Hindawi Publishing Corporation
Advances in Operations Research
Volume 2012, Article ID 393890, 25 pages
doi:10.1155/2012/393890

Research Article

Exact and Heuristic Solutions to Minimize Total Waiting Time in the Blood Products Distribution Problem

Amir Salehipour[1] and Mohammad Mehdi Sepehri[1,2]

[1] *Department of Industrial Engineering, School of Engineering, Tarbiat Modares University, Tehran 14117-13114, Iran*
[2] *Hospital Management Research Center, Tehran University of Medical Sciences, Tehran 19697-14713, Iran*

Correspondence should be addressed to Mohammad Mehdi Sepehri, mehdi.sepehri@gmail.com

Received 18 October 2011; Revised 29 March 2012; Accepted 14 May 2012

Academic Editor: Silvano Martello

This paper presents a novel application of operations research to support decision making in blood distribution management. The rapid and dynamic increasing demand, criticality of the product, storage, handling, and distribution requirements, and the different geographical locations of hospitals and medical centers have made blood distribution a complex and important problem. In this study, a real blood distribution problem containing 24 hospitals was tackled by the authors, and an exact approach was presented. The objective of the problem is to distribute blood and its products among hospitals and medical centers such that the total waiting time of those requiring the product is minimized. Following the exact solution, a hybrid heuristic algorithm is proposed. Computational experiments showed the optimal solutions could be obtained for medium size instances, while for larger instances the proposed hybrid heuristic is very competitive.

1. Introduction

This paper presents a novel application of operations research to support decision making in blood distribution management with the objective of minimizing the total waiting time of those hospitals and medical centers requesting for blood products. The increasing demand for healthcare services coupled with their importance and higher costs make it obligatory to better utilize the medical resources and facilities. The rapid and dynamic increasing demand for blood, criticality of the product, storage, handling, and distribution requirements and limitations, and the different geographical locations of hospitals and medical centers have made blood distribution a complex and very important problem. Furthermore, increasing rate of surgeries coupled with the new advances in healthcare have magnified the complexity

and importance of an efficient blood distribution system. If the right blood products are not available at the hospital and medical centers at the right time, then health issues or delays of operations may arise, which results in extra days of hospitalization and cost. Besides, overstocking blood at hospitals and medical centers leads to low utilization, as most blood products may only be used to a patient of the same blood type within 21 days of collection. For these reasons, blood must be collected regularly. Thus, any low utilization increases costs and wastes the scarce blood resource. The latter may have fatal consequences.

The human blood is one of the most important components of a healthcare system. According to [1], the most important reasons for the need for this vital medical resource are as follows.

(i) There is no exact substitute for human blood.

(ii) Blood and products made from blood play an important role in advances in life-saving techniques.

(iii) Blood products cannot be stored for an indefinite period (e.g., according to [2], red blood cells must be used within 35–42 days of collection; platelets have shorter life, as they must be used within five days of collection).

Blood is composed of many components (red cells, white cells, platelets, plasma). Each of these components supplies a separate function in the human organism and has a different use in medical treatment. Despite the various requirements to the number of unit of blood, almost all critical medical treatments and operations require this vital life resource, among which are accidental victims and injuries, surgeries, organ transplantations, several cancer treatments, and so forth. For instance, 8 units of platelets may be required on a daily basis by a patient undergoing leukemia cancer treatment [1].

The previous studies on the blood products distribution have focused on the inventory related problems [3–7]. However, there is a lack of study regarding the problem of distributing blood and blood products among hospitals and medical centers with the objective of minimizing the total waiting time of hospitals waiting for blood arrival. Delen et al. studied blood supply chain management by analyzing inventory consumption patterns and supply chain status. They implemented the approach in the supply chain facilities at the two United States Air Force Bases [1]. Katsaliaki performed a simulation study towards a more cost-effective management of blood supply chain in the United Kingdom. The study revealed substantial improvements in inventory and distribution operations involved and resulted in cost reductions and increased safety [8]. Katsaliaki and Brailsford studied ordering policies with respect to reductions in shortages and wastage, associated costs, as well as improving service levels and safety by applying a discrete-event simulation model [9]. Hemmelmayr et al. studied the problem of a cost-effective delivery of blood products to Austrian hospitals. They presented solution approaches based on integer programming and Variable Neighborhood Search meta-heuristic [10]. Although less related, Rego and Sousa performed a thorough study on the design of hospital supply chains systems and developed a hybrid Tabu Search-Variable Neighborhood Search meta-heuristic for the problem [11].

In this study, a real-blood distribution problem was considered. The case was experienced by authors in the city of Tehran, Iran. The authors were supposed to derive practical solutions, preferably optimal distribution of blood products among hospitals and medical centers, with the objective of minimizing total waiting time of those requiring the product. As solution procedures, a mixed-integer programming formulation was presented followed by a hybrid heuristic to find optimal distribution of blood products among hospitals and

medical centers such that the total waiting time of those requiring the product is minimized. Computational experiments showed the mixed-integer programming formulation is quite capable of finding optimal solutions for medium-size instances, and for the real cases where 24 hospitals were serviced, while for larger random instances, the hybrid heuristic derives near optimal solutions very quickly. The remainder of this paper is organized as follows. In Section 2, we give an exact definition of the problem together with the notations used in this paper. In Section 3, we develop a new mixed-integer programming formulation for the problem. Following incapability of the exact procedures in deriving optimal solution for large instances in reasonable time, Section 4 is devoted to the hybrid heuristic algorithm developed to solve the large-scale instances of the problem. In Section 5, computational experiments are reported. The paper ends with conclusion.

2. Problem Definition

This section defines the problem, and the notations used throughout this paper. Apart from many improvements applied to blood donor service processes since its foundation in 1921, blood products supply chain and distribution process are composed of the four steps:

(1) collection of blood from donors,

(2) testing, processing and storage of blood,

(3) delivery of blood products,

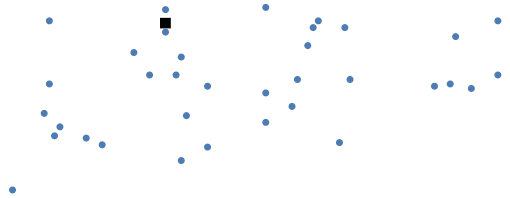
(4) consumption of blood products to recipients.

The problem studied in this paper is inherent in process three, and it is to deliver the blood to hospitals and medical centers upon their requests such that the total waiting time of those requesting for is minimized. For the sake of simplicity, when "hospital" is used, we mean both hospitals and medical centers, or any other medical institute where blood and blood products are required. Typically, a central blood transfusion depot distributes blood among hospitals. Apart from impartiality in blood distribution to hospitals when a central depot is implemented, the total costs and waiting times can be minimized while managing the whole operation is much simpler. Besides, it is not appropriate to keep the blood for a long time and several blood products have a very short life, hence, keeping them for even days results in unusable blood products which cannot be transferred to patients. These limit banking large volume of blood by hospitals. Apart from these limitations which build this problem very interesting and challengeable, the operational limitations avoid a single blood distribution vehicle to cover all hospitals on a single route. At the most, these operational limitations arise from the fact that the blood should be delivered to hospitals before the latest time (an upper bound on the delivery time), and also each vehicle has a limited capacity. In fact, in reality each vehicle covers a set of hospitals where a route is built.

An example of the problem is illustrated in Figure 1 where a set of 50 hospitals are served by two vehicles. The vehicles leave from blood transfusion depot (the black rectangle). Initial studies show that at least two vehicles are required to cover these 50 hospitals. Thus, two decisions should be made: Which hospitals should be assigned to each vehicle? What the objective function would be?, and How the permutation of visiting each hospital by each vehicle (in fact, in each route) should be determined? (Again, what the objective function would be?). In this study, these two decisions are made separately by employing two different mathematical formulations, where, at first, hospitals are assigned to blood

(a) The geographical location of 50 hospitals

(b) A set of 15 hospitals covered by vehicle 1 (the visiting route is not shown)



(c) A set of 35 hospitals covered by vehicle 2 (the visiting route is not shown)

Figure 1: An illustration of the problem where 50 hospitals are covered by two vehicles.

distribution vehicles (or simply constructing routes), and then permutation of visiting hospitals by each vehicle are optimized, according to the appropriate objective function criterion. To better understand the second problem, that is, optimal permutation of visiting hospitals in each route, the following example is provided.

