Optimization of blood sample collection with timing and quality constraints

Amir Elalouf, Dmitry Tsadikovich and Sharon Hovav

Department of Management, Bar-Ilan University, Ramat-Gan, 52900, Israel
E-mail: amir.elalouf@biu.ac.il [Elalouf]; dmitrytsadikovich@gmail.com [Tsadikovich]; SharonH@clalit.org.il [Hovav]

Received 30 June 2015; received in revised form 16 August 2016; accepted 28 August 2016

Abstract

We focus on planning transportation operations within a blood sample supply chain, which comprises clinics and a laboratory. Specifically, the main goal of this study is to obtain the optimal number of vehicles to be deployed and the scheduling of the pickup process. First, we formulate a mixed-integer programming (MIP) problem. Next, we develop a heuristic scheme composed of two heuristic algorithms and numerical search, and a new genetic algorithm. In an extensive numerical study, based on the data from a real-life blood sample collection process, we illustrate the potential of the new heuristic scheme.

Keywords: optimization; blood samples collection; MIP; tabu search; genetic algorithm

1. Introduction

Microbiological and virological laboratory testing has a key role in the management of disease. Accurate and rapid identification of relevant microorganisms is vital for guiding antimicrobial therapy and improving the outcomes of infectious disease. Specifically, many pathogens can be identified only from the analysis of blood samples. Such analysis can entail detection of antigens, culture and isolation of microbial agents, or DNA sequencing. Prior to being analyzed, blood must be collected from patients, processed for testing (through centrifugation, in which the blood is separated into its different components), and transported to the testing laboratory.

Because blood samples are perishable, it is necessary to analyze them within a specific period of time after collection in order for the results to be valid (Miller, 1998). Meeting this time constraint is one of the key challenges associated with the blood sample collection process (Alfonso et al., 2012; Yücel et al., 2013; Grasas et al., 2014). Failure to analyze samples in a timely fashion can lead to analysis errors or misdiagnosis, which may result in inappropriate treatment (Haijema et al., 2007). This challenge is compounded by several factors: First, clinics where blood samples are collected are
highly diverse and geographically dispersed (Yi, 2003), whereas the availability of testing facilities is likely to be more limited. In addition, a high percentage of samples are collected in the morning, from customers who have fasted overnight; as a result, the delivery burden is especially high during this time of day. An additional challenge is the cycle of measuring time from taking the blood samples following the doctor’s order until the results are transmitted back to the doctor (Grasas et al., 2014).

The effectiveness of a blood sample collection process, in terms of operational costs, customer satisfaction, and resource utilization, depends heavily on the manner in which the corresponding healthcare system regulates and operates its blood sample supply chain. Herein, we focus on improving the effectiveness of a blood sample supply chain comprising clinics, where blood samples are collected and processed, and a centralized testing laboratory, where blood samples are analyzed (Yücel et al., 2013; Grasas et al., 2014). Specifically, we seek to optimize the process by which blood samples are transported from clinics to the laboratory, bearing in mind that, as noted above, blood samples must be transported and analyzed within a strict time constraint.

We aim to determine the number of vehicles that should be used for sample delivery and to identify the optimal schedule for the pickup process (Revere, 2004; Grasas et al., 2014). For this purpose, we formulate a multiobjective mixed-integer programming (MIP) problem that minimizes the vehicle fleet size, total transportation time, and the expected quantity of collected samples delivered “too late” to the testing laboratory. In order to evaluate the effectiveness of the chain, we introduce two performance measures: operational efficiency and quality of service. Operational efficiency refers to operational costs, for example, the size of the vehicle fleet involved in the pickup and transportation process. Quality of service refers to the quantity of samples delivered to the testing laboratory within a certain time span after collection. We develop an efficient heuristic scheme to approximate a solution for this problem, as it is not feasible to obtain exact optimal solutions for MIP problems—even for small-sized instances—in a reasonable period of time. A comparison analysis between the MIP solutions and the outputs of the heuristic shows that the heuristic closely approximates the exact solutions, but requires substantially less CPU time. We then present a case study to investigate the capacity of this heuristic algorithm to improve the operational efficiency of the blood sample supply chain in a health maintenance organization (HMO) called Meuhedet (Israel).

The remainder of this paper is organized as follows. Section 2 presents a comprehensive literature review, and Section 3 presents a description of the problem. The mathematical formulation of the problem and description of the MIP are presented in Section 4. In Section 5, advanced heuristic scheme that includes simple heuristic, tabu search, and numerical search algorithms is proposed; this scheme is analyzed on different problem instances. Section 6 provides a comparison analysis between the MIP and heuristic algorithm, a case study, and sensitivity analysis. Finally, Section 7 presents conclusions and proposes directions for future research.

2. Literature review

Our work relates to two domains of operational problems associated with blood products. The first domain, which we refer to as blood supply chain problems, relates to blood products intended for donation or other forms of future use (see Civelek et al., 2015). We review this stream of literature with an emphasis on the vehicle routing aspects of the problem. The second domain, referred to as
clinical sample collection, refers to the collection of blood samples (or other types of clinical samples) for the purpose of medical analysis. Both domains involve allocation aspects related to perishable products; however, problems in which the blood is intended for immediate medical analysis do not involve inventory aspects.

2.1. Blood supply chains

Blood supply chains deal primarily with the collection and transportation of blood donations from geographically dispersed sites to blood banks, where units of whole blood or processed blood components (e.g., red blood cells, platelets, plasma, and frozen blood) are stored. Some of the blood components have short shelf lives, and this limitation, in addition to the limited storage space in blood banks, pose constraints on scheduling policies for collection and transportation. Operations must take these constraints into account while maintaining a stockpile that is sufficient to enable healthcare providers to respond rapidly to changing demands; for a comprehensive review of the literature on inventory and supply chain management of blood products, see Beliën and Forcé (2012).

