215.0 | 0.00 | 141.4 | 295,215 | 0.00 | 295,215.0 | 0.00 | 20.8 |

*Continued on next page*

Table 2 – *Continued from previous page*

| Instance | $\|\mathcal{P}\|$ | $\|\mathcal{L}\|$ | Branch-and-Bound | | | | Hybrid heuristic | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Minimum | Gap (%) | Average | Gap (%) | Time (s) |
| DMS15 | 40 | 35 | 310,224 | 310,224.0 | 0.00 | 93.3 | 310,224 | 0.00 | 310,224.0 | 0.00 | 23.1 |
| DMS16 | 40 | 40 | 258,299 | 258,299.0 | 0.00 | 124.3 | 258,299 | 0.00 | 258,299.0 | 0.00 | 20.3 |
| DMS17 | 40 | 45 | 246,563 | 246,563.0 | 0.00 | 344.9 | 246,563 | 0.00 | 246,563.0 | 0.00 | 21.3 |
| DMS18 | 40 | 50 | 255,968 | 255,968.0 | 0.00 | 594.7 | 255,968 | 0.00 | 255,968.0 | 0.00 | 26.4 |
| DMS19 | 50 | 10 | 536,155 | 536,155.0 | 0.00 | 659.5 | 536,155 | 0.00 | 539,870.0 | 0.69 | 24.2 |
| DMS20 | 50 | 15 | 483,182 | 483,182.0 | 0.00 | 343.6 | 483,182 | 0.00 | 483,182.0 | 0.00 | 18.4 |
| DMS21 | 50 | 20 | 308,951 | 308,951.0 | 0.00 | 455.6 | 308,951 | 0.00 | 308,951.0 | 0.00 | 23.3 |
| DMS22 | 50 | 25 | 427,465 | 427,465.0 | 0.00 | 192.5 | 427,465 | 0.00 | 427,465.0 | 0.00 | 26.5 |
| DMS23 | 50 | 30 | 444,803 | 440,315.9 | 1.02 | 7,200.0 | 444,803 | 0.00 | 444,803.0 | 0.00 | 27.2 |
| DMS24 | 50 | 35 | 338,337 | 338,337.0 | 0.00 | 3,451.6 | 338,337 | 0.00 | 338,337.0 | 0.00 | 26.9 |
| DMS25 | 50 | 40 | 369,609 | 369,609.0 | 0.00 | 554.4 | 369,609 | 0.00 | 369,609.0 | 0.00 | 24.5 |
| DMS26 | 50 | 45 | 281,806 | 281,806.0 | 0.00 | 483.6 | 281,806 | 0.00 | 281,806.0 | 0.00 | 37.9 |
| DMS27 | 50 | 50 | 255,322 | 255,322.0 | 0.00 | 336.3 | 255,322 | 0.00 | 255,322.0 | 0.00 | 36.9 |
| DMS28 | 60 | 10 | 510,723 | 510,723.0 | 0.00 | 786.4 | 510,723 | 0.00 | 510,723.0 | 0.00 | 47.9 |
| DMS29 | 60 | 15 | 387,533 | 387,533.0 | 0.00 | 658.3 | 387,533 | 0.00 | 387,533.0 | 0.00 | 34.9 |
| DMS30 | 60 | 20 | 420,748 | 420,748.0 | 0.00 | 1,003.0 | 420,748 | 0.00 | 420,748.0 | 0.00 | 25.8 |
| DMS31 | 60 | 25 | 526,724 | 526,724.0 | 0.00 | 3,515.5 | 526,724 | 0.00 | 526,724.0 | 0.00 | 40.5 |
| DMS32 | 60 | 30 | 461,333 | 431,363.1 | 6.95 | 7,200.0 | 461,333 | 0.00 | 461,333.0 | 0.00 | 45.8 |
| DMS33 | 60 | 35 | 324,238 | 324,238.0 | 0.00 | 411.2 | 324,238 | 0.00 | 324,238.0 | 0.00 | 48.0 |
| DMS34 | 60 | 40 | 330,589 | 330,589.0 | 0.00 | 890.9 | 330,589 | 0.00 | 330,589.0 | 0.00 | 62.4 |
| DMS35 | 60 | 45 | 326,513 | 326,513.0 | 0.00 | 501.3 | 326,513 | 0.00 | 326,513.0 | 0.00 | 46.5 |
| DMS36 | 60 | 50 | 230,152 | 230,152.0 | 0.00 | 313.7 | 230,152 | 0.00 | 230,152.0 | 0.00 | 43.0 |
| DMS37 | 70 | 10 | 554,804 | 554,804.0 | 0.00 | 1,577.3 | 554,804 | 0.00 | 554,804.0 | 0.00 | 90.9 |
| DMS38 | 70 | 15 | 530,990 | 516,662.6 | 2.77 | 7,200.0 | 530,990 | 0.00 | 530,990.0 | 0.00 | 71.6 |