Assume the problem of Figure 1(b), where a set of 15 hospitals is covered by a single vehicle (vehicle 1). Without loss of generality and for the sake of better understanding, assume that the travel time is our only component of the objective function. Table 1 shows these travel times as a symmetric travel time matrix. We hold "0" as the blood transfusion depot of vehicles, or simply depot according to the traveling salesman problem (TSP) and vehicle routing problem (VRP) literatures. We define the "arrival time" to a hospital as the time required for the vehicle to visit this hospital from depot for the first time. For instance, according to Figure 2(a), the permutation of visiting hospitals is

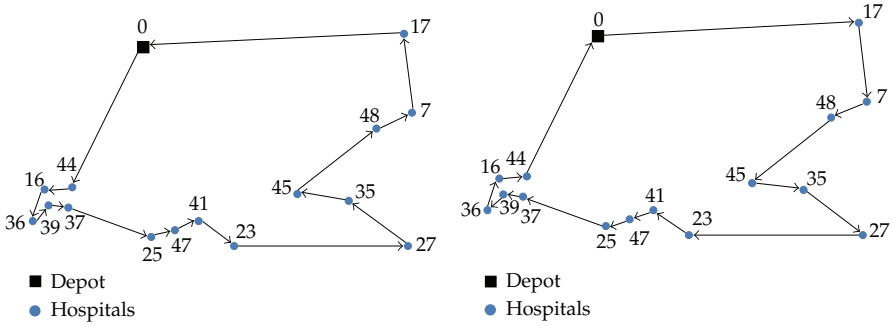
$$p_{3a} = \{0, 44, 16, 36, 39, 37, 25, 47, 41, 23, 27, 35, 45, 48, 7, 17, 0\}. \tag{2.1}$$

Thus arrival time to hospital 44 would be 64 (see Table 1). This is the total waiting time of hospital 44 to be visited by the vehicle 1.

Similarly, the arrival time to the hospital 16 would be the arrival time to hospital 44 (as hospital 16 cannot be visited any sooner than hospital 44 in permutation p_{3a}) plus the travel time from hospital 44 to hospital 16. Thus, $a_{16} = 44 + t_{44,16} = 64 + 7 = 71$.

Table 1: The symmetric travel time matrix of example in Figure 1(b).

	0	7	16	17	23	25	27	35	36	37	39	41	44	45	47	48
0	0	73	67	66	90	84	110	85	81	73	74	78	64	75	81	69
7		0	99	35	74	85	59	42	107	96	100	72	92	46	79	11
16			0	114	54	34	95	77	14	10	7	41	7	64	37	88
17				0	103	110	94	75	125	114	117	97	108	75	104	42
23					0	21	44	35	52	45	50	14	48	28	16	63
25						0	65	52	30	24	29	13	29	41	6	74
27							0	25	95	87	92	54	88	36	59	52
35								0	80	71	76	39	70	13	45	32
36									0	10	8	42	18	68	36	96
37										0	5	33	9	58	28	85
39											0	38	10	63	33	89
41												0	35	27	7	60
44													0	57	32	81
45														0	34	35
47															0	68
48																0



(a) A route of visiting 15 hospitals (Figure 1(b)). Here, the direction of the route is from left to right

(b) A route of visiting 15 hospitals (Figure 1(b)) where the direction of the route is from right to left

Figure 2: Two permutations of visiting hospitals of Figure 1(b). Note that the permutation presented in (b) has a worse objective function than the one in (a).

Obviously, this arrival time concept is similar to the waiting time concept. In fact, what we are looking for is to decide how to locate the hospitals so as to minimize the total waiting time of all hospitals assigned to a vehicle. It is not surprising that the most important issue when developing healthcare systems, especially in an emergency situation, is the waiting time. Figures 2(a) and 2(b) illustrate two examples of waiting time issue, where changing only direction of visits in Figure 2(b) in comparison with Figure 2(a) leads to another solution with a different objective function value (assuming routes are constructed). Minimizing the total waiting time of hospitals waiting for blood can be modeled within the framework of the traveling repairman problem ((TRP), (other names are minimum latency problem (MLP) and travelling delivery-man problem (TDP)). The TRP has been proved to be NP-Hard and even more difficult than the TSP [12, 13]. Recently, Salehipour and Sepehri have developed a strong mixed-integer mathematical programming formulation for the problem based on

the flow network problem [14]. Fischetti et al. [15] and Eijl [16] have developed two mathematical programming formulations for the TRP, however, their formulations are much weaker than that of [14], as their models cannot solve even small instance size of 15 hospitals in a reasonable time. Sarubbi and Luna presented a different mixed-integer mathematical programming model at the International Network Optimization Conference in 2007 [17]. However, their model still lacked certain sets of constraints and resulted in infeasibility. Mendez-Diaz et al. [18] formulated TRP based on the linear ordering problem. Although model's performance is much better than previous ones, its computational time is not comparable to that of Salehipour and Sepehri [14]. Later, we report on dimensions (total number of decision variables and constraints), and the computational performance of these models.

Note that here a solution is represented as a permutation of hospitals starting from 0 and ending in 0, that is, $p = \{0, \ldots, i, j, \ldots, 0\}$. From this representation, the objective function value can be easily calculated. Assume the solution $p = \{0, 1, 2, \ldots, n, 0\}$ with 1 being the first hospital in the solution; 2, the second, and so on. The total arrival time (equivalently waiting time) for this solution can be calculated as

$$F_p = \sum_{i=1}^{n} a_i, \tag{2.2}$$

where

$$a_i = \sum_{j=1}^{j \leq i} t_{j-1,j}. \tag{2.3}$$

Combining (2.2) and (2.3) and simplifying it result

$$F_p = \sum_{i=1}^{n} ((n+1) - i + 1) t_{i-1,i} + t_{n,0}. \tag{2.4}$$

Equation (2.4) offers the advantage of making the calculations much easier and quicker, especially in heuristic algorithms. Let us back to Example of Figure 2. The permutation p_{3a} has a better (lower) objective function value than the permutation p_{3b}:

$$\begin{aligned} F_{p_{3a}} &= \sum_{i=0}^{n} a_i \\ &= 16(64) + 15(7) + 14(14) + 13(8) + 12(5) + 11(24) + 10(6) + 9(7) \\ &\quad + 8(14) + 7(44) + 6(25) + 5(13) + 4(35) + 3(11) + 2(35) + 1(66) \\ &= 2820, \end{aligned}$$

$$
\begin{aligned}
F_{p_{3b}} &= \sum_{i=0}^{n} a_i \\
&= 16(66) + 15(35) + 14(11) + 13(35) + 12(13) + 11(25) + 10(44) \\
&\quad + 9(14) + 8(7) + 7(6) + 6(24) + 5(5) + 4(8) + 3(14) + 2(7) + 1(64) \\
&= 3606.
\end{aligned}
\tag{2.5}
$$

2.1. Notations

We assume that the time component consists of the total traveling time of blood distribution vehicles between every two arbitrary hospitals, i and j, and that the service time (the delivery time) is assumed to be same for all vehicles and hospitals, and hence is ignored. Even not ignored, this delivery time can be dealt with easily [19]. We also assume "0" as the initial position of all vehicles.

2.1.1. Decision Variables

The following hold.

y_{ij} is a decision variable which takes 1 if the vehicle travels from hospital i to hospital j; and 0, otherwise.

x_{ij} is a decision variable which takes $n - u + 1$, if hospital j is visited after hospital i and is visited by a vehicle in order u; and 0, otherwise.

p_{uj} is a decision variable which takes 1 if hospital j is visited in order u; and 0, otherwise.

δ_{vi} is a decision variable which takes 1 if hospital i is visited by vehicle v; and 0, otherwise.

γ_v is a decision variable which takes 1 if vehicle v is used; and 0, otherwise.