The optimal location–allocation of blood collection centers is examined in the works of Jacobs et al. (1996), Pierskalla (2005), and Cetin and Sarul (2009). Yi (2003), Doerner et al. (2008), and Pathomsiri and Sukaboon (2013) explore different vehicle routing problems (VRPs) that aim to obtain optimal scheduling policies for blood donation delivery. Specifically, Doerner et al. (2008) extend the original VRP with time windows (VRPTW) to a VRP with multiple interdependent time windows (VRPmiTW) in which blood donations are continuously collected at each customer location during a certain production time window. The model aims to find optimal dispatching policies for the vehicles used for blood donation delivery, and, since blood is a perishable product, the model incorporates a constraint that collected samples should be delivered to the central depot within a certain time span after collection. The authors formulate an MIP and, owing to the complexity of obtaining optimal solutions, propose an effective heuristic procedure to approximate its solution. Our research also relates to more general works on the VRP for perishable products. This problem has been researched using loss of quality and value as a function of time (Osvald and Stirn, 2008), and sensitivity to material change during transport. Amponsah and Salhi (2004), for example, researched garbage collection, assuming that the garbage is affected by the environment while waiting to be collected. Naso et al. (2007) studied delivery of ready-mixed concrete under a maximum routing time constraint. Or and Pierskalla (1979) focus on the analysis of an integrated location–allocation and transportation model, which is frequently referred to in the literature as the blood transportation–allocation problem. Jabbarzadeh et al. (2014) suggest a dynamic supply chain network design for the supply of blood in disasters. In their paper, they demonstrate a robust network design model that is able to supply blood both during and after a disaster occurrence.

In addition to the works cited above, which use integer and linear programming methods to solve the problems they address, the blood supply chain literature includes several studies that use simulation tools to identify means of improving overall operational efficiency (Vrat and Khan, 1976; Rytila and Spens, 2006; Katsaliaki and Brailsford, 2007; Özgen, 2007; Mustafee et al., 2009). In turn, Catassi and Peterson (1967) and Frankfurter et al. (1974) note the role of information systems in blood supply chain management.
2.2. Clinical sample collection

The clinical sample collection process requires all samples to be tested on the day of collection; that is, in contrast to the blood donation collection process, storage is not allowed. Thus, the main challenge in the clinical sample process is to deliver the collected samples to the testing laboratory within a certain time span after collection, given limited resources and unpredictable circumstances that might delay the trip, such as traffic congestion and road closures (the detailed impacts of congestion on vehicle tour characteristics can be found in the work of Figliozzi, 2010). This problem, referred to as the blood sample collection problem (see Grasas et al., 2014), is a special case of the capacitated VRP (see Toth and Vigo, 2001). More specifically, it can be defined as a capacitated time-constrained open VRP (CTCOVRP), a time-constrained version of the capacitated open VRP (COVRP), which has been studied extensively in OR literature (Li et al., 2007). The COVRP is a VRP in which vehicles’ carrying capacity is limited and the vehicles do not need to return to their starting point (in our case, they begin at the first collection point and finish at the laboratory). There are numerous applications of the CTCOVRP, including a special version of the school bus problem known as the bus route generation problem (Park and Kim, 2007).

McDonald (1972) was the first to formulate a mathematical model based on a VRP to minimize the total traveling time associated with the collection process. Revere (2004) studied a business process of a re-engineering project for a laboratory courier service, targeted at minimizing both laboratory courier and staffing costs. That study tackled the problem by dividing it into two subproblems, each dedicated to one of the two objectives. A traveling salesman model was implemented to minimize laboratory courier costs, whereas an integer programming formulation was developed to minimize staffing costs. More recently, Yücel et al. (2009) considered the collection for processing problem, wherein a fleet of vehicles collects samples accumulated at various sites and delivers them to the processing facility. The researchers proposed an MIP model with the primary objective of maximizing the number of collected items before a defined deadline, and a secondary objective of minimizing total transportation cost. Owing to the intractability of obtaining optimal solutions for such models, even for small instances, Salman et al. (2012) proposed a prioritized bicriteria metaheuristic algorithm based on a combination of tabu search with linear programming. Recently, Yücel et al. (2013) suggested a new heuristic approach to the model suggested by Yücel et al. (2009), based on solving MIP by presenting additional constraints into the model that search for the feasible solutions. Jørgensen and Jacobsen (2012) and Jørgensen et al. (2013) used a simulation approach to improve the effectiveness of the blood sample transportation process within clinics. Liu et al. (2013) were the first to study a new type of optimization problem called the home healthcare logistics optimization problem, which consists of collecting and delivering special drugs from a hospital to patients and transportation of blood samples from patients’ homes to a laboratory. They developed a mathematical model with the objective of minimizing the maximal routing costs across all routes, and solved it by implementing the tabu search algorithm. Similar to Salman et al. (2012) and Liu et al. (2013), we use a metaheuristic algorithm based on tabu search. We compare this algorithm with the MIP and an evolutionary (genetic) algorithm in terms of the CPU time required and the quality of the solution obtained. Our choice of this type of metaheuristic algorithm is based on its capacity to solve different kinds of optimization problems efficiently (Talbi, 2009; Nesmachnow, 2014). Metaheuristics are characterized by global optimization methods, which are useful in attempts to reproduce natural phenomena or social behavior (e.g., biological evolution,
animal behavior). Such algorithms are developed to increase computational efficiency, solve larger problems, and implement robust optimization codes (Talbi, 2009).

Since blood is perishable, samples must be analyzed within a specific period of time after collection in order for the results to be valid. Supply disruptions may result in situations in which tainted samples are sent to the testing laboratory, which may have catastrophic consequences for patients and other stakeholders (Madadi et al., 2014). Several studies have examined disruptions in VRPs, resulting from unpredictable events such as vehicle breakdown during the execution stage (Li et al., 2009; Mu et al., 2011). Jiang et al. (2013) researched the use of redundancy in VRPs, considering the combination of disruptions and recovery. Wang et al. (2012) and Li et al. (2009) studied the use of time slack when considering unexpected events such as traffic jams and breakdowns.

Similarly, our study takes into account general unpredictable circumstances during the trip, such as traffic congestion and road closures, which influence transportation time. These considerations are incorporated into a utility coefficient reflecting a delay in delivery of collected samples to the testing laboratory.