| DMS39 | 70 | 20 | 542,751 | 542,751.0 | 0.00 | 3,133.8 | 542,751 | 0.00 | 542,751.0 | 0.00 | 48.6 |
| DMS40 | 70 | 25 | 447,599 | 447,599.0 | 0.00 | 1,755.0 | 447,599 | 0.00 | 447,599.0 | 0.00 | 56.4 |
| DMS41 | 70 | 30 | 494,524 | 494,524.0 | 0.00 | 2,451.3 | 494,524 | 0.00 | 494,524.0 | 0.00 | 70.4 |
| DMS42 | 70 | 35 | 578,068 | 578,068.0 | 0.00 | 2,670.0 | 578,068 | 0.00 | 578,068.0 | 0.00 | 66.6 |
| DMS43 | 70 | 40 | 561,772 | 544,812.2 | 3.11 | 7,200.0 | **556,447** | **−0.95** | **556,447.0** | **−0.95** | 81.8 |
| DMS44 | 70 | 45 | 380,021 | 380,021.0 | 0.00 | 1,668.1 | 380,021 | 0.00 | 380,021.0 | 0.00 | 83.5 |
| DMS45 | 70 | 50 | 302,074 | 302,074.0 | 0.00 | 463.6 | 302,074 | 0.00 | 302,074.0 | 0.00 | 75.5 |
| DMS46 | 80 | 10 | 548,988 | 487,482.2 | 12.62 | 7,200.0 | **547,449** | **−0.28** | **547,449.0** | **−0.28** | 114.7 |
| DMS47 | 80 | 15 | 446,764 | 425,069.0 | 5.10 | 7,200.0 | 446,764 | 0.00 | 446,764.0 | 0.00 | 123.0 |
| DMS48 | 80 | 20 | 450,842 | 432,862.9 | 4.15 | 7,200.0 | 450,842 | 0.00 | 450,842.0 | 0.00 | 71.3 |
| DMS49 | 80 | 25 | 450,048 | 408,033.4 | 10.30 | 7,200.0 | **426,489** | **−5.23** | **426,489.0** | **−5.23** | 80.5 |
| DMS50 | 80 | 30 | 496,847 | 426,685.8 | 16.44 | 7,200.0 | **480,258** | **−3.34** | **480,258.0** | **−3.34** | 69.6 |
| DMS51 | 80 | 35 | 485,028 | 485,028.0 | 0.00 | 4,725.1 | 485,028 | 0.00 | 485,028.0 | 0.00 | 105.2 |
| DMS52 | 80 | 40 | 471,189 | 471,189.0 | 0.00 | 3,664.5 | 471,189 | 0.00 | 471,189.0 | 0.00 | 113.3 |
| DMS53 | 80 | 45 | 623,723 | 492,884.7 | 26.55 | 7,200.0 | **549,577** | **−11.89** | **549,577.0** | **−11.89** | 83.8 |
| DMS54 | 80 | 50 | 1,126,035 | 478,220.6 | 135.46 | 7,200.0 | **575,217** | **−48.92** | **575,217.0** | **−48.92** | 114.0 |
| DMS55 | 90 | 10 | 560,816 | 516,511.6 | 8.58 | 7,200.0 | **548,753** | **−2.15** | **548,753.0** | **−2.15** | 137.2 |
| DMS56 | 90 | 15 | 512,996 | 376,683.2 | 36.19 | 7,200.0 | **467,334** | **−8.90** | **467,334.0** | **−8.90** | 116.6 |
| DMS57 | 90 | 20 | 620,039 | 588,981.6 | 5.27 | 7,200.0 | **617,562** | **−0.40** | **617,562.0** | **−0.40** | 103.8 |
| DMS58 | 90 | 25 | 647,540 | 546,252.0 | 18.54 | 7,200.0 | **594,855** | **−8.14** | **594,855.0** | **−8.14** | 95.3 |
| DMS59 | 90 | 30 | 462,562 | 417,127.1 | 10.89 | 7,200.0 | **441,713** | **−4.51** | **441,713.0** | **−4.51** | 123.1 |
| DMS60 | 90 | 35 | 422,362 | 383,535.5 | 10.12 | 7,200.0 | **420,702** | **−0.39** | **420,702.0** | **−0.39** | 116.3 |
| DMS61 | 90 | 40 | 573,543 | 518,992.3 | 10.51 | 7,200.0 | **553,063** | **−3.57** | **553,063.0** | **−3.57** | 113.9 |
| DMS62 | 90 | 45 | 632,222 | 595,328.2 | 6.20 | 7,200.0 | **617,045** | **−2.40** | **617,045.0** | **−2.40** | 134.8 |