2.1.2. Parameters and Indices

The following hold.

i Hospital i, $i = 0, \ldots, n$, where $i = 0$ is a dummy hospital and corresponds to the blood transfusion depot of vehicles (vehicles' home).

$n + 1$ is total number of hospitals (including the depot).

a_i is arrival time to hospital i, where $a_0 = 0$.

F_p is total cumulative travel time for route p (permutation p).

c_v is the cost of using vehicle v, $v = 1, \ldots, m$.

UT is the upper bound on the total travel time of each vehicle (since we have m vehicles, thus we have m routes).

t_{ij} is a parameter which states the total travel time from hospital i to hospital j.

3. Problem Formulation

This section describes one of the two major contributions of this study by proposing an exact solution approach for the problem. The idea is to decompose the problem into subproblems using set covering formulations, and then to solve optimally each subproblem using TRP formulations. Thus, at first hospitals are assigned to vehicles (route construction), and then visiting order of hospitals inside each route is optimized. Computational complexity of the problem restricts developing a single combined mathematical programming formulation which assigns hospitals to vehicles while minimizes total waiting time inside each route.

3.1. Constructing Routes: A Location Covering Model

The assignment of hospitals to vehicles can be accomplished using the set covering formulations. One of such formulations has appeared in [20] where the goal is to locate a least-cost set of facilities such that each customer (hospital) can be reached within a maximum allowed travel time from the closest facility (blood transfusion depot). To apply this formulation to construct blood distribution routes, however, small modifications are required. These modifications are performed by imposing a set of constraints inspired from operational limitations and problem's nature (see constraints (3.4) below), and several changes into the model's parameters. Furthermore, as all vehicles leave from a central blood transfusion depot, the concept of closest facility as appeared in the location covering formulation is not applicable here. However, applying the constraints (3.4) coupled with the location covering model provided in [20] (see Model 1 below), ensures assignment of a set of hospitals to each vehicle where minimum travel time is satisfied.

Model 1

$$\text{Minimize } z = \sum_{v=1}^{m} c_v \gamma_v + \sum_{i=0}^{n} \sum_{v=1}^{m} t_{0i} \delta_{vi} \tag{3.1}$$

subject to

$$\sum_{v=1}^{m} \delta_{vi} \geq 1, \quad i = 0, \ldots, n, \tag{3.2}$$

$$\sum_{i=0}^{n} \delta_{vi} \leq |n+1| \gamma_v, \quad v = 1, \ldots, m, \tag{3.3}$$

$$\sum_{i=0}^{n} t_{0i} \delta_{vi} \leq UT, \quad v = 1, \ldots, m, \tag{3.4}$$

$$\delta_{vi}, \gamma_v \in \{0, 1\}, \quad i = 0, \ldots, n, \; v = 1, \ldots, m. \tag{3.5}$$

Constraints (3.2) guarantee that all hospitals are visited, while constraints (3.3) ensure that if vehicle v is not set up, then no hospital can be covered by it. Constraints (3.4) ensure that

total travel time of each route associated with each vehicle satisfies operational limitations. Constraints (3.5) state that all δ_{vi} and γ_v are binary.

3.2. Visiting Hospitals: A Traveling Repairman Model

After assigning hospitals to vehicles and constructing routes, optimal permutation of visiting hospitals with respect to minimizing total waiting time (equivalently total cumulative traveling time) of hospitals should be determined. This section describes a novel mixed-integer programming formulation for this purpose (Model Flow-SP). The proposed formulation is strong enough and allows optimal solutions for instances of up to 25 hospitals in a single route, and very close to optimality for instances with 30 hospitals.

Model Flow-SP

$$\text{Minimize } z = \sum_{i=0}^{n}\sum_{j=0}^{n} t_{ij} \cdot x_{ij} \tag{3.6}$$

subject to

$$\sum_{i=0}^{n} y_{ij} = 1, \quad j = 1, \ldots, n, \tag{3.7}$$

$$\sum_{j=0}^{n} y_{ij} = 1, \quad i = 1, \ldots, n, \tag{3.8}$$

$$\sum_{i=1}^{n} y_{i0} = 1, \tag{3.9}$$

$$\sum_{i=1}^{n} y_{0j} = 1, \tag{3.10}$$

$$\sum_{j=0}^{n} x_{0j} = n + 1, \tag{3.11}$$

$$\sum_{j=0}^{n} x_{j0} = 1, \tag{3.12}$$

$$\sum_{i=0}^{n} x_{i0} - \sum_{j=0}^{n} x_{0j} = 1 - (n + 1), \tag{3.13}$$

$$\sum_{i=0}^{n} x_{ik} - \sum_{j=0}^{n} x_{kj} = 1, \quad k = 1, \ldots, n, \tag{3.14}$$

$$\sum_{i=0}^{n}\sum_{j=0}^{n}x_{ij}=\sum_{s=1}^{n+1}s, \tag{3.15}$$

$$x_{ij}\le (n+1)\cdot y_{ij},\quad i,j=0,\dots,n, \tag{3.16}$$

$$y_{ij}\in\{0,1\},\quad i,j=0,\dots,n, \tag{3.17}$$

$$x_{ij}\ge 0,\quad i,j=0,\dots,n, \tag{3.18}$$

$$\sum_{u=0}^{n}p_{uj}=1,\quad j=0,\dots,n, \tag{3.19}$$

$$\sum_{j=0}^{n}p_{uj}=1,\quad u=0,\dots,n, \tag{3.20}$$

$$p_{00}=1, \tag{3.21}$$

$$\sum_{j=0}^{n}p_{nj}=1, \tag{3.22}$$

$$\sum_{k=0}^{n}x_{jk}-(n+1-(i+1))\cdot p_{ij}\ge 0,\quad i,j=0,\dots,n, \tag{3.23}$$

$$\sum_{j=0}^{n}y_{ij}-\sum_{u=0}^{n}p_{ui}\ge 0,\quad i=0,\dots,n, \tag{3.24}$$

$$p_{ui}\in\{0,1\},\quad u,i=0,\dots,n. \tag{3.25}$$

Constraints (3.7) and (3.8) are the assignment problem constraints that ensure every hospital is visited exactly once. Constraints (3.9) and (3.10) ensure that the vehicle starts from depot and finishes at depot. Redundant constraints (3.11) and (3.12) together with constraints (3.13), (3.14), and (3.15) define a network flow problem to remove subtours, that is, ensuring feasibility of solutions. Constraints (3.19) to (3.25) are scheduling constraints working according to the visiting order of the hospitals. Constraints (3.19) and (3.20) ensure that there is just one hospital visited in position i and vice verse. Constraints (3.21) and (3.22) ensure that the vehicle starts from depot and goes back to depot at the end of its service. Constraint (3.16), and constraints (3.23) and (3.24), links variables x_{ij} to variables y_{ij}, and variables y_{ij} to variables p_{ui}, respectively. The fact that all x_{ij}, p_{ui}, and y_{ij} variables are binary and nonnegative, respectively, is assured by constraints (3.17), (3.25), and (3.18).

Table 2 reports the dimensions of four mathematical models developed for TRP, namely Model of Fischetti et al. [15], Model of Eijl [16], Model of Mendez-Diaz et al. [18], and that of Salehipour and Sepehri [14] (also Model Flow-SP).

Apart from Table 2, another comparison regarding each model's performance is reported in Table 3. In this table, 50 problems in 10 different sizes ranging from 10 nodes to 50 nodes were solved. In each size, 5 instances were considered, where nodes' coordinates were randomly generated, and Euclidean costs were calculated and rounded down to the nearest integer. For each size, mean, minimum, and maximum computational time in seconds (over the five instances), and mean, minimum, and maximum gap in percent were reported. Note

Table 2: Total number of decision variables and constraints of models proposed for TRP.

	Models				
	1	Flow-SP	Fischetti et al. 1993 [15]	Eijl, 1995 [16]	Mendez-Diaz et al. 2008 [18]
Number of variables	$n^2 + m$	$3n^2$	$2n^2$	$2n^2 - n$	$n^3 - n^2 + n$
Number of constraints	$n + 2m$	$2n^2 + 6n + 5$	$n^2 + 3n + 1$	$n^2 + n - 1$	$\frac{n(n-1)(8n+5)}{6} + 1$

n: Total number of nodes.
m: Total number of vehicles.