3. Problem description

We study a blood sample supply chain consisting of clinics and a centralized testing laboratory. The blood sampling supply chain starts in the clinics, where nurses collect patients’ blood in tubes. The next step, also carried out in clinics, is to process collected tubes in a centrifuge. A vehicle fleet managed by an outsourced courier service is responsible for transporting centrifuged blood samples from clinics to the testing laboratory. As discussed above, as blood is perishable, delivery must be completed within a certain time span after collection. When a vehicle arrives at a given clinic, centrifuged tubes are loaded into the vehicle, and the vehicle then proceeds to pick up additional samples from other clinics. Finally, the vehicle arrives at the testing laboratory, where it unloads all the tubes it has collected. At this moment, the testing process begins. At the end of this process, the results obtained from analyzing the blood samples are recorded in a computer, and the final results are sent in electronic form to the doctors who requested the tests.

We use two performance measures to evaluate the effectiveness of the blood sampling supply chain: operational efficiency and quality of service. Operational efficiency is measured in terms of operational costs, such as the size of the vehicle fleet involved in the pickup and delivery process. Quality of service is measured as the quantity of samples delivered to the testing laboratory within a certain time span after collection. The purpose of our work is twofold: (a) to identify the optimal number of vehicles to deploy in the delivery process, by deriving optimal routing and scheduling policies; (b) to maximize the quality of service by maximizing the number of samples delivered to the testing laboratory within the allotted time window.

In addition, we adopt the following assumptions: (a) given that the real-life HMO used in our case study operates only one centralized testing laboratory, in this paper, we did not take into consideration scenarios with more than one laboratory; (b) for the sake of simplicity, we assume that the demand of patients (customers) in the clinics and the transportation times are deterministic; (c) since the size of the blood samples is sufficiently small, we assume that the vehicle capacity is unlimited; and (d) all collected blood samples must be tested on the day of collection. The fourth assumption is based on current practice at HMO Meuhedet, the HMO at the focus of our case.
study (Section 6). Specifically, a patient’s blood sample that is taken in the morning is tested in a robotics-based laboratory on the same day; results are uploaded to the information system so that the physician who requested the tests can access them that same evening. In contrast to the case of blood donations, there is no inventory planning process; samples are disposed of after testing.

4. Mathematical model

In this section, we develop an MIP model that determines the number of vehicles and the sequence of clinics visited by each vehicle. We define the following notations:

**Parameters:**
- \( G \) complete and undirected graph, \( G(C^+, D) \)
- \( C^+ \) set of nodes, \( C^+ = \{0, 1, 2, ..., n\} \); node 0 represents the testing laboratory, and \( C = C^+ \setminus \{0\} \) is the set of clinics
- \( D \) set of arcs, \( D = \{(i, j) | i, j \in C^+, i < j\} \)
- \( \eta_{ij} \) the cost per unit of travel time on arc \((i, j)\)
- \( d_{ij} \) expected travel time on arc \((i, j)\); we assume that the service time at clinic \(i\) is included in the travel time over each arc \((i, j)\)
- \( Q_i \) expected average number of accumulated blood samples in clinic \(i\)
- \( m \) index of the collection path, \( m = \{1, 2, ..., M\} \)
- \( P_m \) collection path (each vehicle is associated with a collection path), \( P_m = \{i_m^{(1)}, i_m^{(2)}, ..., i_m^{(f)}, 0\} \), where \( i_m^{(1)} \) is the first visited clinic, \( i_m^{(f)} \) is the last visited clinic, and 0 is the testing laboratory. Note that each clinic is visited exactly once in a collection path, that is, each clinic is included once in exactly one path, and each path ends at the testing laboratory.
- \( B_m \) the number of samples collected from path \(m\), \( B_m = \sum_{i \in P_m} Q_i \)
- \( \rho \) time span to deliver the blood samples from the clinics to the testing laboratory. This is the maximal amount of time that can pass from the moment when the sample is collected from the clinic until it is tested in the testing laboratory. After this period of time, the analysis of the sample is no longer valid, and the sample should be discarded
- \( U_{MAX} \) the maximum allowed utilization rate (see definition of a path’s utilization rate next)
- \( \tau \) cost associated with operating the vehicle (each collection path requires one vehicle)
- \( A \) a sufficiently large number
- \( k \) a penalty cost associated with late delivery of samples

**Decision variables:**
- \( x_{ij}^m \) a binary routing-decision variable that receives a value of 1, if clinic \(j\) is visited exactly after clinic \(i\) within path \(m\)

**Auxiliary variables:**
- \( U_m \) utilization rate of path \(m\), \( U_m = \frac{d_{0,i_m^{(1)}}^m x_{i_m^{(1)}}^m + 2 \sum_{\forall i, j \in P_m} d_{ij} x_{ij}^m}{\rho} \). This formula can be explained as follows: within a path \(m\), from the moment the vehicle leaves a clinic, new samples are taken. The expression in the numerator of \(U_m\) consists of the maximum waiting time of the sample in any given
clinic up until the vehicle’s arrival, that is, 
\[ d_{(0,i_m^1)}^m x_{(0,i_m^1)}^m + \sum_{(i,j) \in P_m} d_{ij} x_{ij}^m , \]
and the delivery time from this clinic to the testing laboratory, that is, 
\[ \sum_{(i,j) \in P_m} p_{ij} x_{ij}^m, \]
\[ p_m \] probability that the samples will not get to the testing laboratory in time. Similar to Gibson et al. (2002), Ionescu et al. (2014), and Pérez Herreroa et al. (2014), we assume that this probability, 
\[ p_m = p(U_m), \]
is a nondecreasing function of the utilization rate \( U_m \), that is, for higher values of the utilization rate, there is less probability of successfully delivering the samples in time.

\[ F_m \] loss expectancy for path \( m \). The expected quantity of samples delivered “too late” to the testing laboratory, 
\[ F_m = p_m B_m, \]
\[ F = k \sum_{m=1}^{M} p_m B_m \] total expenditure associated with loss expectancies over all paths. This function reflects a quality of service measure; that is, a lower value of this function is associated with a greater number of collected samples delivered to the testing laboratory on time.