*Continued on next page*

Table 2 – *Continued from previous page*

| | | | Branch-and-Bound | | | | Hybrid heuristic | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | $\|\mathcal{P}\|$ | $\|\mathcal{L}\|$ | $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Minimum | Gap (%) | Average | Gap (%) | Time (s) |
| DMS63 | 90 | 50 | 653,711 | 470,398.9 | 38.97 | 7,200.0 | **579,540** | **–11.35** | **579,540.0** | **–11.35** | 163.7 |
| DMS64 | 100 | 10 | 860,960 | 592,047.2 | 45.42 | 7,200.0 | **656,683** | **–23.73** | **656,683.0** | **–23.73** | 204.9 |
| DMS65 | 100 | 15 | 582,832 | 539,966.7 | 7.94 | 7,200.0 | **564,899** | **–3.08** | 564,946.0 | –3.07 | 172.7 |
| DMS66 | 100 | 20 | 1,142,970 | 608,207.6 | 87.92 | 7,200.0 | **647,355** | **–43.36** | **647,355.0** | **–43.36** | 128.8 |
| DMS67 | 100 | 25 | 661,951 | 535,501.9 | 23.61 | 7,200.0 | **595,560** | **–10.03** | **595,560.0** | **–10.03** | 116.7 |
| DMS68 | 100 | 30 | 593,085 | 559,529.2 | 6.00 | 7,200.0 | **590,941** | **–0.36** | **590,941.0** | **–0.36** | 150.6 |
| DMS69 | 100 | 35 | 450,905 | 450,905.0 | 0.00 | 5,184.3 | 450,905 | 0.00 | 450,905.0 | 0.00 | 156.8 |
| DMS70 | 100 | 40 | 934,901 | 612,546.5 | 52.63 | 7,200.0 | **647,879** | **–30.70** | **647,879.0** | **–30.70** | 155.6 |
| DMS71 | 100 | 45 | 716,290 | 411,011.1 | 74.28 | 7,200.0 | **532,112** | **–25.71** | **532,112.0** | **–25.71** | 149.9 |
| DMS72 | 100 | 50 | 589,691 | 474,491.8 | 24.28 | 7,200.0 | **514,801** | **–12.70** | **514,801.0** | **–12.70** | 150.1 |
| Average | | | | | 9.61 | 3,421.6 | | **–3.64** | | **–3.63** | 68.9 |