Table 3: Performance of the four models proposed for TRP over randomly generated instances.

Size		Flow-SP		Fischetti et al. 1993 [15]		Eijl, 1995 [16]		Mendez-Diaz, 2008 [18]	
		Time	GAP	Time	GAP	Time	GAP	Time	GAP
	Mean	14.37	0	325.28	0	661.51	0	34.65	0
10	Min	9.13	0	106.38	0	186.84	0	20.75	0
	Max	17.73	0	890.12	0	1667.9	0	29.11	0
	Mean	105.79	0	3507.02	19.53	3600	60.02	426.36	0
15	Min	23.79	0	3135.12	0	3600	49.76	112.84	0
	Max	182.34	0	3600	41.02	3600	65.95	719.04	0
	Mean	605.11	0	3600	43.47	3600	63.51	2639.94	9.64
18	Min	120.09	0	3600	35.98	3600	56.76	902.44	0
	Max	1261.23	0	3600	51.88	3600	70.22	3600	12.57
	Mean	208.59	0	3600	48.77	3600	68.96	3133.59	13.48
20	Min	59.27	0	3600	39.81	3600	63.12	2753.04	0
	Max	393.13	0	3600	54.83	3600	72.77	3600	17.84
	Mean	2718.98	3.84	3600	67.3	3600	82.54	3600	18.89
25	Min	1382.57	0	3600	63.46	3600	79.37	3600	14.35
	Max	3600	7.39	3600	70.45	3600	87.2	3600	20.64
	Mean	3600	14.16	3600	70.75	3600	83.46	3600	23.83
30	Min	3600	7.33	3600	65.58	3600	81.02	3600	19.72
	Max	3600	19.93	3600	76.17	3600	84.6	3600	26.38
	Mean	3600	19.82	3600	80.56	3600	86.74	3600	29.28
35	Min	3600	11.47	3600	78.9	3600	85.48	3600	26.63
	Max	3600	30.37	3600	82.82	3600	87.86	3600	35.11
	Mean	3600	20.68	3600	84.51	3600	87.48	3600	33.68
40	Min	3600	14.18	3600	79.25	3600	86.76	3600	27.82
	Max	3600	22.4	3600	87.62	3600	88.27	3600	38.92
	Mean	3600	23.46	3600	89.24	3600	89.09	3600	38.98
45	Min	3600	15.61	3600	86.28	3600	88.02	3600	34.79
	Max	3600	29.02	3600	91.03	3600	89.8	3600	39.03
	Mean	3600	26.37	3600	88.28	3600	91.52	3600	44.13
50	Min	3600	18.11	3600	85.84	3600	—	3600	37.04
	Max	3600	33.33	3600	89.95	3600	92.26	3600	48.77

that the limit on the computational time is set to be 1 hour. The commercial solver Cplex 9.0 from ILOG was implemented.

According to the table, the best results were reported by Model Flow-SP developed by Salehipour and Sepehri [14]. The performance of Model of Mendez-Diaz et al. is quite promising. Even ignoring the computational time, and gaps, Model of Mendez-Diaz et al. is not still comparable to Model Flow-SP, as the largest size solved by Model Flow-SP has 25 nodes.

4. Heuristic Model

Despite a strong mixed-integer formulation developed in Section 3.2, the problem of finding optimal permutation hospitals in a route (traveling repairman problem) is still very difficult to solve. To derive near-optimal solutions for medium and larger instances, an efficient hybrid heuristic algorithm is developed in this section. The basic idea is to control the improvement procedure accomplished by the variable neighborhood search (VNS) algorithm using the simulated annealing (SA) algorithm. The basis of the heuristics developed in the following sections require finding the closest, second-closest, and so forth hospital to any given hospital. We, therefore, preprocess the travel time data and maintain in memory a proximity matrix that contains one row per hospital. The ith row of the matrix contains all hospitals (except for hospital i) in order of proximity to hospital i. Followings are through details of this hybrid algorithm.

4.1. The Construction Algorithm

The construction heuristic is the greedy randomized adaptive search procedure (GRASP) developed by Feo and Resende [21, 22]. The motivation for using the GRASP for the TRP is the observation that the first few hospitals in a route are more important than the later ones. This immediately follows from (2.4), from which it is clear that the ith distance in the solution is multiplied by a factor $(n+1)-i+1$. Hence, it is expected that a greedy algorithm performs well. A completely greedy algorithm on the other hand can be expected to miss some interesting opportunities.

As the most distinguishing characteristic of the GRASP where greediness is combined with randomness in the construction phase, a restricted candidate list (RCL) is built by selecting a subset of all elements (hospitals) in a greedy fashion. Assuming a minimization problem, the RCL contains the elements whose incorporation into the partially built solution would yield the smallest increase in the objective function value. From the RCL, an element is then selected at random, after which the RCL is updated to reflect the fact that a new element was added to the solution and is no longer available for selection. Selection of an element and update of the RCL are repeated until a complete solution has been built. From this solution, an improvement procedure starts until a local optimum is found. The size of the RCL, α, is a parameter of the GRASP algorithm that controls the balance between greediness and randomness. If α is small, the search is relatively greedy. If α is large, it is relatively random.

The RCL concept can be easily translated to the TRP, and efficiently implemented using the proximity matrix. Starting from 0, the RCL is filled with the α closest hospitals. From this RCL, a random hospital is chosen, and then the RCL is filled with the α hospitals closest to it, removing any hospitals that have already been selected. If less than α hospitals remain, then

$U \leftarrow \{0, v_1, v_2, \ldots, v_n\}$;
$v_c \leftarrow 0$;
Repeat
  Create RCL with α vertices $v_i \in U$ closest to v_c;
  Select random vertex $v_r \in$ RCL;
  $U \leftarrow U\{v_r\}$;
  $v_c \leftarrow v_r$
Until $U = \emptyset$;

Algorithm 1: Outline of the GRASP for the TRP.

the size of the RCL is decreased. Algorithm 1 outlines the GRASP construction algorithm for the TRP.

4.2. The Improvement Procedure: SA + VNS Algorithms

The improvement procedure includes the five local searches, acquired from the routing literature and modified for this problem, changed in a systematic way using the variable neighborhood search (VNS) meta-heuristic [23–26]. This VNS algorithm is itself controlled by the simulated annealing (SA) meta-heuristic to avoid being trapped in local optima.

The VNS meta-heuristic systematically explores different neighborhood structures. The main idea underlying the VNS is that a local optimum relative to a certain neighborhood structure is not necessarily a local optimum relative to another neighborhood structure. For this reason, escaping from a local optimum can be accomplished by changing the neighborhood structure. Although this is not required, many implementations of VNS use a sequence of nested neighborhoods, N_1 to $N_{k\,\max}$, in which each neighborhood in the sequence is a superset of its predecessor, that is, $N_k \subset N_{k+1}$. The neighborhoods used in our VNS do not possess this property. Furthermore, usually in the VNS algorithm larger neighborhoods are only examined when all smaller neighborhoods have been depleted, that is, when the current solution is a local optimum of all smaller neighborhoods. Another option is to use the neighborhoods in the order of their effectiveness with respect to the problem at hand, which can be established by a small pilot study. Algorithm 2 presents the pseudocode for a basic VNS.

Based on our previous experience, we set $k_{\max} = 5$ in the hybrid algorithm. These five neighborhoods were applied according to the order presented in Table 4.