The mathematical model is presented as follows:

\[
\begin{align*}
\min Z &= \sum_{m=1}^{M} \left[ \sum_{(i,j) \in C^+} d_{ij} \eta_{ij} x_{ij}^m + \tau \sum_{j \in C} x_{(0,j)}^m \right] + k \sum_{m=1}^{M} B_m \left[ \frac{d_{(0,i_m^1)}^m x_{(0,i_m^1)}^m + 2 \sum_{(i,j) \in P_m} d_{ij} x_{ij}^m}{\rho} \right] \\
\text{subject to} \\
&\sum_{i \in C^+} \sum_{m=1}^{M} x_{ij}^m = 1, \quad \forall j \in C \\
&\sum_{i \in C} x_{(i,0)}^m = 1, \quad \forall m \in M \\
&\sum_{i \not\in j} \sum_{m=1}^{M} x_{ij}^m = \sum_{i \not\in j} \sum_{m=1}^{M} x_{ji}^m, \quad \forall j \in C, \forall m \in M \\
&\sum_{i \in S} \sum_{j \not\in i} x_{ij}^m \leq |S| - 1, \quad \forall m \in M, \forall S \subset C^+, |S| > 1 \\
&\frac{d_{(0,i_m^1)}^m x_{(0,i_m^1)}^m + 2 \sum_{(i,j) \in P_m} d_{ij} x_{ij}^m}{\rho} \leq U_{MAX}, \quad \forall m \in M \\
&x_{ij}^m \in \{0, 1\}, \quad \forall m \in M, \forall i, j \in C^+.
\end{align*}
\]

The objective function (1) is the sum of three costs: delivery cost, cost associated with operating the vehicle, and a penalty cost associated with late delivery of blood samples. The importance of each cost is expressed by its corresponding cost parameters. The equalities in (2) state that each clinic \( i, i \in C \), is visited by at most one vehicle. The constraints (3) ensure that each collection path ends at the testing laboratory, whereas constraints (4) require that the same vehicle enters and leaves
a given clinic. Inequalities (6) indicate that all samples must be delivered to the testing laboratory within a certain time span after collection. Constraints (7) define the binary variable $x_{ij}^m$.

5. Heuristic algorithm

The procedure for determining the optimal solution for the considered problem is NP-hard; that is, the size of problems that can be solved optimally is limited. Thus, since it is not practical to obtain an optimal solution for the MIP in a reasonable amount of time, we develop a heuristic search procedure (named Advanced Heuristic). This procedure is a combination of three algorithms: (a) Basic Heuristic (Algorithm 1), (b) a tabu search based algorithm called Tabu Heuristic (Algorithm 2), and (c) a numerical search based algorithm called $M$-search (Algorithm 3). Here, we present the detailed description of each algorithm.

5.1. Basic Heuristic algorithm

This algorithm is based on an extension of the nearest neighbor (NN) algorithm. The extension of NN introduced here takes into account both distances between the clinics and accumulated number of samples at each site. This is done in order to maximize the number of collected samples delivered to the testing laboratory in time. For a more detailed description, let us first define additional notations:

$DM$ matrix of distances; this matrix is of size $[n \times n]$ and includes distances $d_{ij}$ between clinics

$PM$ matrix of collection paths of size $[n \times n + 2]$, where the last two columns are used for $U_m$ and $B_m$, respectively

At the first step (lines 01–06), the algorithm starts by generating an initial $PM$ matrix, in which each clinic is included in exactly one path (i.e., is assigned to exactly one row). Each row (path) $m$ in $PM$ is associated with corresponding values of $U_m$ and $B_m$. Next, the rows of the generated matrix are sorted in ascending order according to the values of $U_m$. At the second step (lines 07–28), for each $m/22601$, the algorithm begins searching for the nearest path $P_m$ to path $P_{(m=1)}$ based on the distance matrix $DM$ (lines 08–18). If a new combined path does not exceed $U_{MAX}$, then the new path, together with its $U_m$ and $B_m$ values, replaces path $P_m$ in the matrix $PM$ (and the original path $P_m$, together with $U_m$ and $B_m$, is eliminated) (lines 21–23). An increase in the utilization rate, $U_m$, is associated with an increase in the corresponding probability, $p(U_m)$, that the samples will not get to the testing laboratory in time. This, in turn, leads to an increase in $F_m$. In order to minimize $F_m$, we imply that the maximum allowed utilization rate, $U_{MAX}$, decreases proportionally with respect to the number of samples collected within path $m$ as follows: $U_{MAX} = U_{MAX} - \alpha B_m$, where $\alpha$ is a small number (line 21). Then, $PM$ is sorted once again by $U_m$. This procedure is repeated until there is no longer any path that can be added to path $P_{(m=1)}$. In this case, the algorithm shifts to the second row in $PM$, that is, $P_{(m=2)}$, and the procedure is reiterated. The algorithm terminates when all the paths in $PM$ have been checked. The detailed description of the algorithm is presented in Fig. 1.

**Proposition 1.** The complexity of Basic Heuristic is $O(n^2)$, where $n$ is the number of clinics.

© 2016 The Authors.
International Transactions in Operational Research © 2016 International Federation of Operational Research Societies
Fig. 1. Algorithm 1: Basic_Heuristic.

**Proof.** The complexity of the first step is $O(M)$. In the second step, the loop in lines 9–13 is done in $O(M)$ time, and the loop in lines 16–26 is also done in $O(M)$ time. These two loops are embedded in a loop (while) that starts at line 7 and ends at line 28. Since $M \leq n$, and this loop is performed $O(M)$ times, the overall complexity is $O(n^2)$.

5.2. Tabu_Heuristic algorithm

Next, we present a detailed description of Tabu_Heuristic (Algorithm 2) for determining the assignment between a given number of clinics $n$ and $M$ vehicles to minimize the objective function. Note that although Basic_Heuristic runs through several possible values of $M$, in Tabu_Heuristic $M$ is fixed.
Denote by $s_{Candidate}$ a specific assignment between the clinics and the vehicles, and by $Z(s_{Candidate})$ the value of the objective function corresponding to $s_{Candidate}$. A given instance of $s_{Candidate}$ is defined as a matrix of size $n \times M$, where each cell is a binary variable that receives a value of 1, if the clinic is included in the path of the corresponding vehicle.