For the random instances with up to 100 patients, the branch-and-bound algorithm finds an optimal solution for 46 out of 72 instances within the predefined time limit. The average gap between $\overline{z}$ and $\underline{z}$ is 13.24% and the average computation time is equal to 3476.8 seconds. Out of the 46 instances for which an optimal solution is proven, the hybrid heuristic finds it for 41 instances in all ten runs and it finds the optimal solution for 44 instances in at least one run. The average gap between *Minimum* and $\overline{z}$ is –4.18% and between *Average* and $\overline{z}$ is –4.16%, which shows that our heuristic is very robust and stable. This implies that the average solution found over ten runs by the hybrid heuristic improves the best solution found by the branch-and-bound algorithm within the predefined time limit on average with 4.16%. The average computation time for the hybrid solution is 77.2 seconds and is therefore significantly less than the computation time for the branch-and-bound algorithm.

For the DMS instances with up to 100 patients, the branch-and-bound algorithm finds an optimal solution for 44 out of 72 instances within the predefined time limit. The hybrid heuristic finds the optimal solution for all of these instances in at least one run and for all but one of the instances in all ten runs. The average gap between $\overline{z}$ and $\underline{z}$ is 9.61%. The average computation time is 3421.6 seconds. The average gap between *Minimum* (*Average*) and $\overline{z}$ is –3.64% (–3.63%). The average computation time for the hybrid solution is 68.9 seconds and is therefore significantly less than the computation time for the branch-and-bound algorithm.

## 4.4 Sensitivity Analysis

In order to evaluate how the various opening and closing locker procedures affect the performance of our hybrid heuristic, we have modified it to a version we call *random heuristic*. In this version the procedure of opening and the procedure of closing a locker is only based on the random method, i.e., $\eta_3 = \gamma_3 = 1$.

We have then evaluated all instances in the sets of random and DMS instances, where the instances are grouped based on the number of potential locker locations. Table 3 reports comparative results for the hybrid heuristic and the random heuristic. For each group, the table reports $|\mathcal{L}|$, the number of potential locker locations, and for both the hybrid heuristic and the random heuristic it reports *Gap* (%), the average percentage gap to the best known solutions. For each instance, the gap is computed by $(z_i - z_{best})/z_{best} \cdot 100$, where $z_i$, $i \in \{hybrid, random\}$ is the average solution value over ten runs of heuristic $i$ and $z_{best}$ is the solution value of the best solution found over ten runs of the hybrid heuristic and ten runs of the random heuristic. The running times of the two heuristics are quite similar.

**Table 3:** Average gap with best known solutions over all instances

| | Hybrid Heuristic | Random Heuristic |
|---|---|---|
| $\|\mathcal{L}\|$ | Gap (%) | Gap (%) |
| 10 | 0.13 | 0.17 |
| 15 | 0.00 | 0.40 |
| 20 | 0.00 | 1.97 |
| 25 | 0.00 | 3.91 |
| 30 | 0.00 | 5.09 |
| 35 | 0.01 | 9.63 |
| 40 | 0.00 | 11.43 |
| 45 | 0.00 | 14.15 |
| 50 | 0.01 | 17.87 |
| Average | 0.02 | 7.18 |

The results in Table 3 show the effect of incorporating more sophisticated methods for opening and closing lockers. The hybrid heuristic as proposed in this paper, performs on average 0.13% worse compared to the best found solutions. On the other hand, the random heuristic that only incorporates a random method for opening and closing a locker performs on average 6.74% worse compared to the best found solutions. Moreover, while the performance of the hybrid heuristic is quite stable when increasing the number of potential locker locations, the gap with respect to the best known solutions increases with an increasing number of potential locker locations for the random heuristic.