Introduced by Kirkpatrick et al. [27], the SA algorithm is one of the well-known meta-heuristic algorithms where trapping in local optima is avoided by sometimes accepting a neighborhood move which worsens the value of the objective function. The acceptance or rejection of a move (even the worse moves) is determined by a sequence of random numbers, but with a controlled probability. The probability of accepting a move, causing an increase Δ in the objective function f, is called the acceptance function and is normally set to $e^{(-\Delta)/T}$, where T is a control parameter corresponding to the temperature in the analogy with physical annealing. This acceptance function implies that small increases in f are more likely to be accepted than large increases. When T is high, most moves will be accepted, but as T approaches zero most uphill moves will be rejected. Usually, the SA starts with a relatively high value of T to avoid being trapped in a local optimum too early. The algorithm proceeds

Input: Initial solution x;
Until the stopping criterion is met **do**
  Initialize: $k \leftarrow 1$;
  Until $k = k_{\max}$ **repeat**
    Shake: Generate $\acute{x}$ randomly from $N_k(x)$ and set it as the incumbent solution;
    Local search: Apply some local search method on $\acute{x}$; Denote with $\overline{x}$ the so obtained local optimum;
    If $\overline{x}$ is better than the x **then**
    $x \leftarrow \overline{x}$ and set $k \leftarrow 1$;
    Else
    $k \leftarrow k + 1$ (switch to another neighborhood);

Algorithm 2: Outline of the basic variable neighborhood search.

Input: Initial solution x
Select an initial temperature $T > 0$;
Repeat
    Local search: $\acute{x} \leftarrow \arg\min N(x)$;
    Calculate $\Delta = f(\acute{x}) - f(x)$;
    If $\Delta < 0$ then $x \leftarrow \acute{x}$;
    Elseif random(0,1) $< e^{-\Delta/T}$, then $x \leftarrow \acute{x}$;
    Else
    Reject $\acute{x}$;
    $T = T(c)$;
Until stopping condition is met;

Algorithm 3: The outline of simulated annealing.

Table 4: The neighborhoods and their applied order in the VNS improvement scheme.

k	Neighborhood
1	Swap-adjacent
2	Swap
3	Remove-insert
4	2-OPT
5	3-OPT

by attempting a certain number of neighborhood moves at each temperature (here, in a VNS algorithm), while the temperature gradually drops. We implemented the SA algorithm as a controlling scheme on local searches to avoid being trapped in local optima. The outline of the SA is brought in Algorithm 3, and the outline of the hybrid algorithm is shown in Algorithm 4.

4.3. Neighborhoods

Our search uses five neighborhood structures: swap-adjacent, swap, remove-insert, 2-opt, and 3-opt. The swap heuristic attempts to swap the positions of each pair of hospitals in the

Initial procedure GRASP, report the best found solution x;
Initial procedure SA + VNS;
Repeat
Local search: apply VNS procedure with the five neighborhoods, report the incumbent solution $\acute{x}$;
Calculate Δ;
If $\Delta < 0$ then $x \leftarrow \acute{x}$;
Elseif random(0,1) $< e^{-\Delta/T}$, then $x \leftarrow \acute{x}$;
Else Reject $\acute{x}$, and modify temperature;
Until stopping criterion is met;

Algorithm 4: The outline of hybrid algorithm.

permutation. The remove-insert heuristic examines randomly each hospital i in the solution and places it at the end of the permutation. The swap-adjacent heuristic, a subset of the swap heuristic, attempts to swap each pair of adjacent hospitals in the permutation. The 2-opt heuristic removes each pair of edges (partial routes between two hospitals) from the solution and reconnects the nodes (hospitals). The 3-opt heuristic removes each triplet of edges from the solution and reconnects the solution in such a way that the orientation of the solution parts is preserved.

4.4. Neighborhood Reduction

Several attempts are made to decrease the size of the neighborhoods to examine during the improvement phase. The implemented neighborhood reduction scheme is based on the observation that in a reasonably good solution, most improving moves will involve hospitals that appear in relative proximity to each other in this solution. For example, in a good solution it is unlikely that a swap will yield a better solution if it exchanges a hospital that appears close to the depot with one that appears far from it. We, therefore, introduce a proximity factor β to determine the maximum distance between hospitals that may be involved in a move. The factor β has a slightly different influence, depending on the move. The neighborhood reduction strategy is not used for the swap-adjacent and remove-insert heuristics.

5. Computational Results

In this Section, we provide detailed computational experiments of the mathematical formulation provided in Section 3, and the hybrid heuristic of Section 4. In this study, two datasets were considered, random datasets, and a real dataset. All computational experiments were carried out on a Pentium 4 PC with 2 GB of memory and 2.0 GHz of CPU. We employed solver Cplex 9.0 from ILOG. The hybrid heuristic was coded in C++.

5.1. Random Datasets

A set of 30 random instances were generated with sizes ranging of 50, 100, and 150 hospitals (for each size, 10 problems were generated). For each problem, travel time among hospitals

Table 5: The general details of the generated instances.

File name	Number of hospitals	(min, max)
TRP-S50-Rx	50	(0, 100)
TRP-S100-Rx	100	(0, 100)
TRP-S150-Rx	150	(0, 100)

Table 6: The best value of hybrid heuristic algorithm parameters.

Parameter	Description	Value
t_0	Initial temperature	10
c	Cooling rate ($0 < c < 1$)	0.01
iter	Maximum number of iterations in each temperature	∞
α	RCL size in GRASP construction algorithm	2

has been generated randomly using uniform distribution in $[1, 100]$. Table 5 shows the general details of the instances generated. In each problem, a maximum number of 10 blood distribution vehicles are considered, although, according to Model 1 not all the vehicles may be implemented in optimal solution. In all instance, we set UT (upper bound on the total travel time of each vehicle) to be 1500.

To tune the parameters of our hybrid algorithms, that is, the GRASP algorithm parameter, and the SA algorithm parameters, we refer the interested readers to [14, 28] where a comprehensive study regarding this tuning process for the TRP has been accomplished. The best values for these parameters are shown in Table 6.

Table 7 reports the complete computational results of both mixed-integer mathematical programming and the hybrid heuristic. The forth column of the table, "OFV of Model 1" shows the objective function value associated with Model 1, that is, the minimum cost associated with the constructed routes. The column "OFV of Model Flow-SP" refers to the computational results of the Model Flow-SP, that is, optimal permutation of visiting hospitals in each route. All reported computational times are in seconds, and we set the upper bound on the CPU time to be 500 seconds. For each instance, under the "OFV of Model Flow-SP," we have shown average, minimum, and maximum gap and time associated with each of the solutions methods (mixed-integer programming and hybrid heuristic). All reported gaps for the exact solution are those from Cplex solver. This gap is calculated from the lower bound derived by the solver over the best objective function found by the solver (obviously, for the optimal solution this gap is 0%). In fact, Cplex solver proves optimality of the solution by using this gap. Thus, this gap can be a measure of strength of the formulation as the convergence rate of this gap towards 0% in a reasonable amount of computational time is of importance. It is worth mentioning that Model Flow-SP yielded an average gap of 14.63% for instances with 50 hospitals, and around 18%, for instances with 100 and 150 hospitals, respectively. The average of "Min. Gap" is even much lower as it is below 8%.

Table 7: The computational results of the two solution approaches on the generated instances.