The algorithm begins with an initial clinic–vehicle assignment denoted by $best_{Candidate}$, whose initial value is obtained from the execution of Basic_Heuristic (lines 01–02). The assignment $best_{Candidate}$ is improved in an iterative manner as follows: in each iteration, a collection of $s_{Candidate}$s, denoted by $candidateList$, is constructed on the basis of an alteration to the best candidate obtained in the previous iteration (denoted $last_{Candidate}$). This alteration is made by random swapping: First, two vehicles are chosen randomly (line 07). Next, for each chosen vehicle, one of the clinics assigned to that vehicle is selected randomly. Then, the two selected clinics are swapped between the vehicles (see, e.g., Fig. 2). Finally, among the $s_{Candidates}$ in $candidateList$, the $s_{Candidate}$ with the best objective function value is chosen as a new $last_{Candidate}$, to be used in the next iteration.
the objective function value of lastCandidate is better than the objective function value of bestCandidate, bestCandidate is replaced with lastCandidate (lines 16–19). This procedure is repeated until the maximum allotted time is reached (Max_Time) and then the bestCandidate and its corresponding objective function value are returned.

The remaining notations used in Tabu_Heuristic are as follows:

- \( Z_{\text{best}} \) the current best value of the objective function
- \( \text{Diff}(\text{Candidate 1}, \text{Candidate 2}) \) the difference between the Candidate 1 and Candidate 2 matrices.
- \( \text{tabuList} \) a set of Diff, determined according to the difference between the best solution of the current iteration and the best solution of the previous iteration
- \( \text{Running_time} \) the accumulated running time of the algorithm
- \( Z_{\text{best_temp}} \) the value of the objective function of the best candidate in the current candidate list

A detailed description of Tabu_Heuristic is presented in Fig. 3.
5.3. Bisection search algorithm

As noted, Tabu_Heuristic searches for the best assignment for a given number of vehicles. An increase in $M$ causes higher vehicle costs, yet reduces the expected penalty cost. The cost increases linearly in $M$, while the benefit is characterized by diminishing returns. It is possible to increase $M$ in increments of one ($M = M + 1$) and run Tabu_Heuristic until reaching the $M$ for which the cost is equivalent to the benefit. In order to find the best $M$ more efficiently, we use an $M$-search that reduces the number of iterations from $O(UB)$ to $O(\log UB)$ (see Algorithm 3; Fig. 4).

The following proposition is derived from the complexity analysis of $M$-search, which executes Tabu_Heuristic.

**Proposition 2.** The running time of $M$-search is $O(\text{Max\_Time } \log (UB))$, where Max\_Time is the maximum time allowed for Tabu\_Heuristic, and UB is the upper bound on the number of vehicles ($M$).

**Proof.** $M$-search is a binary search that runs over the interval $[1, UB]$. Thus, the number of iterations is $\log(UB)$. In each iteration, Tabu\_Heuristic is executed at most two times, and the maximum time allowed for each run is Max\_Time. Thus, the overall running time of $M$-search is $O(\text{Max\_Time } \log (UB))$.

5.4. Evaluation of the Advanced\_Heuristic procedure

In this section, we present the main steps in evaluation of the proposed Advanced\_Heuristic procedure in determining optimal number of vehicles. The process starts with generating distance matrix.
Then, Basic_Heuristic is executed, and the best assignment of clinics to vehicles, in addition to the objective function value corresponding to this assignment, are kept for each number of vehicles. These results are used as input for M-search, which directs the tabu search to find the best possible solution (in terms of the objective function value). This is done as follows: at the initial stage, M-search defines lower (LB) and upper (UB) bounds for the number of vehicles. Then, two internal points $M_1$ and $M_2$ between LB and UB are found, such that $LB < M_1 < M_2 < UB$. The next step is related to Tabu_Heuristic that searches for the best clinic-vehicle assignment ($A^*(M)$) for a given number of vehicles $M$. Tabu_Heuristic is executed twice, once for $M_1$ and once for $M_2$. This results in $A^*(M_1)$, $A^*(M_2)$ and their respective objective function values, which are denoted by $Z(A^*(M_1))$ and $Z(A^*(M_2))$, respectively. $Z(A^*(M_1))$ and $Z(A^*(M_2))$ are returned to M-search and used to update the values of LB and UB. Since the optimization problem seeks to identify the minimal value of the objective function, then in the case when $Z(A^*(M_1)) < Z(A^*(M_2))$, the LB in M-search remains the same, whereas UB is updated to $Z(A^*(M_2))$; otherwise, $LB = Z(A^*(M_1))$, and UB remains the same. This procedure is repeated until $|M_1 - M_2| = 1$; the best clinic-vehicle assignment is returned. The flowchart of Advanced_Heuristic is presented in detail in Fig. 5.

6. Computational experiments

In this section, we run a comparison analysis between the optimal solution and the heuristic. The comparison demonstrates the effectiveness of the heuristic. Since finding the optimal solution
Table 1
Determining the size of the problem

<table>
<thead>
<tr>
<th>Network size</th>
<th>Number of nodes (C)</th>
<th>Number of arcs (^a)</th>
<th>CPU time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>28</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>36</td>
<td>420</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>45</td>
<td>750</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>55</td>
<td>1502</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>66</td>
<td>3250</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>78</td>
<td>6500</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>91</td>
<td>12,500</td>
<td></td>
</tr>
</tbody>
</table>

\(^a\)Since the graph is undirected, the number of arcs is calculated as follows: \(\frac{C \times (C - 1)}{2}\).

Fig. 6. CPU time as a function of problem size.

requires a long running time, we carry out the comparison for problems of limited size. Then we run the heuristic on real-life data.

6.1. Comparison analysis between MIP and heuristic solutions

In order to validate the effectiveness and efficiency of the heuristic algorithm (Advanced_Heuristic), we carried out a comparison analysis between our approximations and actual MIP solutions. At the first stage, we sought to define the size of the problem that would be used as a template for further analysis. For this purpose, we evaluated MIPs based on different network sizes, starting from eight nodes (including the testing laboratory). Table 1 summarizes the obtained results, and Fig. 6 depicts CPU time as a function of the problem size.