Furthermore, we have formally assessed the stability of our hybrid heuristic by calculating the *coefficient of variation* ($CV$), which is a dimensionless and normalized measure of dispersion of a distribution. The $CV$ is defined as the ratio of the standard deviation ($s$) to the mean ($\overline{X}$), $CV = s/\overline{X}$. The lower the $CV$ value, the more stable and robust the algorithm is. The $CV$ for the DMS and random instances, computed over the 10 runs reported in the previous tables, is not larger than 0.013 for each instance, which shows how strong and stable our algorithm is.

## 4.5 Managerial Insights

In this section we show the effects of changing the parameter values $r_j$, $F_j$, and $\phi$, which have a deep effect on the quality of service and offer many managerial insights. We have created a data set consisting of a subset of 50 instances from the DMS instance set. For these instances, we changed the values of the parameters $r_j$, $F_j$, and $\phi$ by multiplying them by 0.25, 0.50, 0.75, 1.00, and 1.50 for the parameters $r_j$ and $F_j$, and 0.20, 0.40, 0.60, 0.80, and 1.00 for the parameter $\phi$. Note that the base case corresponds to a factor of 1.00. For each instance and each parameter value, we run the heuristic ten times and use the result with the smallest objective. The results are shown in Figures 2, 3, and 4, where the average length of the locker and patient routes and the average number of opened lockers over all instances are reported for the different parameter values.

New regulations may impose a change in the coverage distance. As can be seen in Figure 2, if the coverage distance decreases, the number of opened lockers decreases and thereby the length of the locker routes decreases, whereas the length of the patient routes increases. Thus, a decrease in coverage distance reduces the number of opened lockers and thereby increases the number of home deliveries. On the other hand, an increase in coverage distance results in less home deliveries and more patients served by the lockers. This effect occurs since a larger coverage distance implies that more patients are within a locker's range, which in turn implies that a locker is more cost-effective.

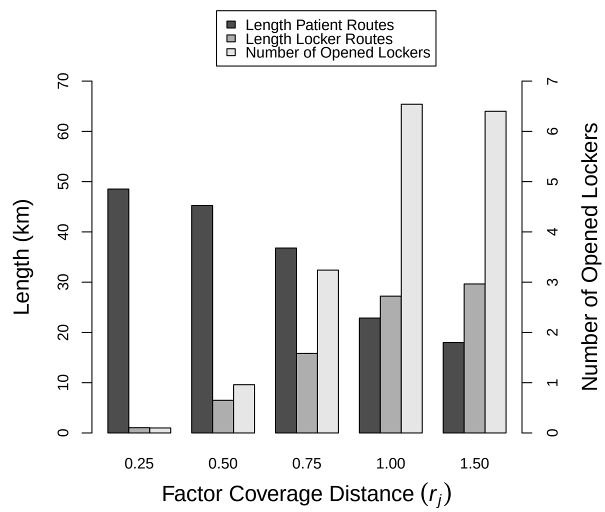


**Figure 2:** The effect of changing the value for $r_j$

Figure 3 shows the effects of the opening costs of the lockers. The number of opened lockers and thereby the length of the locker routes decrease when the opening costs increase, whereas the length of the patient routes increase. We see the opposite effect when the opening costs decrease. Hence, decreasing the opening costs for the lockers results in more opened lockers and thereby less home deliveries.