Number of hospitals	Instance name	Optimal number of vehicles	OFV of model 1	OFV of Model Flow-SP											
				Exact						Hybrid heuristic					
				Ave. gap	Ave. time	Min. gap	Min. time	Max. gap	Max. time	Ave. gap	Ave. time	Min. gap	Min. time	Max. gap	Max. time
50	TRP-S50-R1	2	4607	17.45	283.23	0	66.45	34.9	500	7.05	3.45	0	0.12	14.59	8.95
	TRP-S50-R2	2	4268	18.89	286.11	0	72.22	37.78	500	6.43	2.99	0	1.23	7.34	12.41
	TRP-S50-R3	2	4029	9.98	500	3.71	500	16.24	500	7.12	3.16	3.71	0.67	11.04	7.56
	TRP-S50-R4	2	4627	10.06	322.95	0	145.89	20.11	500	11.01	7.35	0	0.19	2.10	8.72
	TRP-S50-R5	3	6198	2.73	195.10	0	10.92	0	74.39	2.74	4.23	0	1.23	0	9.99
	TRP-S50-R6	2	4347	18.50	272.72	0	45.44	37	500	11.57	2.17	0	1.02	14.12	13.24
	TRP-S50-R7	2	4029	24.87	303.61	0	107.22	49.74	500	4.11	3.40	0	0.76	23.19	12.76
	TRP-S50-R8	2	4983	19.60	500	14.5	500	24.69	500	7.65	9.11	12.30	0.33	4.99	9.06
	TRP-S50-R9	3	6255	5.29	333.56	0	0.69	10.6	500	4.11	1.76	0	2.45	0.96	11.63
	TRP-S50-R10	2	4689	18.97	500	6.51	500	31.43	500	13.07	5.09	5.54	1.89	11.51	7.66
Average				14.63	349.73	2.47	194.88	26.25	457.44	7.49	4.27	2.16	0.99	8.98	10.20
100	TRP-S100-R1	4	9301	13.39	416.56	0	166.23	19.29	500	10.33	33.34	0	16.23	17.9	103.13
	TRP-S100-R2	4	9086	18.73	399.41	0	97.64	33.2	500	8.04	42.97	0	22.67	16.21	79.17
	TRP-S100-R3	4	9495	16.38	345.28	0	171.30	36.52	500	2.17	35.78	0	18.91	14.33	145.22
	TRP-S100-R4	5	11773	3.51	476.09	0	500	9.11	500	5.15	56.03	0	27.33	3.65	183.93
	TRP-S100-R5	4	8889	35.50	500	31.52	500	38.16	500	13.49	43.11	3.24	29.01	20.71	95.18
	TRP-S100-R6	4	9680	13.03	500	3.15	500	32.04	500	3.03	37.98	3.13	19.76	9.47	105.62
	TRP-S100-R7	4	8860	11.35	500	8.71	500	13.02	500	9.67	49.67	7.62	32.19	3.92	186.91
	TRP-S100-R8	3	6914	19.00	500	5.7	500	32.13	500	15.01	51.12	0.99	33.56	27.49	234.77
	TRP-S100-R9	3	7213	22.69	500	1.58	500	40.44	500	20.79	44.46	0.76	39.66	18.98	218.39
	TRP-S100-R10	3	8111	26.71	500	13.23	500	34.65	500	22.07	39.57	3.41	16.96	24.11	399.89
Average				18.03	463.73	6.39	393.52	28.86	500	10.98	43.40	1.92	25.63	15.68	175.22
150	TRP-S150-R1	5	12055	20.57	500	7.08	500	32.70	500	14.22	99.03	7.08	76.44	13.34	203.52
	TRP-S150-R2	6	14885	8.27	423.29	0	219.75	41.68	500	5.21	101.11	0	93.23	1.52	189.01
	TRP-S150-R3	6	13823	17.23	272.39	0	25.53	58.78	500	11.13	84.56	0	108.02	21.38	500
	TRP-S150-R4	6	14282	12.28	345.52	0	36.50	28.92	500	6.08	89.95	0	66.88	7.12	123.95
	TRP-S150-R5	4	9626	29.26	500	20.02	500	51.40	500	12.61	143.10	12.4	87.50	19.05	295.91
	TRP-S150-R6	5	12113	24.16	500	7.65	500	38.02	500	12.462	105.02	7.65	99.10	14.34	165.55
	TRP-S150-R7	4	9976	41.05	500	34.19	500	62.04	500	2.78	146.31	13.32	109.11	11.44	293.60
	TRP-S150-R8	7	16982	7.17	348.14	0	57.06	32.26	500	3.99	223.90	0	120.32	7.06	473.92
	TRP-S150-R9	7	16763	8.96	361.39	0	31.44	24.67	500	4.89	117.06	0	59.34	15.61	308.86
	TRP-S150-R10	5	12366	18.01	500	2.88	500	29.91	500	9.17	389.89	2.88	140.08	10.31	500
Average				18.70	425.07	7.18	287.03	40.04	500	9.04	149.99	4.33	96.00	13.42	305.43

According to the table, the hybrid heuristic exhibited much better performance, both in CPU time and in reported gap, especially when the size of problem increased. This reveals the strength of the developed heuristic for the problem. Here, the reported gap was calculated from the best lower bound found by Cplex solver over the best objective function found by the heuristic. Furthermore, maintaining the same quality, even in many cases with a lower gap, the hybrid heuristic rarely reached the upper bound on the computational time while for instances with 100 and 150 hospitals, the mixed-integer programming formulation reached this upper bound in several cases. Note that according to the table, Model Flow-SP is very strong as for instances with 50 hospitals, its performance is well enough and its average gap is below 15%. But due to the complexity of the problem, still the problems of 50 hospitals are not solvable in a reasonable time.

The developed hybrid-heuristic performed quite well, and tracking back its performance, we observed that implementing the GRASP has been a very good initialization. Also powerful neighborhoods in the VNS meta-heuristic coupled with the SA algorithm has magnified this performance which resulted in near optimal solutions very quickly.

5.2. Real Dataset: A Case Study in Tehran, Iran

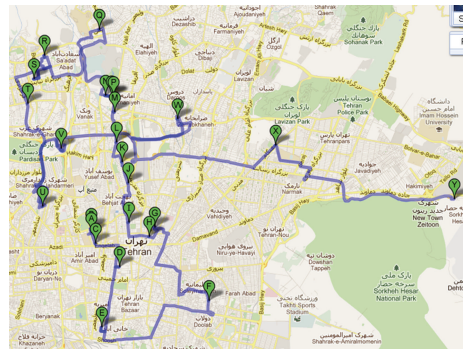
Here, the method of Section 3 is applied to a real blood distribution problem, experienced by the authors. In this study, 24 hospitals were considered in the city of Tehran, capital of Iran. Generally, in Tehran a single blood transfusion depot supplies the demands of all hospitals, in a way that each hospital sends its vehicle to deliver the demand. The data of this study includes the travelling distances to reach each hospital, the daily demand (blood products) of each hospital, the maximum travelling distance allowed for vehicles, and the vehicles' capacity. Note that when considering the travelling cost of each vehicle, we take the travelling distance instead, usually by taking into account the approximate travelling cost per kilometer.

The travelling distances, in kilometer, are given in the appendix. For the sake of simplicity, we assumed the table is symmetric. The distances were extracted by Google Map service of Google. Typically, this service offers several routes between every two points, among which we chose the shortest one.

We were restricted to provide the daily demands of hospitals to public domain. As for vehicles' capacity, we set 600 units of blood products per vehicle, although different vehicles may hold different capacities. On a standard vehicle, it is common in practice that the maximum travelling distance allowed for vehicle is reached before the capacity. Hence, from practical viewpoint, the capacity may less affect the final decision. It is worth mentioning that the maximum travelling distance allowed differs when different numbers of vehicles are hired. In fact, as the demands of all hospitals should be met, this upper bound on travelling distance should be set such that all demands are met. The maximum travelling distance allowed when hiring different numbers of vehicles are shown in Table 8. Except the case when there is only one vehicle, the data of the table reflect the concern of several hospitals.

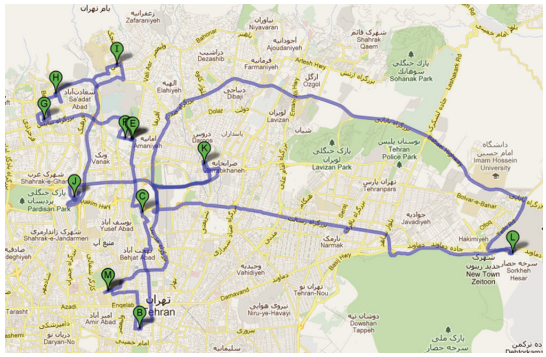
According to the number of hospitals considered in this study, it is expected that at most three vehicles could satisfy the demands of all hospitals. The routes, constructed by hiring different numbers of vehicles, are depicted in Figures 3, 4, and 5. Table 9 elaborates more on the results.

In the table, the first column shows the number of vehicles hired for supplying demands while the second column shows the objective function of subproblem of Section 3.1. The remaining columns which are same for each vehicle are interpreted as objective function

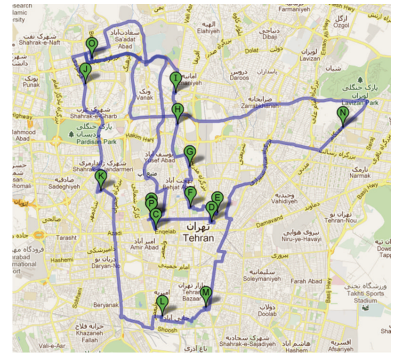


Delivery order: {0, 14, 17, 4, 16, 21, 19, 9, 13,
15, 7, 12, 22, 3, 8, 24, 1, 2, 6, 11, 20, 23, 10, 18, 5, 0}

Figure 3: The best found delivery order when hiring one vehicle. Note that from location Y, the vehicle must return to its origin, A.