In our comparison analysis, we chose to compare networks with 11 nodes, ensuring a reasonable running time of about 30 minutes. With this network size as a guideline, we used a general algebraic modeling system (GAMS) random generator to generate a set of 45 different problem instances. These instances spanned a diverse set of problem features, taking into account the stochastic nature
of traffic and quantities of blood samples collected in clinics. MIP was employed in GAMS and solved using a Gurobi solver via a NEOS solver with the following characteristics: CPU—2× Intel Xeon E5-2430 @ 2.2 GHz (12 cores total), HT Enabled. The heuristic was designed and run in Matlab. To evaluate the numerical performance, we used five different measures: vehicle fleet size, utilization rate ($U_m$), value of the objective function ($Z$), and the expected number of samples delivered to the testing laboratory in time. Table 2 summarizes the results of the comparisons for the 45 problem instances. The columns “#v,” “UR,” “Obj,” and “EX” in Table 2 describe the vehicle fleet size, utilization rate (in percentage), the value of the objective function, and expected number of samples delivered on time, respectively.

The columns “UR” and “#v” in Table 2 show that, on an average, the solutions obtained with the heuristic are close to the actual solutions of the MIP. However, the column “Obj” indicates that the value of the objective function obtained with the MIP ($1132.5$) is superior to the solution obtained with the heuristic ($1294.5$). Overall, these results suggest that the proposed heuristic provides a “quite good” approximation of the MIP solution. A paired $t$ confidence interval test supports this proposition, showing that, for $\alpha = 0.1$, there is no difference between the heuristic and the MIP. It is noteworthy that the solution proposed by the heuristic costs 14% more than the optimal solution. However, as we will show in the case study (Section 6.4), implementation of the suggested heuristic on real-life data has the potential to lead to significant cost savings, as compared with current routing practices. Moreover, unlike the MIP, this heuristic can be applied to a wide range of large-scale real-life problems.

### 6.2. Comparison analysis between advanced heuristic and GA

To verify the effectiveness of the suggested advanced heuristic, we carried out a comparison analysis with a genetic algorithm (GA). Specifically, the GA starts by generating an initial population of chromosomes, where each chromosome is a vector of $n$ randomly generated numbers between 0 and 1. Next, each chromosome is translated into a corresponding feasible solution such that the constraints of the mathematical model are not violated. Based on the current obtained set of solutions, a new population and corresponding set of solutions are constructed. This process is repeated for a prescribed number of iterations. A detailed description of the GA can be found in the Appendix.

The computational results based on 45 instances are summarized in Table 3. On average, the advanced heuristic outperforms the GA. However, in 11 of 45 runs, the solution of the objective function obtained with the GA was slightly better than the solution obtained with the advanced heuristic.

---

**Table 2**

Comparison analysis between MIP and heuristic

<table>
<thead>
<tr>
<th>Parameters</th>
<th>MIP</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>#v</td>
<td>2.8</td>
<td>3.1</td>
</tr>
<tr>
<td>UR</td>
<td>25%</td>
<td>28%</td>
</tr>
<tr>
<td>Obj in ($)</td>
<td>1132.5</td>
<td>1294.5</td>
</tr>
<tr>
<td>EX</td>
<td>676</td>
<td>672</td>
</tr>
</tbody>
</table>
Table 3
Comparison analysis between the GA and the advanced heuristic

<table>
<thead>
<tr>
<th>Parameters</th>
<th>GA</th>
<th>Advanced heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#v</td>
<td>UR</td>
</tr>
<tr>
<td>Average</td>
<td>3.3</td>
<td>28%</td>
</tr>
</tbody>
</table>

Table 4
Calculated running times (CPU(s)) for the proposed heuristic for different problem sizes (#CL)

<table>
<thead>
<tr>
<th>#CL</th>
<th>CPU(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>568</td>
</tr>
<tr>
<td>125</td>
<td>844</td>
</tr>
<tr>
<td>150</td>
<td>1136</td>
</tr>
<tr>
<td>175</td>
<td>1405</td>
</tr>
<tr>
<td>200</td>
<td>1825</td>
</tr>
<tr>
<td>225</td>
<td>1945</td>
</tr>
<tr>
<td>250</td>
<td>2245</td>
</tr>
</tbody>
</table>

6.3. Running times for the suggested heuristic

In this section, we show the running time (column “CPU(s)” in Table 4) of the proposed heuristic algorithm as a function of the problem size (column “#CL” in Table 3). In the experiments, the number of clinics is varied between 100 and 250.

From Table 4, it clearly follows that “CPU(s)” is strictly increasing as a function of “#CL.” Overall, however, the computational time for the largest investigated instance, with 250 clinics, is about 40 minutes. This computational time is still reasonable, indicating that it is possible to implement the proposed heuristic successfully to find near-optimal solutions in real-life environments.

6.4. Case study

We evaluate the performance of our heuristic on data from HMO Meuhedet, a healthcare organization in Israel comprising more than 250 clinics throughout the country. Currently, decision makers at Meuhedet rely primarily on historical routing practices, making decisions myopically, based on intuition, according to sets of dynamic ongoing constraints in the blood sample collection process. For example, choices such as adding a clinic to a vehicle’s route or otherwise altering a vehicle’s path are not examined from the perspective of the entire supply chain; rather, a new clinic is likely to be added to the route of the vehicle traveling closest to it, assuming that the new route still meets the time constraint. Clearly, however, quantitative analysis, such as the analysis facilitated by the proposed heuristic, is necessary in order to make the most economical and operationally efficient decisions and thereby to best achieve the goals of the organization. The key contribution of the heuristic is a novel technique for obtaining both vehicle fleet size and sequence of clinics.
Table 5
Case study—heuristic solution

<table>
<thead>
<tr>
<th>Maximum utilization rate $U_{MAX}$</th>
<th>#v</th>
<th>Aur (%)</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.80</td>
<td>11</td>
<td>49.5</td>
<td>10</td>
</tr>
<tr>
<td>0.85</td>
<td>10</td>
<td>50.6</td>
<td>9</td>
</tr>
<tr>
<td>0.90</td>
<td>8</td>
<td>56.9</td>
<td>7</td>
</tr>
<tr>
<td>0.95</td>
<td>6</td>
<td>63.3</td>
<td>4</td>
</tr>
</tbody>
</table>

that each vehicle should visit, taking into account timing and quality constraints. The simplicity of the algorithm and its short run time enable it to be easily used to investigate different scenarios encompassing a variety of parameters.