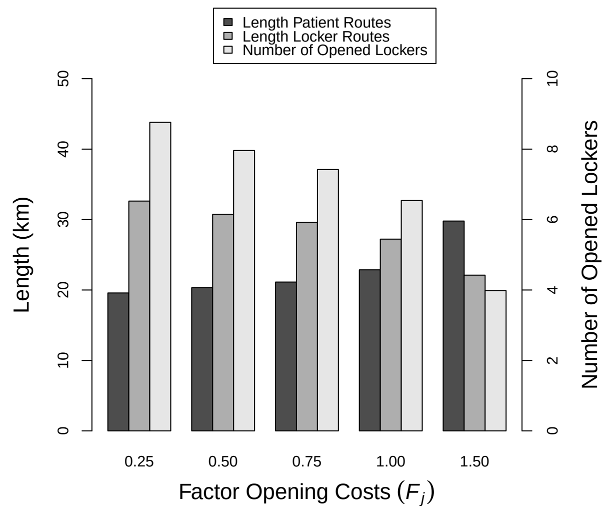


**Figure 3:** The effect of changing the value for $F_j$

The effect of the value for the parameter $\phi$ is shown in Figure 4. The value of $\phi$ reflects the resistance of a company against home delivery. The higher the value of $\phi$, the higher the stimulus towards delivery through lockers. Thus, with an increasing value of $\phi$, the number of opened lockers is increasing and thereby the length of the locker routes is increasing, whereas fewer patients need home delivery and thus the length of the patient routes is decreasing.

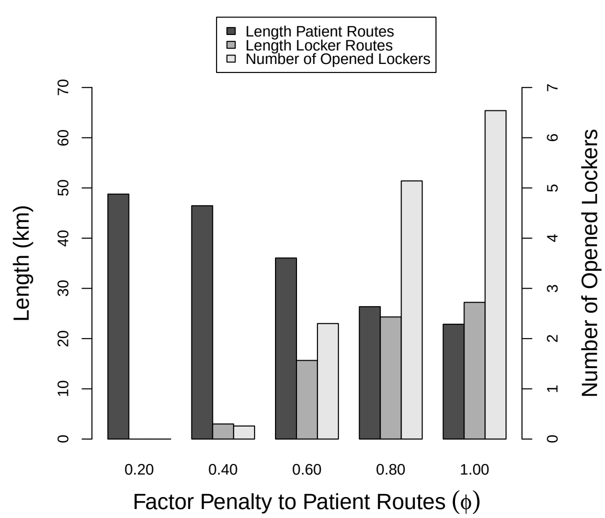


**Figure 4:** The effect of changing the value for $\phi$

# 5 Conclusions

In this paper, we have introduced, modeled, and solved a simultaneous facility location and vehicle routing problem that arises in health care logistics in the Netherlands, and can also be applied to other areas, such as the location of pick-up lockers for e-commerce operations. In this problem, the distribution of medication from a local pharmacy to a set of patients can occur via locker locations, from where patients within the coverage distance of the locker can collect their medication, or by home delivery. The goal of this integrated problem is to determine which lockers in a set of potential locker locations to open, to generate vehicle routes that visit the opened lockers and to generate the vehicle routes that visit the patients that are not within the coverage distance of one of the opened lockers, such that the total travel and locker costs are minimized. We are the first to formally define this practice-inspired problem and we give a mathematical formulation that can be solved by a branch-and-bound algorithm. Moreover, we propose a hybrid heuristic to solve the problem. We propose two sets of benchmark instances, a randomly generated set and an instance set based on data from our industrial partner DMS. We show that our branch-and-bound algorithm is able to solve instances with up to 100 patients and 50 potential locker locations to optimality within a time limit of 7200 seconds. Our hybrid heuristic algorithm was able to consistently outperform CPLEX using only a fraction of the running time. Our solutions were up to 4.18% better for the random set and 3.64% better on the real DMS set (average over the best of 10 runs per instance) than the branch-and-bound solution obtained within a time limit of 7200 seconds. Moreover, we have shown that our heuristic is very stable, capable of yielding the same (best) results consistently over the 10 runs per instance. We have also shown that the incorporation of sophisticated methods for opening a locker and closing a locker has a large impact on the results of our hybrid heuristic. Results show that when the opening cost of the lockers decrease or the coverage distance of the lockers increases, the number of opened lockers increases and thereby the number of home deliveries decreases.