(a) Optimal delivery order of vehicle 1: {0,4,7,8,24,3,6,2,1,23, 10,5,0}



(b) Optimal delivery order of vehicle 2: {0, 14,17,9,19,13,15,12,22,11,20,16,21,18,6,0}

Figure 4: The optimal delivery orders when hiring two vehicles, (a) vehicle 1, and (b) vehicle 2.

Table 8: The limit on travelling distances for vehicles.

Number of vehicle(s)	Maximum allowed travelling distance
1	220 km
2	120 km
3	80 km

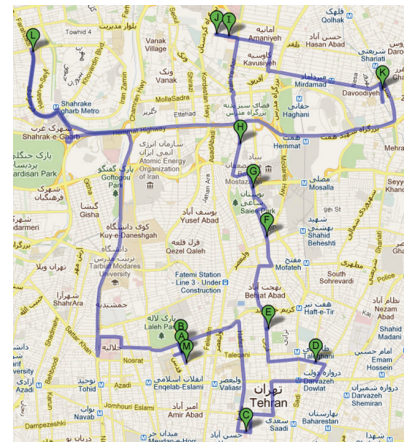
of the TRP (column OFV), total distances travelled by vehicle in kilometers (column Trv. Dis.), and total travelling time of vehicle ignoring the delivery time (column Trv. Time).

5.3. Current Situation

Currently, in Tehran, each hospital maintains its own vehicle to deliver its demand. However, an investigation performed by the authors revealed that the vehicle is not properly suited for

(a) Optimal delivery order of vehicle 1: {0,2,6,24,23,5,0}

(b) Optimal delivery order of vehicle 2: {0,17,16,21, 19,18,22,24,20,0}



(c) Optimal delivery order of vehicle 3: {0, 14,4,9,13,15,7,12,8,3,10,11,0}

Figure 5: The optimal delivery orders when hiring three vehicles, (a) vehicle 1, (b) vehicle 2, and (c) vehicle 3.

Table 9: The experimental result of solving the blood product distribution problem for 24 hospitals.

Number of vehicle (route)	Vehicle cost*	Vehicle no. 1			Vehicle no. 2			Vehicle no. 3		
		OFV	Trv Dis. (km)	Trv Time (hr:min)	OFV	Trv Dis. (km)	Trv Time (hr:min)	OFV	Trv Dis. (km)	Trv Time (hr:min)
1	304.75	922.314**	135.4	03:42	—	—	—	—	—	—
2	404.75	386.36	103	02:22	331.15	92.8	02:40	—	—	—
3	504.75	192.10	89.9	02:24	235.7	63.2	01:53	206.5	45.7	01:25

*The cost of each vehicle was assumed to be 100.
**The Cplex was manually stopped after one hour of running; the reported gap was 10.7%.

this purpose, which may affect the quality of the service. The disadvantages of this scenario (Scenario 2) are the follows.

(i) The central blood transfusion of Tehran which distributes blood products to all hospitals is located in a highly crowded part of the city, where the traffic is limited. Furthermore, only a few hospitals hire vehicles allocated for blood products delivery, which as emergency vehicles, are also allowed to freely enter this area. For hospitals not hiring such vehicles, the most appropriate vehicle would be taxis.

Although, taxis are allowed to enter this area, they are not considered as emergency vehicles. Thus, in case of an accident, traffic jam, or any other similar circumstances, they will end up in delay, even to several hours, as an unofficial report by one hospital states.

(ii) When hospitals hire taxis, obviously, the vehicles are not specifically designed for the purpose of blood products distribution. Furthermore, the driver has not been trained regarding any special conditions of keeping and transporting blood products (e.g., appropriate temperature).

(iii) The human errors could make the situation much worse. For instance, the driver cannot distinguish between different blood products, which probably results in wrong delivery.

Apart from these disadvantages, this scenario incurs higher costs compared to case where specific vehicles are sent to hospitals from blood transfusion depot (Scenario 1, see also Section 2). For this purpose, Table 10 illustrates the associated costs.

The costs of the table are calculated by twice of the travelling distances in kilometers plus 10% of this cost, as for delays in delivery. As the table shows, the total cost is 450.45 (kilometers). Interpreting this cost as the total travelling times of all vehicles, Table 10 shows superiority of Scenario 1. Note that here the cost structure has also changed. In Scenario 1 the blood transfusion depot covers the cost of delivery, whereas in Scenario 2 hospitals cover the cost, such that each hospital covers for its own. Furthermore, Scenario 1 improves distribution of blood products in several ways.

(i) Total flow time and transportation costs are minimized, as the blood products distribution problem is modeled as Traveling Repairman Problem where the objective is to minimize the total waiting time of the service to be delivered.

(ii) Delays in transportation are minimized. This is attained as special-purpose vehicles are allocated for distributing of blood products, which are considered as emergency vehicles and can overpass traffic limitations.

(iii) The highest level of quality of service is maintained compared to Scenario 2. This is because vehicles are highly equipped and staffs are well trained. Obviously, this reduces wrong delivery and improves on time delivery, and most importantly "as ordered" delivery.

According to results (Tables 9 and 10), lower travelling time (equivalently travelling distance) is reached when employing Scenario 1. Thus, it is not very far if we state Scenario 1 outperforms Scenario 2. Finally, considering both technical difficulties, which are much more important than costs (as the problem concerns human lives), and also costs, Scenario 1 is superior to Scenario 2. This is indeed very important, because on a real scale problem, like city of Tehran, improvements are substantial. (Considering both private and public hospitals, currently Tehran holds around 150 hospitals. Also note that we have solved random instances of size 150 hospitals in reasonable amount of time, see Table 7.)

It is worthy to add that currently authors are negotiating with Iran Ministry of Health to design a distribution network of blood products in Tehran. This not only requires software infrastructure, but also hardware ones, most importantly, well-equipped vehicles and well-trained staffs. Here, we believe solving the problem should not be an issue, although the access to the appropriate data is.

Table 10: Cost of each hospital maintaining its demands.

Hospital	Cost (km)
Taleghani	22.66
Modarres	20.24
Motahhari	15.18
Sherkat-e-Naft	8.14
Dr. Lavasani	53.46
Erfan	31.9
Kasra	13.42
Khatam-ol-Anbiya	21.12
Arad	8.58
Mofid	28.38
Atiyeh	30.36
Dey	14.96
Apadana	6.6
Ariya	1.87
Asiya	10.56
Ashrafi Esfahani	15.62
Alborz	2.64
Al-Ghadir	31.46
Iranshahr	8.58
Rasol Akram	15.4
Akbar Abadi	25.08
Moheb	17.6
Milad	22.44
Vali-e-Asr	24.2

6. Conclusion

In this paper, we presented a new application of operations research in healthcare. The problem is to find optimal routes when distributing blood products among hospitals such that the total waiting time of hospitals requiring them are minimized. We studied a real blood distribution problem where 24 hospitals were to be served by a maximum of three vehicles. The proposed solution approaches are a mixed-integer programming formulation and a hybrid heuristic. Computational experiments showed the efficiency of the exact approach in finding optimal solutions for the real case. For the purpose of providing high-quality solutions for much larger instances, several random instances containing up to 150 hospitals and 10 vehicles were generated. The performance of the developed hybrid heuristic on these instances is promising, and near-optimal solutions were reported in a short time (almost 3 minutes on average for these problems). The proposed formulation and solution approach provide a basis for many other home healthcare problems and applications. Currently, the authors are working on these problems and applications.

Appendix

For more details see Table 11.