We start off by determining the lower ($LB$) and upper ($UB$) bounds for the fleet size recommended by the proposed heuristic algorithm. These bounds will be used in our case study to measure the quality of performance of the heuristic, implemented on real-life, large-scale data.

Let us first assume that the utilization rate $U_m$ for every single clinic is less than $U_{MAX}$, $U_m \leq U_{MAX}$. This assumption is reasonable because it is obvious that, for $U_m > U_{MAX}$, no single vehicle can visit the clinic and deliver the samples in time, and thus no feasible solution for this problem exists. Furthermore, it is clear that the maximal possible number of vehicles cannot exceed the total number of clinics; that is, $UB = (n - 1)$. In turn, the lower bound is calculated as $LB = \frac{\bar{U}}{U_{MAX}}$, where $\bar{U}$ is a utilization rate of the shortest possible route that visits each clinic exactly once and returns to the origin node (testing laboratory). The calculation of the length of this route is based on the well-known traveling salesman problem (TSP). For information on TSP solution algorithms, we refer the interested reader to the paper by Laporte (1992).

Next, for our case study, we focus on the data from HMO Meuhedet’s largest urban district, Tel Aviv-Jaffa, which comprises 50 Meuhedet clinics and a central testing laboratory. The Tel Aviv-Jaffa district is characterized by high traffic that moves in unpredictable patterns, and thus it is challenging to obtain optimal scheduling policies for the organization’s vehicles. The necessary data for implementation of the algorithm, such as transportation time and the numbers of samples collected in clinics, are derived from the HMO’s information systems. The main output of the heuristic algorithm is twofold: (a) vehicle fleet size (operational efficiency) and (b) average utilization rate—$U_m$ (quality of service). Currently, Meuhedet operates 14 vehicles per day in its Tel Aviv-Jaffa district, with an average utilization rate of about 55%. The maximal utilization rate, $U_{MAX}$, defined by the decision makers at Meuhedet, varies between 85% and 90%. The implementation of the heuristic leads to the following results (see Table 5).

As follows from the solution outlined in Table 5 column “#v,” the suggested vehicle fleet size varies between 8 and 10 (9 on average), and “Aur” ranges between 50.6% and 56.9%. This, in turn, leads us to the conclusion that the current vehicle fleet size can be reduced from 14 to 9 vehicles. The savings that would arise from reducing the vehicle fleet size amount to $1,150 daily or about $287,500 annually.

To check the quality of performance of the heuristic, we use the lower and upper bounds defined in Section 6.4. As mentioned above, the upper bound, $UB$, is equal to the number of clinics, that is, $UB = 50$. The lower bound, $LB$, is the ratio between the utilization rate of the shortest possible
route that visits each clinic exactly once, $\tilde{U}$, and the maximal utilization rate, $U_{MAX}$. Parameter $\tilde{U}$ is found by solving a TSP and is equal to 6.48, which, in turn, implies that $LB = 8$ vehicles. As the data presented in bold in Table 5 shows, the heuristic recommends nine vehicles (on average), a number that is close to the lower bound; this indicates the high quality of performance of the proposed algorithm.

HMO Meuhedet operates in a complex and unpredictable environment, meaning that the data used as input for the model may be highly variable. To assess the impact that changes in specific parameter values might have on the model's conclusions and to determine which parameters are the key drivers of the model's results, we carried out a sensitivity analysis, presented in what follows.

### 6.5. Sensitivity analysis

We start off by investigating the influence of the maximum utilization rate ($U_{MAX}$) on the output of the algorithm (see Table 6). The columns “#v” and “Aur” present the vehicle fleet size and the average utilization rate in percentage, respectively.

Figure 7 depicts the vehicle fleet size as a function of the maximal utilization rate. The figure clearly shows that the vehicle fleet size is negatively dependent on the maximal utilization rate ($U_{MAX}$).

As discussed above, transportation time within urban networks depends heavily on road conditions, including traffic congestion, road repairs, and so on. Thus, we carried out a sensitivity analysis that takes into consideration the possibility of variability in transportation time. On the basis of the collected data, we estimated that the transportation times could differ from their original values by a factor of 0.1–2. We then ran the algorithm using new input data (see Table 7 and Fig. 8). The results indicate that incorporation of variability in transportation times does not substantially affect the number of vehicles required (see column “#v” in Table 7).

![Table 6](https://example.com/table6.png)

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Maximum utilization rate</th>
<th>#v</th>
<th>Aur (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.50</td>
<td>23</td>
<td>34.8</td>
</tr>
<tr>
<td>2</td>
<td>0.55</td>
<td>21</td>
<td>35.8</td>
</tr>
<tr>
<td>3</td>
<td>0.60</td>
<td>18</td>
<td>40.1</td>
</tr>
<tr>
<td>4</td>
<td>0.65</td>
<td>15</td>
<td>44.6</td>
</tr>
<tr>
<td>5</td>
<td>0.70</td>
<td>14</td>
<td>47.1</td>
</tr>
<tr>
<td>6</td>
<td>0.75</td>
<td>13</td>
<td>47.1</td>
</tr>
<tr>
<td>7</td>
<td>0.80</td>
<td>11</td>
<td>49.5</td>
</tr>
<tr>
<td>8</td>
<td>0.85</td>
<td>10</td>
<td>50.6</td>
</tr>
<tr>
<td>9</td>
<td>0.90</td>
<td>8</td>
<td>56.9</td>
</tr>
<tr>
<td>10</td>
<td>0.95</td>
<td>6</td>
<td>63.3</td>
</tr>
</tbody>
</table>
The time span, $\rho$, during which blood can be delivered to the laboratory after collection has a tremendous influence on the operation of the blood supply chain. Table 8 and Fig. 9 show the results of a sensitivity analysis in which we varied the value of this parameter in the range of 4–10 hours. Figure 9 clearly shows that reducing the maximum time span between sample collections and testing causes a substantial increase in the number of vehicles required for blood sample delivery. For example, for a maximum utilization rate of 70%, 33 vehicles are required when the time span is 4 hours, whereas 10 vehicles are required when the time span is 10 hours.