# Appendix

An additional set of instances are modified from a set of benchmarks for the LRP. These specify the x-coordinates and y-coordinates of the patients and the depots. For each instance, we adopted the coordinates, randomly selected one of the depots to serve as the depot for our instance and we marked the remaining depots as potential locker locations. This results in a set of 15 instances with up to 14 potential locker locations and up to 150 patients. We adopted the opening cost of a depot as the opening cost of the corresponding locker. The remaining parameters are generated similarly to the random instances. The generation of the service time and the coverage distance is slightly changed such that the proportion to the distances is the same as for the random instances.

The results for these instances are given in Table 4. The format of the table is identical to that of Tables 1 and 2. The results in this table are comparable to the results for the random and DMS instances. Since the set of potential locker location $\mathcal{L}$ is relatively small, these instances are somewhat less interesting and are therefore incorporated in the Appendix. Unfortunately, as far as we are aware, there are no benchmark instances for simultaneous routing and location problems that consider larger sets of locker locations.

**Table 4:** Computational Results for Additional Instances

| | | | Branch-and-Bound | | | | Hybrid heuristic | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | $\lvert\mathcal{P}\rvert$ | $\lvert\mathcal{L}\rvert$ | $\overline{z}$ | $\underline{z}$ | Gap (%) | Time (s) | Minimum | Gap (%) | Average | Gap (%) | Time (s) |
| Christofides69 | 50 | 4 | 3,387 | 3,387.0 | 0.00 | 342.9 | 3,387 | 0.00 | 3,387.0 | 0.00 | 54.0 |
| Christofides69 | 75 | 9 | 3,354 | 3,354.0 | 0.00 | 1,416.9 | 3,354 | 0.00 | 3,354.0 | 0.00 | 94.0 |
| Christofides69 | 100 | 9 | 4,072 | 3,555.1 | 14.54 | 7,200.0 | **3,890** | **−4.47** | **3,890.0** | **−4.47** | 194.6 |
| Daskin95 | 88 | 7 | 460 | 460.0 | 0.00 | 18.4 | 460 | 0.00 | 460.0 | 0.00 | 86.8 |
| Daskin95 | 150 | 9 | 228,182 | 121,516.2 | 87.78 | 7,200.0 | **174,948** | **−23.33** | **174,948.0** | **−23.33** | 456.7 |
| Gaskell67 | 21 | 4 | 2,518 | 2,518.0 | 0.00 | 6.3 | 2,518 | 0.00 | 2,518.0 | 0.00 | 7.3 |
| Gaskell68 | 22 | 4 | 5,400 | 5,400.0 | 0.00 | 44.4 | 5,400 | 0.00 | 5,400.0 | 0.00 | 6.7 |
| Gaskell69 | 29 | 4 | 4,710 | 4,056.1 | 16.12 | 7,200.0 | 4,710 | 0.00 | 4,710.0 | 0.00 | 13.7 |
| Gaskell70 | 32 | 4 | 4,090 | 4,090.0 | 0.00 | 328.6 | 4,090 | 0.00 | 4,124.0 | 0.83 | 15.8 |
| Gaskell71 | 36 | 4 | 2,670 | 2,670.0 | 0.00 | 58.6 | 2,670 | 0.00 | 2,670.0 | 0.00 | 13.5 |
| Min92 | 27 | 4 | 23,952 | 23,952.0 | 0.00 | 90.3 | 23,952 | 0.00 | 24,064.0 | 0.47 | 6.6 |
| Min93 | 134 | 7 | – | 29,931.3 | – | 7,200.0 | **35,933** | – | **35,933.0** | – | 513.1 |
| Or76 | 117 | 13 | – | 58,066.2 | – | 7,200.0 | **73,323** | – | **73,323.0** | – | 172.3 |
| Perl84 | 55 | 14 | 2,528 | 2,528.0 | 0.00 | 1,677.7 | 2,528 | 0.00 | 2,528.0 | 0.00 | 25.9 |
| Perl85 | 85 | 6 | 2,065 | 2,065.0 | 0.00 | 1,754.2 | 2,065 | 0.00 | 2,065.0 | 0.00 | 94.9 |
| Average | | | | | 9.11 | 2,782.5 | | **−2.14** | | **−2.04** | |

# References


[1] Somayeh Allahyari, Majid Salari, and Daniele Vigo. A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem. *European Journal of Operational Research*, 242(3):756–768, 2015.