Table 11: Symmetric travelling distance matrix for the 24 hospital example.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
0	0	10.3	9.2	6.9	3.7	24.3	14.5	6.1	9.6	3.9	12.9	13.8	6.8	3	0.85	4.8	7.1	1.2	14.3	3.9	7	11.4	8	10.2	11
1		0	6.8	7.8	16.9	31	7.2	10.1	7.6	14	10.2	7.3	9.3	13.5	14.4	10.8	19.5	14.8	17.4	13.7	12.3	21.5	6.6	8.6	6.3
2			0	9.9	15.1	34.4	2	13.4	9	17.4	11.8	3.9	9.5	16.8	12.5	14.2	17.6	13	20.7	17	13.4	21.6	6.7	11.5	6.4
3				0	12.4	23.6	6.4	4.8	0.35	10.3	5.1	8	4.1	9.7	9.8	5.8	18.9	10.6	13.6	10.1	11.8	14.5	1.4	5.9	0.4
4					0	20	17.7	6.6	12	3.3	12.3	15.2	7.4	2.9	3.7	5.3	3.9	2.9	14.8	3.8	8.4	4.7	9	11.6	12
5						0	31.9	28.3	25.7	16.8	23.7	32.8	27.5	31.7	33.2	29	27.5	19.5	13.4	16.9	37	21.2	20.9	24.2	23.2
6							0	14.6	10.2	18.6	13	3.4	10.1	18	13.1	15.3	18.2	13.6	21.9	18.2	14	22.8	7.1	9	6.8
7								0	4.1	5	5.5	9.1	1.4	4.4	5	1.1	9.7	5.6	8.9	4.7	11.2	9	2.8	5.8	4.3
8									0	9.9	4.8	8.3	3.8	9.4	10.1	5.5	19.2	10.9	13.3	9.6	12.1	13.9	1	6.2	0.074
9										0	7.6	12.7	6.2	1.8	3.9	4.1	6.7	3.1	10.2	0.95	8.7	5.9	10	10.8	9.9
10											0	10.8	6.2	9.1	10.7	6.4	15.4	11.2	13	9.3	12.9	13.9	4.2	9.2	4.7
11												0	7.2	12.8	11.6	10.1	16.7	12	16.7	13	7.3	17.4	8.7	7.9	8.3
12													0	7.1	8.5	2.6	11.2	9.2	9.6	7.4	10	11.7	2.2	4.6	3.7
13														0	2.6	2.4	7.3	3.1	12	2	8.9	6.3	9.5	10.2	9.3
14															0	3.9	7.5	1	13.4	4	6.9	8.3	8.7	9	10.2
15																0	8.6	4.5	9.6	3.6	11.9	8.2	3.9	7.6	5.4
16																	0	5.1	18.3	7.8	10.3	3	19.6	14.1	19.3
17																		0	14.2	3.7	5.8	8	9.4	9.5	11
18																			0	9.8	19.7	13.8	10.9	14.1	13.2
19																				0	9.4	5.4	9.7	10.5	9.5
20																					0	12.7	12.5	7	12.1
21																						0	14	14.8	13.9
22																							0	5.6	2.3
23																								0	9
24																									0

References

[1] D. Delen, M. Erraguntla, R. J. Mayer, and C. N. Wu, "Better management of blood supply-chain with GIS-based analytics," *Annals of Operations Research*, vol. 185, no. 1, pp. 181–193, 2011.

[2] J. Chapman and J. MacPherson, "Unlocking the essentials of effective blood inventory management," *Transfusion*, vol. 47, no. 2, pp. 190–196, 2007.

[3] E. Brodheim and G. P. Prastacos, "The Long Island blood distribution system as a prototype for regional blood management," *Interfaces*, vol. 9, no. 5, pp. 3–20, 1979.

[4] G. P. Prastacos, "Blood inventory management: an overview of theory and practice," *Management Science*, vol. 30, no. 7, pp. 777–800, 1984.

[5] C. Sapountzis, "Allocating blood to hospitals from a central blood bank," *European Journal of Operational Research*, vol. 16, no. 2, pp. 157–162, 1984.

[6] S. L. Sime, "Strengthening the service continuum between transfusion providers and suppliers: enhancing the blood services network," *Transfusion*, vol. 45, no. 4, pp. 206–223, 2005.

[7] J. S. Rytila and K. M. Spens, "Using simulation to increase efficiency in blood supply chains," *Management Research News*, vol. 29, no. 12, pp. 801–819, 2006.

[8] K. Katsaliaki, "Cost-effective practices in the blood service sector," *Health Policy*, vol. 86, no. 2-3, pp. 276–287, 2008.

[9] K. Katsaliaki and S. C. Brailsford, "Using simulation to improve the blood supply chain," *Journal of the Operational Research Society*, vol. 58, no. 2, pp. 219–227, 2007.

[10] V. Hemmelmayr, K. F. Doerner, R. F. Hartl, and M. W. P. Savelsbergh, "Delivery strategies for blood products supplies," *OR Spectrum*, vol. 31, no. 4, pp. 707–725, 2009.

[11] N. Rego and J. P. Sousa, "Supporting the definition of strategies for the configuration of health care supply chains," in *Logistik Management Systeme, Methoden, Integration*, S. Voß, J. Pahl, and S. Schwarze, Eds., Springer, 2009.

[12] S. Sahni and T. Gonzalez, "*P*-complete approximation problems," *Journal of the Association for Computing Machinery*, vol. 23, no. 3, pp. 555–565, 1976.

[13] L. Bianco, A. Mingozzi, and S. Ricciardelli, "The traveling salesman problem with cumulative costs," *Networks*, vol. 23, no. 2, pp. 81–91, 1993.

[14] A. Salehipour and M. M. Sepehri, "A new mixed-integer programming formulation for locating workstations in tandem automated guided vehicle systems," Working paper, Department of Industrial Engineering, Tarbiat Modares University, Tehran, Iran, 2010.

[15] M. Fischetti, G. Laporte, and S. Martello, "The delivery man problem and cumulative matroids," *Operations Research*, vol. 41, no. 6, pp. 1055–1064, 1993.

[16] E. Eijl, "A polyhedral approach to the delivery man problem," Memorandum COSOT 95, Eindhoven University of Technology, 1995.

[17] J. F. M. Sarubbi and H. P. L. Luna, "A flow formulation for the minimum latency problem," in *Proceedings of the International Network Optimization Conference (INOC '07)*, Spa, Belgium, April 2007.

[18] I. Mendez-Diaz, P. Zabala, and A. Lucena, "A new formulation for the traveling deliveryman problem," *Discrete Applied Mathematics*, vol. 156, no. 17, pp. 3223–3237, 2008.

[19] D. Simchi-Levi and O. Berman, "Minimizing the total flow time of n jobs on a network," *IIE Transactions*, vol. 23, no. 3, pp. 236–244, 1991.

[20] G. Ghiani, G. Laporte, and R. Musmanno, *Introduction to Logistics Systems Planning and Control*, Wiley, 2004.

[21] T. A. Feo and M. G. C. Resende, "A probabilistic heuristic for a computationally difficult set covering problem," *Operations Research Letters*, vol. 8, no. 2, pp. 67–71, 1989.

[22] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedures," *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, 1995.

[23] P. Hansen and N. Mladenovic, "Variable neighborhood search for the *p*-median," *Location Science*, vol. 5, pp. 207–226, 1997.

[24] P. Hansen and N. Mladenovic, "An introduction to variable neighborhood search," in *Metaheuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I. Osman, and C. Roucairol, Eds., pp. 433–458, Kluwer Academic, Boston, Mass, USA, 1999.

[25] P. Hansen and N. Mladenovic, "Industrial applications of the variable neighborhood search metaheuristic," in *Decisions and Control in Management Science*, pp. 261–274, Kluwer Academic, Boston, Mass, USA, 2001.

[26] P. Hansen and N. Mladenović, "Variable neighborhood search: principles and applications," *European Journal of Operational Research*, vol. 130, no. 3, pp. 449–467, 2001.

[27] S. Kirkpatrick, J. C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[28] A. Salehipour, P. Goos, K. Sorensen, and O. Braysy, "The traveling repairman problem," in *Proceedings of the 21st Annual Conference of the Belgian Operational Research Society (ORBEL '83)*, Luxembourg, January 1983.