The time span, $\rho$, during which blood can be delivered to the laboratory after collection has a tremendous influence on the operation of the blood supply chain. Table 8 and Fig. 9 show the results of a sensitivity analysis in which we varied the value of this parameter in the range of 4–10 hours.

Figure 9 clearly shows that reducing the maximum time span between sample collections and testing causes a substantial increase in the number of vehicles required for blood sample delivery. For example, for a maximum utilization rate of 70%, 33 vehicles are required when the time span is 4 hours, whereas 10 vehicles are required when the time span is 10 hours.
Table 8  
Sensitivity analysis—time span after collection ($\rho$)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\rho = 4$ hours</th>
<th>$\rho = 6$ hours</th>
<th>$\rho = 10$ hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum utilization rate (%)</td>
<td>#v</td>
<td>Aur (%)</td>
<td>#v</td>
</tr>
<tr>
<td>70</td>
<td>33</td>
<td>53.4</td>
<td>19</td>
</tr>
<tr>
<td>80</td>
<td>31</td>
<td>56.2</td>
<td>15</td>
</tr>
<tr>
<td>90</td>
<td>24</td>
<td>62.1</td>
<td>12</td>
</tr>
<tr>
<td>100</td>
<td>21</td>
<td>65.9</td>
<td>11</td>
</tr>
</tbody>
</table>

Fig. 9. Sensitivity analysis—time span ($\rho$).

7. Conclusions

This research is an attempt to develop an operations research tool that can be used to investigate and improve the operational efficiency of blood sample supply chains. Such tools can be useful for researchers and practitioners who seek to understand blood sample supply chains from a broader perspective. Specifically, we propose an MIP model that can be used to obtain optimal solutions—that is, minimize vehicle fleet size, total transportation time, and the risk that samples will not be delivered in time—for small-scale VRPs in the context of blood sample pickup and delivery. Because substantial CPU time is required to obtain solutions even for small-scale instances of the MIP model, an effective heuristic scheme is proposed to approximate the MIP solution. A comparison analysis shows that the heuristic obtains results that are close to the MIP solution yet requires far less CPU time.

We implemented the heuristic to analyze data from HMO Meuhedet (Israel). We found that the HMO’s current vehicle fleet size in a busy urban district can be reduced from 14 to 9. Such a change would yield annual savings of about $287,500. In order to estimate the impact that changes in certain parameters might have on the algorithm’s outputs, and to identify the parameters are the key drivers of the model’s results, we carried out a sensitivity analysis. The results of this analysis lead to the conclusion that the time span after collection ($\rho$) is the most influential parameter. Even a small change in this parameter can cause substantial variations in vehicle fleet size.

We believe that managers can use the proposed heuristic algorithm to investigate and improve the operation of blood sample supply chains in different healthcare organizations. However, there are still many open research questions that should be addressed in order to obtain a more complete understanding of the operation of blood supply chains. In particular, we propose that stochastic
behavior of the system should be taken into account in the assessment of the optimal vehicle fleet size and pickup schedule. An additional promising direction for future research is to address issues related to reconstruction of the existing blood supply chain by adding new testing laboratories as well as clinics, and the influence of such efforts on the effectiveness of the entire chain.

Acknowledgment

This work was supported by grant 135/2012 from the Israel National Institute for Health Policy Research.

References


© 2016 The Authors.
International Transactions in Operational Research © 2016 International Federation of Operational Research Societies


### Appendix

**Genetic algorithm (GA)**

Each chromosome in the initial population contains $n$ genes associated with $n$ clinics.

**Main procedure: genetic algorithm (GA)**

1. **Initialization:**
   a. Generate initial population
   b. $\text{Counter}_i = 0$
2. For $\text{Counter}_i = 1$ to max_iterations
3. **Interpret** each chromosome into a possible solution, $s \in S$, where $S$ is a set solutions
4. Create $SE$ and $\overline{SE}$ ($SE$ is best 20% of chromosomes)
5. Generate new population $S$ by
   a. 20% $\rightarrow SE$
   b. 79% $\rightarrow \text{Crossover}$ of $SE$ and $\overline{SE}$
   c. 1% $\rightarrow$ Mutant generation
6. Next $\text{Counter}_i$
7. Find best chromosome in $S$
8. Return best solution

**Interpret procedure**

Input: a chromosome; Output: set of feasible paths

1. Let the sequential vector be Seq = (1,2, ..., $n$)
2. For each chromosome in the population do
   a. Combine Seq and the chromosome vector into a 2-row matrix (MX)
   b. Sort MX according to ascending order with respect to the chromosome row
   c. $m = 1$, $i = 1$
   d. while $i \leq n$ do
   e. $\text{path}(m) = \{\}$
3. Endfor

**Crossover procedure**

1. Choose a random vector from $SE$, $V_{SE} \in SE$, and a random vector from $\overline{SE}$, $V_{\overline{SE}} \in \overline{SE}$.
2. Create a random vector with $n$ values ($V_{\text{RND}}$) distributed Uniform(0,1)
3. $V_{\text{child}} = \{}$
4. For $i = 1$ to $n$
   a. If $V_{\text{RND}} < 0.7$ then $V_{\text{child}}(i) = V_{SE}(i)$
      else $V_{\text{child}}(i) = V_{\overline{SE}}(i)$
5. Endfor
6. End

In a GA, we start off by generating an initial population of chromosomes. Each chromosome is a vector of $n$ randomly generated numbers between 0 and 1 (Procedure GA 1a). Using the procedure “Interpret,” each chromosome is translated into a corresponding feasible solution such that the constraints of the mathematical model are not violated. Based on the current set of solutions, the population is then partitioned into two groups: $SE$, the best 20% of the solutions in terms of the objective function; and $\overline{SE}$, the remaining solutions. Next, the new population is constructed from a combination of three groups (see line 5 in GA).