[2] T Ávila, Á Corberán, I Plana, and J M. Sanchis. A new branch-and-cut algorithm for the generalized directed rural postman problem. *Transportation Science*, 50(2): 750–761, 2016.

[3] R. Cuda, G. Guastaroba, and M. G. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2015.

[4] Jing Dong, Ning Yang, and Ming Chen. Heuristic approaches for a TSP variant: The automatic meter reading shortest tour problem. In Edward K. Baker, Anito Joseph, Anuj Mehrotra, and Michael A. Trick, editors, *Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies*, pages 145–163. Springer US, 2007.

[5] Michael Drexl. On the generalized directed rural postman problem. *Journal of the Operational Research Society*, 65(8):1143–1154, 2014.

[6] Michel Gendreau, Gilbert Laporte, and Frédéric Semet. The covering tour problem. *Operations Research*, 45(4):568–576, 1997.

[7] Bruce L. Golden, Zahra Naji-Azimi, S. Raghavan, Majid Salari, and Paolo Toth. The generalized covering salesman problem. *INFORMS Journal on Computing*, 24 (4):534–553, 2012.

[8] Damon J. Gulczynski, Jeffrey W. Heath, and Carter C. Price. The close enough traveling salesman problem: A discussion of several heuristics. In Francis B. Alt, Michael C. Fu, and Bruce L. Golden, editors, *Perspectives in Operations Research: Papers in Honor of Saul Gass' 80th Birthday*, pages 271–283. Springer US, 2006.

[9] Minh Hoàng Hà, Nathalie Bostel, André Langevin, and Louis-Martin Rousseau. An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, 226(2):211–220, 2013.

[10] Minh Hoàng Hà, Nathalie Bostel, André Langevin, and Louis-Martin Rousseau. Solving the close-enough arc routing problem. *Networks*, 63(1):107–118, 2014. ISSN 1097-0037.

[11] Mondher Hachicha, M John Hodgson, Gilbert Laporte, and Frédéric Semet. Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, 27(1):29–42, 2000.

[12] Gerard A P Kindervater and Martin W P Savelsbergh. Vehicle routing: handling edge exchanges. In Emile Aarts and Jan Karel Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 337–360. Wiley, Chichester, 1997.

[13] Scott Kirkpatrick, C D Gelatt Jr., and M P Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[14] William Kenneth Mennell. *Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, sequence-dependent team orienteering problem*. PhD thesis, University of Maryland, 2009.

[15] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997. ISSN 0305-0548.

[16] Gábor Nagy and Said Salhi. Nested heuristic methods for the location-routeing problem. *Journal of the Operational Research Society*, 47(9):1166–1174, 1996.

[17] Gabor Nagy and Said Salhi. Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672, 2007.

[18] Z. Naji-Azimi, J. Renaud, A. Ruiz, and M. Salari. A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, 222(3):596–605, 2012.

[19] Caroline Prodhon and Christian Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

[20] Majid Salari and Zahra Naji-Azimi. An integer programming-based local search for the covering salesman problem. *Computers & Operations Research*, 39(11):2594–2602, 2012.

[21] Andreas Stenger, Michael Schneider, Michael Schwind, and Daniele Vigo. Location routing for small package shippers with subcontracting options. *International Journal of Production Economics*, 140(2):702–712, 2012.

[22] Fabien Tricoire, Alexandra Graf, and Walter J. Gutjahr. The bi-objective stochastic covering tour problem. *Computers & Operations Research*, 39(7):1582–1592, 2012.

[23] Bo Yuan, Maria Orlowska, and Shazia Sadiq. On the optimal robot routing problem in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(9):1252–1261, 2007. ISSN 1041–4347.

[24] E E Zachariadis and C T Kiranoudis. A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & Operations Research*, 37(12):2089–2105, 2010.